

DIGITAL CIRCUITS

PREREQUISITES: MODULE 02: INTRODUCTION.

OUTLINE OF MODULE 07:

What you will learn about in this Module:

- Combinatorial Logic
- Standard Logic Gates
- Logic families (operating characteristics)
- Standard digital function packages:
 - Logic gates
 - Oscillators and clocks
 - Shift registers
 - Digital Multiplexers (Mux – Demux)
- High speed logic
- Micro-power design considerations
- Internet-based resources: datasheets, sample parts, application notes

What you will build in the lab:

You will build simple combinatorial logic circuits that will detect the states of switches, and then set an output state based on the input states.

INTRODUCTION:

Digital logic chips are extraordinarily easy to use if you follow a few design guidelines. Recently, individual digital logic chips are increasingly being replaced by programmable gate array (PGA) logic, microprocessors, digital signal processors (DSPs) and other sophisticated and highly integrated digital devices. Individual logic chips are still in use as “glue logic”, providing all of the simple but important functions that relate to interfacing digital devices to the Real World, and performing simple functions within otherwise complex devices. While doing the readings, pay attention to the fact that there are several different “families” of logic chips, and that they sometimes require special handling to avoid destroying the integrated circuit.

READINGS FROM HOROWITZ AND HILL (H&H): *ART OF ELECTRONICS*

Read all of Chapter 8. This is a very long chapter, so focus on:

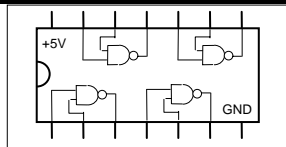
- 8.01-8.04
- 8.06 (note that gates are interchangeable!)
- 8.07, 8.08 and Table 8.2 (page 484)
- 8.9-8.12, 8.14
- Logic Pathology (pg 551)
- 8.33-8.35
- Chapter 9: Introduction, 9.01-9.08
- 4.14-4.16 (micropower design)

ADDITIONAL READINGS & INTERNET RESEARCH:

Several common logic chips have been selected and are available in the laboratory. They are all CMOS logic chips and operate with a power supply of 5 volts. You should go to the Internet and have a look at the datasheet for each of the devices listed below. Good web pages to search will be the corporate sites for Texas instruments, National Semiconductor, and Motorola. For each chip, write down the logic function it performs, draw the chip showing how each pin is connected, and write a truth table for the logic function it performs. For the Truth Tables: H = HIGH (logical "True"); L = LOW (logical "False"); X = does not matter.

| Chip | Function | Pin Diagram | Truth Table for each logic gate |
|------|----------|-------------|---------------------------------|
|------|----------|-------------|---------------------------------|

| 74HC00 | Quad NAND |  | <table border="1"> <thead> <tr> <th colspan="2">Inputs</th> <th rowspan="2">Output</th> </tr> <tr> <th>A</th> <th>B</th> </tr> </thead> <tbody> <tr> <td>H</td> <td>H</td> <td>L</td> </tr> <tr> <td>L</td> <td>X</td> <td>H</td> </tr> <tr> <td>X</td> <td>L</td> <td>H</td> </tr> </tbody> </table> | Inputs | | Output | A | B | H | H | L | L | X | H | X | L | H |
|--------|-----------|---|---|--------|--|--------|---|---|---|---|---|---|---|---|---|---|---|
| Inputs | | | | Output | | | | | | | | | | | | | |
| A | B | | | | | | | | | | | | | | | | |
| H | H | | | L | | | | | | | | | | | | | |
| L | X | H | | | | | | | | | | | | | | | |
| X | L | H | | | | | | | | | | | | | | | |
| 74HC02 | | | | | | | | | | | | | | | | | |
| 74C08 | | | | | | | | | | | | | | | | | |
| 74HC14 | | | | | | | | | | | | | | | | | |



| Inputs | | Output |
|--------|---|--------|
| A | B | |
| H | H | L |
| L | X | H |
| X | L | H |

SELF QUIZ

1: What are the main functional differences between the following two digital logic families: TTL and CMOS.

2: Which logic family is most susceptible to damage from ESD (electrostatic discharge)?

3: In what single important functional feature does an HCT logic gate differ from an HC logic gate?

4: What family of digital logic has can operate over the greatest range of power supply voltages?

PLEASE ANSWER THE ABOVE QUESTIONS AND E-MAIL TO THE INSTRUCTOR
“I have neither given nor received aid on this examination, nor have I concealed any violation of the Honor Code”

X _____

LABORATORY PROJECTS

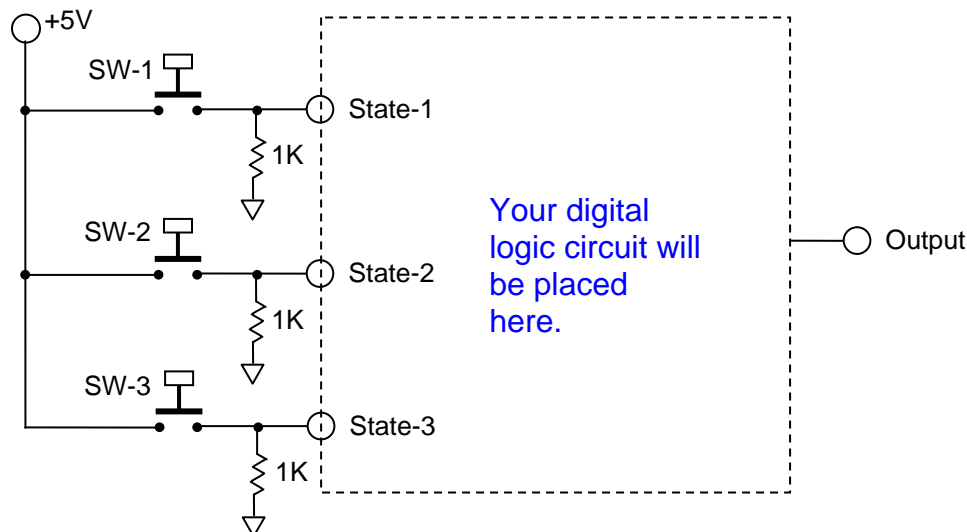
We will use “momentary ON” push button switches to provide inputs to the logic chips. When talking about logic states, there is some terminology that we will use throughout this course to represent a single bit logical value:

True = HIGH = H = 1 = ON = +5 Volts = Positive supply voltage or “rail”

False = LOW = L = 0 = OFF = 0 volts = Ground

These values are used more or less interchangeably in many different contexts, so you should get used to it. There are many factors that can complicate this simple scheme, such as the fact that not all logic chips operate at +5.0 Volts any more: modern logic chips are operating at lower and lower voltages to simultaneously increase the switching speed while decreasing the power consumption.

Thus, for our push button switches, they are “ON” when pressed and held, and “OFF” when released. This allows us to enter a digital logic state of “True” or “False” by just pressing or releasing each switch. Since the switches only make electrical contact when they are pressed, we also need to add “pull down” resistors, to pull the input voltage down to the ground state (0 volts) when the switch is not pressed. So, when the switch is pressed, it connects +5V directly and asserts the state as “ON”; when the switch is released there is no connection to +5V, so the state is pulled to “OFF” through the 1K resistor.



For now you can just wire up the three pushbutton switches. Place the switches in a row and about ½” apart on your PC Board. Then use a marker to label them SW-1, SW-2, and SW-3. Then, test that each switch output state works using the DVM (volt meter). Next you will draw several logic diagrams to perform some simple combinatorial logic functions. Later you will use logic chips to fill in the dashed box so that the OUTPUT will be defined by the three input logic States.

Design a logic diagram that will output TRUE only under the following conditions:

Use ExpressSCH to draw each of the diagrams.

(do not build the circuits yet...just draw the logic diagrams for now)

(be sure to use only logic gates and functions that are available in the lab)

A: $SW-1 = TRUE$ and $SW-2 = TRUE$ and $SW-3 = TRUE$

B: $(SW-1 = TRUE$ or $SW-2 = TRUE)$ and $SW-3 = TRUE$

(use a regular OR function: either input or both may be true at the same time)

C: $(SW-1 = TRUE$ exclusive or $SW-2 = TRUE)$ and $SW-3 = TRUE$

(the exclusive OR function (XOR) means that only one or the other input may be TRUE, but not both at the same time)

D: $(SW-1 = TRUE$ XOR $SW-2 = TRUE)$ and $SW-3 = FALSE$

E: $SW-1 = TRUE$ and $SW-2 = TRUE$ and $SW-3 = FALSE$

F: $SW-1 = SW-2 = SW-3$ (all three inputs must be the same)

G: $SW-1 = \overline{SW-2} = SW-3$ (a bar over a logic state means "invert" or NOT)

H: $(SW-1 = SW-2)$ and $SW-3 = TRUE$

Write out the Truth Table for each of your diagrams above to verify that it will result in a TRUE output *only* if the inputs meet the specified conditions.

Now that you have drawn several logic diagrams, pick one or two of them and go ahead and build them on your PC Board. Don't waste your time building all of them. Your diagrams will fit in the dashed box in the circuit diagram above. Note: you can use the same three switches to drive several logic functions; you will just have different output pins for each of the functions that you build. You may end up with a few unused logic gates, this is common and it is OK. Be sure to intelligently assert (connect) any unused logic gate *inputs* (not the unused *outputs*...see below).

Once you have built the circuits you wish to build, you should test the output of each one for all possible combinations of switch inputs (three switches will give you 8 possible input state combinations). Does the actual function match your Truth Table?

Tips for building circuits with logic chips:

- 1- Make sure that all chips in your circuit are within one logic family type. Some types of logic are not compatible with each other.
- 2- Use CMOS chips whenever possible. They consume *much* less power than the old-fashioned TTL logic chips, they have nearly "ideal" input and output characteristics, and they have become very common and readily available.
- 3- Always be sure to provide a clean power source of the appropriate voltage (this will always be +5 Volts for this laboratory, unless stated very clearly otherwise).
- 4- All unused logic gate **inputs** must be "asserted", i.e., you must connect them to a fixed logic state. In our case this will be either +5V or ground. Unconnected logic inputs can "drift" causing unpredictable logic behavior and generally messing things up. You should connect the unused input in such a way that it makes sense: for example, if you are only using two inputs on a triple-input AND gate, you should tie the third unused input to +5V, so that the gate acts correctly as a two-input AND gate.
- 5- All unused logic gate **outputs** must be left unconnected, so as not to draw excessive power.
- 6- It is often a good idea to put a small capacitor directly across the power pins for *each* logic chip in a circuit. The capacitor should be a ceramic chip capacitor in the range of 0.1 μF to 1 μF . The capacitor is usually connected on the bottom of the circuit board diagonally from the +5V pin to the ground pin, in order to *make the leads as short as possible*. The addition of one of these capacitors to each logic chip has the effect of smoothing out the power supply to each chip, which eliminates flaky behavior when you get a lot of logic chips acting simultaneously. When many logic gates change their logic state at the same time, they can cause enough current flow to result in "ground bounce" which can confuse the logic chip for an instant. To prevent this, the capacitor is added across the power pins. This is how the pros do it.

FEEDBACK

Was this Module useful and informative?

Is there a topic that should get more or better coverage?

In what way can this Module be improved:

Content: _____

Depth of Coverage: _____

Style: _____

Any additional comments that will help us to improve this course:

If you prefer, you may e-mail comments directly to Bob Dennis: yoda@umich.edu