LABORATORY MANUAL

CE/CZ1005 : Digital Logic

[Location: Hardware Laboratory 3, N4-B1a-05]

Experiment 1 :

Logic Gates and
Integrated Circuits

2017/2018

COMPUTER ENGINEERING COURSE
COMPUTER SCIENCE COURSE

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

LOGIC GATES AND INTEGRATED CIRCUITS

1. OBJECTIVES

1.1 To investigate the logic behaviour of a NAND gate.

1.2 To investigate the logic operation of an Exclusive-OR gate.

1.3 To implement a simple combinational logic circuit (a full adder) using logic gates.

1.4 To obtain the three basic logic operations from NAND gates.

2. LABORATORY

This experiment is conducted at **Hardware Laboratory 3**, N4-B1a-05.

3. EQUIPMENT

3.1 Instruments

RDA 2008A Digital-Analog Trainer
Oscilloscope
Digital Multimeter (DMM)

3.2 Components

Component		Unit Cost (in S$ for 100+ devices)
74LS00 IC	Quad two-input NAND	0.45
74LS86 IC	Quad two-input exclusive-OR	0.45
74LS08 IC	Quad two-input AND	0.45
74LS32 IC	Quad two-input OR	0.45

4. INTRODUCTION

4.1 In general, logic gates have one or more inputs and only one output. The circuits respond to various input combinations, and a truth table shows the relationship between a circuit's inputs and its output. The truth table for a particular circuit describes how the circuit behaves under normal conditions. Familiarization with a logic circuit's truth table is essential to the engineer before he or she can design with or troubleshoot the logic gate.

You should recall that the logic levels, 0 and 1, have voltage assignments. For TTL (Transistor-Transistor Logic) circuits, a logic 0 can be anywhere from 0V to +0.8V, and a logic 1 is in the range of +2.0V to +5.0V. CMOS circuits will have different voltage range.

In this experiment, five logic functions are covered: OR, AND, NOT, NAND and XOR.

The OR operation can be summarized as follows:

1) When any input is 1, the output is also 1.
2) When all inputs are 0, the output is also 0.

The AND operation can be summarized similarly:

1) When any input is 0, the output is also 0.
2) When all inputs are 1, the output is also 1.

The NOT operation is said to be complementary. In other words:

1) If the input is 0, the output is 1.
2) If the input is 1, the output is 0.

The NAND gate is nothing more than an AND gate whose output has been passed through a NOT gate. The fact that a NAND gate can be used to create all other gates is important, because it is this fact which has made the NAND gate popular in use. Most digital circuits consist of combinations of the basic logic gates. In this experiment, you will demonstrate that NAND gates can be used to obtain all the basic logic gates.

4.2 Two Boolean expressions occur quite frequently in designing combinational circuits:

1) $X = A'B + AB' = A \oplus B = A \text{ XOR } B$

2) $Y = A'B' + AB = (A \oplus B)' = A \text{ XNOR } B$

The first expression defines the exclusive-OR (i.e. XOR) function to be one which yields an output that is HIGH whenever its inputs are different. Conversely, the second expression defines the exclusive-NOR (i.e. XNOR) function to be one which yields an output that is HIGH whenever its inputs are the same. While these circuits are combinational circuits, they have been given their own symbols, and both have been implemented with ICs (integrated circuits).

4.3 A simple application of the XOR circuit is the arithmetic addition of three 1-digit binary numbers. Table 1 illustrate the possible results from such an addition (Note that arithmetic addition is different from logical addition). This circuit is called a full adder. This circuit has three inputs and two outputs.

The logic expressions of the two outputs are:

$$\text{Sum} = A \oplus B \oplus \text{Cin}$$

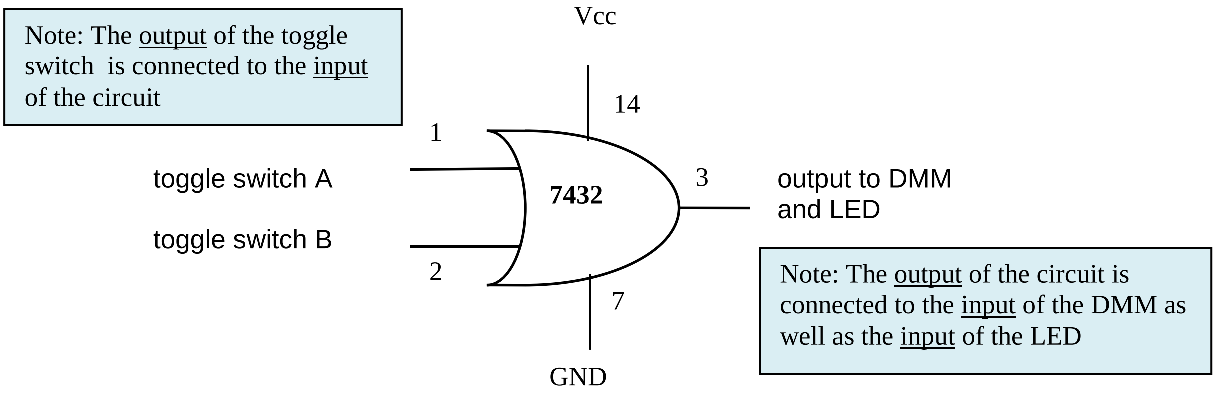
$$\text{Cout} = A \bullet B + B \bullet \text{Cin} + A \bullet \text{Cin}$$

Table 1: Truth table of a full adder

Inputs			**Outputs**		**Decimal equivalent**
A	**B**	**Cin**	**Cout** (msb)	**Sum** (lsb)	
0	0	0	0	0	0 + 0 + 0 = 0
0	0	1	0	1	0 + 0 + 1 = 1
0	1	0	0	1	0 + 1 + 0 = 1
0	1	1	1	0	0 + 1 + 1 = 2
1	0	0	0	1	1 + 0 + 0 = 1
1	0	1	1	0	1 + 0 + 1 = 2
1	1	0	1	0	1 + 1 + 0 = 2
1	1	1	1	1	1 + 1 + 1 = 3

Notice that the sum bit can be easily obtained using one 3-input XOR gate, or two 2-input XOR gates. The carry bit is generated by another simple logic circuit using AND and OR gates.

4.4 For preparation of the lab, you are expected to use a logic circuit connection diagram, such as the one shown in Figure 1, to help you in setting up the experiment. This diagram is drawn with reference to the device's pinout configuration (see Appendix) stipulated on its datasheet. The actual pin numbers may vary depending on which one of the several logic gates on the device that you choose to use.



Note: The output of the toggle switch is connected to the input of the circuit

Note: The output of the circuit is connected to the input of the DMM as well as the input of the LED

7432 is a quad 2-input OR IC

1, 2, 3, 4, 7 and 14 are pin numbers

Figure 1: Logic circuit connection diagram (OR gate)

5. EXPERIMENT

5.1 The NAND Gate

5.1.1 Figure 2 shows the logic symbol, and an equivalent circuit for a two-input NAND gate.

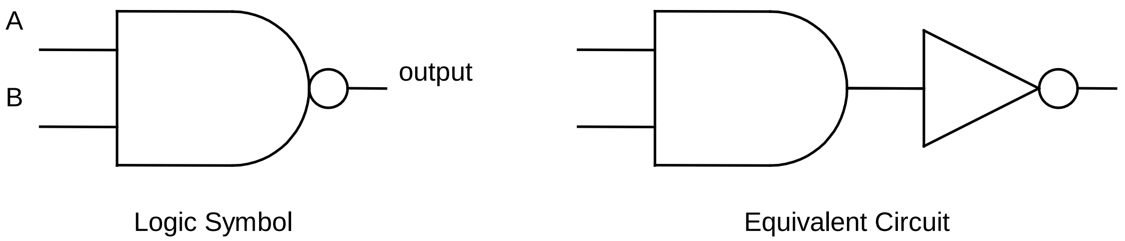


Figure 2: NAND

5.1.2 You may refer to the *Lab 1 circuit connection guide.pdf* (steps 1 – 9) for a pictorial guide.

The 74LS00 IC contains four independent NAND gates (see pinout assignment in the Appendix). Install a 74LS00 device on the circuit board **(make sure that the IC sits firmly across the gutter with all its metal pins fully inserted into the circuit board).**

Select any one of the four NAND gates and connect it as follows:

a) Vcc to +5 V and GND to power ground. **Every IC requires Vcc and GND to be properly connected.** Pause and think: Why are these connections necessary?
b) Inputs A and B to toggle switches.
c) Output to an LED.

Sketch the circuit connection diagram complete with pin numbers and labels in advance on a sheet of paper that you can refer to while connecting up the circuit in the lab.

5.1.3 Verify the NAND operation by setting the toggle switch inputs A and B to each of the logic value combinations listed in Table 2. Measure and record the output voltage of the NAND gate with a DMM (digital multimeter). Make sure that the DMM is set to measure DC voltage in the correct voltage range. The black probe should touch GND while the red probe touches the output pin. Convert the output voltage to a logic level, using

0V - 0.8V = logic 0 and 2V - 5V = logic 1

for the conversions and record your observations in Table 2. Pause and think: is it important to record the voltage up to 3 or 4 significant figures?

Table 2

Data switches		Output voltage (volts)	Logic level (0/1)	LED (On/Off)
A	B			
0	0			
0	1			
1	0			
1	1			

5.1.4 Based on your knowledge of the NAND gate, is your circuit operating correctly?

5.1.5 You may refer to the *Lab 1 circuit connection guide.pdf* (steps 10 – 13) for a pictorial guide.

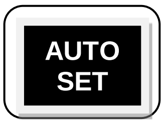
As shown in Figure 3, disconnect input B from the toggle switch and connect it (i.e. input B) to the **TTL** output of the FUNCTION GENERATOR. Select square wave and set the frequency to

1 kHz by adjusting the two knobs. Input A remains connected to the toggle switch. Pause and think: Why is it essential to disconnect input B from the toggle switch?

The next few steps will allow you to observe the variation of circuit's input and output with time using the oscilloscope. Using additional wires,

- connect the NAND output to CHANNEL 2 of the oscilloscope, and
- connect the signal at input B (i.e. the 1 kHz square wave) to CHANNEL 1 of the oscilloscope.

Note that you are now monitoring input B and the output of the NAND gate using CHANNEL 1 and CHANNEL 2 of the oscilloscope respectively. If you do not get a stable display on the oscilloscope, press the AUTO SET button on the oscilloscope. If you still cannot get a stable display, ask for help.



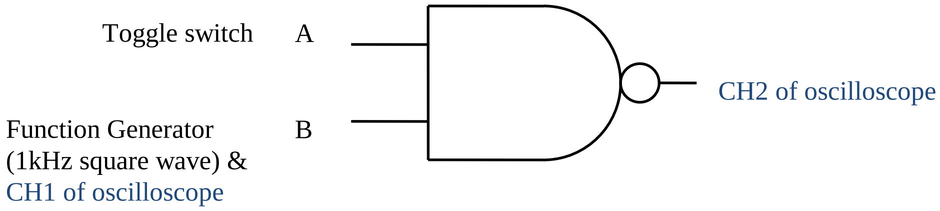


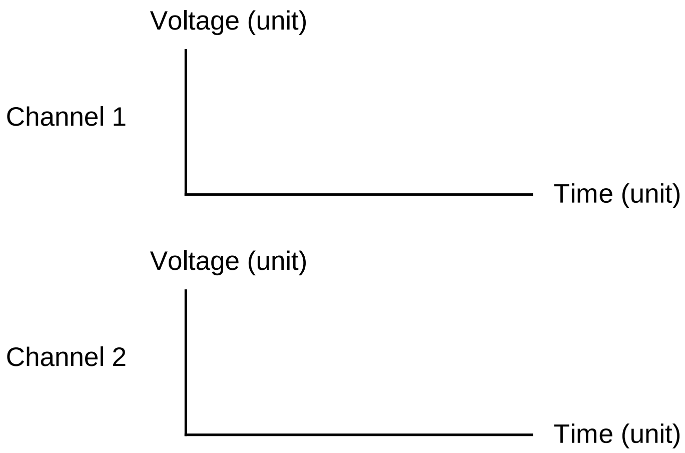
Figure 3

5.1.6 Set input A alternately to 0 and 1, and observe the effect on the output. Sketch the waveforms displayed by both CHANNEL 1 and 2 of the oscilloscope for each setting of A using Timing Diagram 1. Pause and think: How do you distinguish channel 1's waveform from channel 2's waveform on the display?

Both waveforms observed on channel 1 and channel 2 should be drawn on the same page with the same time scale for ease of comparison. **Both X and Y axes should also be clearly labeled with appropriate units (e.g. volt and ms) and values.** Pause and think: How do you read the voltage and time values from the display?

5.1.7 How is the output related to input B when A is 1? Is it the same when A is 0? Are you able to explain this observation from the result recorded in Table 2?

5.1.8 Pause and think: How can the oscilloscope be used to help you adjust the square wave frequency at input B to 1kHz? Assuming that the oscilloscope is more accurate than the labels on the function generator knobs.



Timing Diagram 1

5.2 The Exclusive-OR Gate

5.2.1 Mount a 74LS86 IC on the circuit board and connect its power supply and ground. Wire up an XOR gate of the IC as shown in Figure 4. If you do not get a stable display on the oscilloscope, press the AUTO SET button.

Notice that the setup is essentially the same as 5.1.5 except that an XOR logic gate is used here instead of the NAND. You may use an **IC puller** to carefully extract the 74LS00 IC and replace it with the 74LS86 IC without changing the rest of the connections. This can be done because the two IC components have the same pin layout.

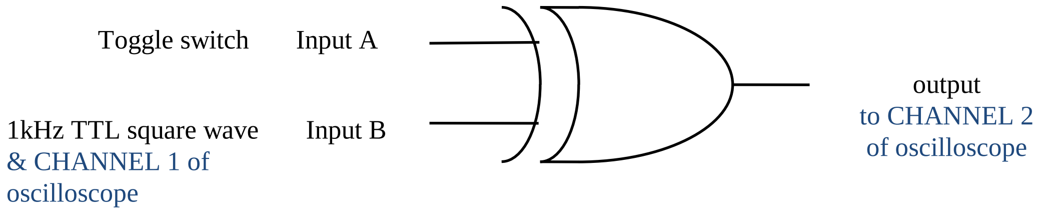
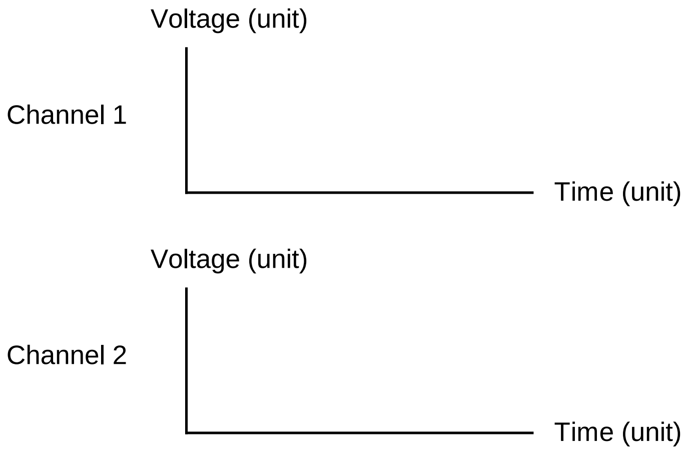


Figure 4

5.2.2 Set input A to logic 0, what is the output of the XOR gate? Next set input A to logic 1, what is the output of the XOR gate? Record your observations using Timing diagram 2. **Make sure that the axes are clearly labeled with the appropriate units and values.**



Timing Diagram 2

5.2.3 How is the output related to input B when A is 1? Is it the same when A is 0? Are you able to explain this observation from the truth table of an XOR gate? Pause and think: How is this observation similar or different from that in 5.1.7?

5.3 **Implementation of a Full Adder**

5.3.1 Figure 5 shows a full adder circuit based on the following logic expressions:

Sum = A ⊕ B ⊕ Cin

Cout = A•B + B•Cin + A•Cin

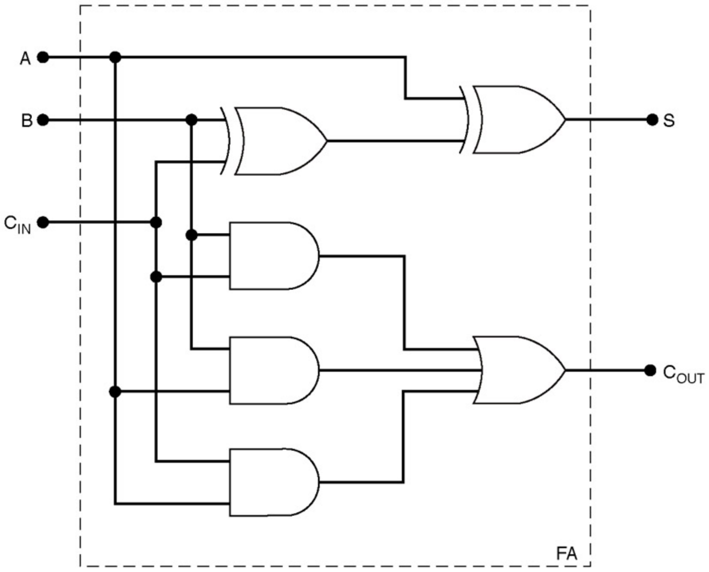


Figure 5: A Full Adder circuit (do not implement this)

With some Boolean expression manipulation (refer to lecture slide 4.8), one can show that

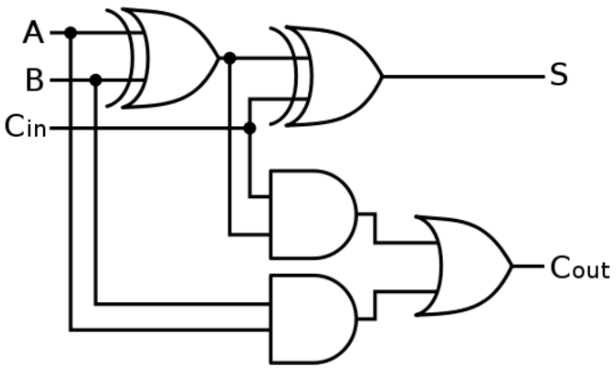
Cout = A•B + Cin•(A ⊕ B)

Figure 6 shows the corresponding circuit which is simpler to implement with fewer connections.

Using the available logic ICs (XOR, AND, OR), implement the full adder circuit in **Figure 6**. Sketch the circuit connection diagram with clear labels and pin numbers. Connect up the circuit and verify its function with a truth table.

It is not advisable to connect up the entire circuit in one go as you may get confused by the wires. You should implement the sum output first since it is simpler. After verifying that it's correct, you may then proceed to implement the Cout output. Note that every IC needs a pair of Vcc and ground.

Inputs connected to toggle switches



Outputs connected to LEDs

Figure 6: A Full Adder circuit (simpler implementation)

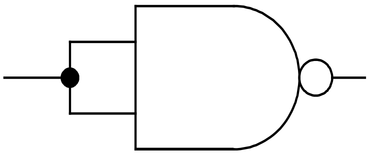
5.3.2 Pause and think:
- Can we say $(A+B) = (A \oplus B)$ based on 5.3.1? Why?
- How do you implement a 3-input OR with two 2-input OR gates?
- How do you implement a 3-input NAND with 2-input NAND gates?
- How many full adders do you need to add a pair of 4-bit numbers?

5.4 The Universality of NAND Gate (Optional)

Complete this part only if time permits and you have understood the earlier parts of this experiment.

5.4.1 **NOT from NAND**: Connect the circuit shown in Figure 6. Connect a toggle switch to the circuit input and an LED to the output. Verify that the circuit is an inverter by setting the input switch to 0 and then 1 and observing the output LED.

Input connected to toggle switch

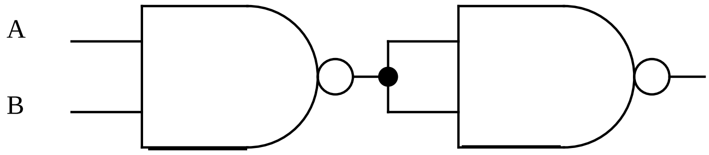


Output connected to LED

Figure 6. NAND gate equivalent circuit of an inverter

5.4.2 **AND from NAND**: Connect the circuit in Figure 7. Connect toggle switches to the circuit inputs. Connect an LED to the output. Show with a truth table that the circuit performs the AND function.

Toggle switches



Output to LED

Figure 7. NAND gate equivalent circuit of an AND gate.

5.4.3 **OR from NAND**: Connect the circuit in Figure 8. Connect toggle switches to the circuit inputs. Connect an LED to the output. Show with a truth table that the circuit performs the OR function.

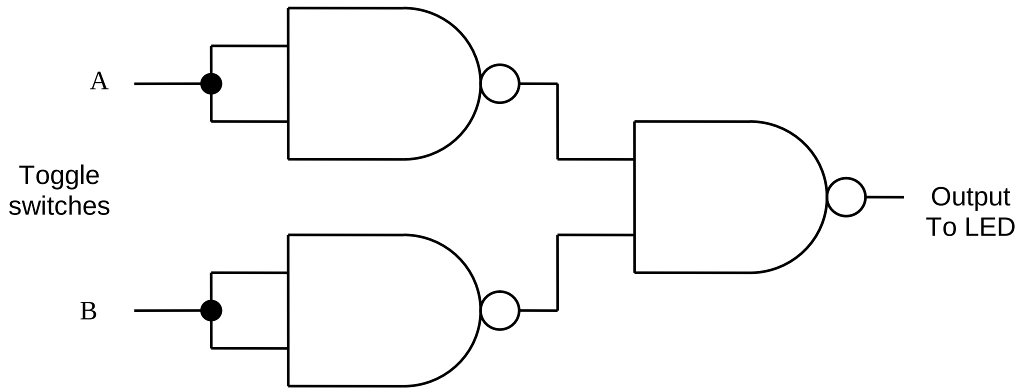


Figure 8. NAND gate equivalent circuit of an OR gate.

5.4.4 Pause and think: How would you implement a 2-input XOR gate with four 2-input NAND gates only?

6. PREPARATION AND OBSERVATIONS

Maintain a proper record of your preparation (e.g. circuit connection diagrams) and observations (e.g. truth tables and waveforms) on paper throughout the experiment. You will be able to refer to this record during the lab quiz.

7. LAB QUIZ

There will be a 10-minute written lab quiz at the end of this experiment. The quiz will include (but not limited to) concepts, components, equipment, measurements and observations covered in this experiment.

8. REFERENCES

8.1 CE/CZ1005 Digital Logic Supplementary Laboratory Manual – Wiring & Troubleshooting Digital Circuits, Digital Logic Couse Site, NTULearn.

8.2 Lab 1 equipment guide.pdf, Digital Logic Couse Site, NTULearn.

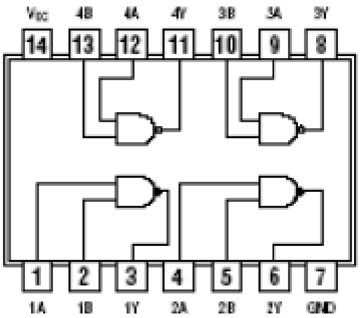
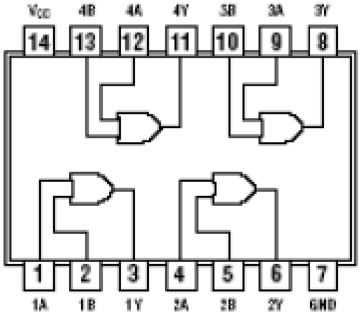
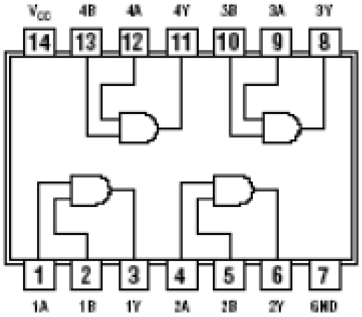
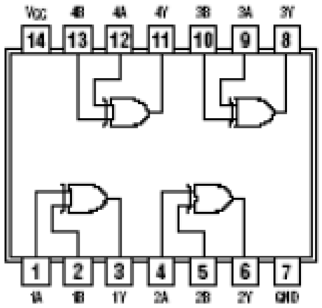
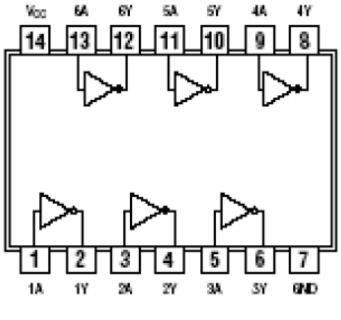
8.3 Lab 1 circuit connection guide.pdf, Digital Logic Couse Site, NTULearn.

8.4 Digital Design Principles and Practices, Ed. 4, John F Wakerly, Prentice Hall, 2007.

8.5 Fundamentals of Digital Logic with Verilog Design, 2nd Ed., by Stephen Brown and Zvonko Vranesic, McGraw Hill, 2008.

Appendix

Pin assignments (from TI Digital Logic Pocket Data Book 2003)

7400: quad 2-input NAND  V_{CC} 4B 4A 4Y 3B 3A 3Y 14 13 12 11 10 9 8 1 2 3 4 5 6 7 1A 1B 1Y 2A 2B 2Y GND	7432: quad 2-input OR  V_{CC} 4B 4A 4Y 3B 3A 3Y 14 13 12 11 10 9 8 1 2 3 4 5 6 7 1A 1B 1Y 2A 2B 2Y GND
7408: quad 2-input AND  V_{CC} 4B 4A 4Y 3B 3A 3Y 14 13 12 11 10 9 8 1 2 3 4 5 6 7 1A 1B 1Y 2A 2B 2Y GND	7486: quad 2-input XOR  V_{CC} 4B 4A 4Y 3B 3A 3Y 14 13 12 11 10 9 8 1 2 3 4 5 6 7 1A 1B 1Y 2A 2B 2Y GND
7404: hex inverter  V_{CC} 6A 6Y 5A 5Y 4A 4Y 14 13 12 11 10 9 8 1 2 3 4 5 6 7 1A 1Y 2A 2Y 3A 3Y GND	