

Istanbul Technical University
Department of Computer Engineering

BLG 242E – Logic Circuits Laboratory

Experiments Booklet

Spring 2019*

Version 1.9

* Version 1.8.0 was prepared by Gökhan Seçinti and Atakan Aral in Spring 2016
Version 1.8.1 was revised by Müge Erel-Özcevik in Spring 2017
Version 1.9 was prepared by Müge Erel-Özcevik and Kadir Özlem in Spring 2019

Contents

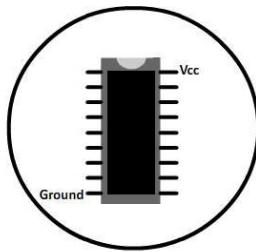
| | |
|--|-----------|
| Lab Guidelines | 1 |
| Introduction | 3 |
| CADET | 3 |
| Components | 4 |
| 1 Learning CADET | 11 |
| 1.1 Introduction | 11 |
| 1.2 Preliminary | 11 |
| 1.3 Equipments and Integrated Circuits (ICs) | 11 |
| 1.4 Experiment | 11 |
| 1.5 Report | 12 |
| 2 Boolean Algebra | 13 |
| 2.1 Introduction | 13 |
| 2.2 Preliminary | 13 |
| 2.3 Equipments and Integrated Circuits (ICs) | 13 |
| 2.4 Experiment | 14 |
| 2.5 Report | 14 |
| 3 Combinational Logic Circuits | 15 |
| 3.1 Introduction | 15 |
| 3.2 Preliminary | 15 |
| 3.3 Equipments and Integrated Circuits (ICs) | 15 |
| 3.4 Experiment | 16 |
| 3.5 Report | 16 |
| 4 Binary Arithmetic | 17 |
| 4.1 Introduction | 17 |
| 4.2 Preliminary | 17 |
| 4.3 Equipments and Integrated Circuits (ICs) | 17 |
| 4.4 Experiment | 18 |
| 4.5 Report | 19 |
| 5 TTL and CMOS Characteristics | 21 |
| 5.1 Introduction | 21 |
| 5.2 Preliminary | 21 |
| 5.3 Equipments and Integrated Circuits (ICs) | 21 |
| 5.4 Experiment | 21 |
| 5.5 Report | 27 |

VI Contents

| | |
|---|----|
| 6 Data Bus Implementation in Digital Systems | 29 |
| 6.1 Introduction | 29 |
| 6.2 Basic Principles | 29 |
| 6.3 Preliminary | 32 |
| 6.4 Equipments and Integrated Circuits (ICs) | 32 |
| 6.5 Experiment | 32 |
| 6.6 Report | 34 |
| 7 Latches and Flip-flops | 35 |
| 7.1 Introduction | 35 |
| 7.2 Preliminary | 35 |
| 7.3 Equipments and Integrated Circuits (ICs) | 35 |
| 7.4 Experiment | 35 |
| 7.5 Report | 36 |
| 8 Sequential Logic Circuits | 37 |
| 8.1 Introduction | 37 |
| 8.2 Preliminary | 37 |
| 8.3 Equipments and Integrated Circuits (ICs) | 37 |
| 8.4 Experiment | 37 |
| 8.5 Report | 38 |
| 9 Design and Implementation of Serial Arithmetic Logic Unit | 39 |
| 9.1 Introduction | 39 |
| 9.2 Preliminary | 40 |
| 9.3 Equipments and Integrated Circuits (ICs) | 40 |
| 9.4 Experiment | 40 |
| 9.5 Report | 41 |
| A Data Sheets | 43 |
| A.1 7400 - Quadruple 2-input Positive-NAND Gates | 43 |
| A.2 7402 - Quad 2-input Positive-NOR Gates | 43 |
| A.3 7404 - Hex Inverters | 44 |
| A.4 7405 - Hex Inverters with Open-Collector Outputs | 44 |
| A.5 7408 - Quadruple 2-input Positive-AND Gates | 44 |
| A.6 7410 - Triple 3-input NAND Gates | 45 |
| A.7 7411 - Triple 3-input AND Gates | 45 |
| A.8 7427 - Triple 3-input NOR Gates | 45 |
| A.9 7432 - Quadruple 2-input Positive-OR Gates | 46 |
| A.10 7475 - Quadruple Bistable Latches | 46 |
| A.11 7483 - 4-Bit Binary Full Adder with Fast Carry | 46 |
| A.12 7486 - Quad 2-Input Exclusive Or Gate | 47 |
| A.13 74138 - 3-to-8 decoder/demultiplexer | 47 |
| A.14 74151 - 8-Input Multiplexer | 47 |
| A.15 74161 - BCD Decade Counter / 4-Bit Binary Counter | 48 |
| A.16 74165 - 8-Bit Parallel-Load Shift Register | 48 |
| A.17 74174 - Hex D-Type Flip-Flop with Reset | 48 |
| A.18 74181 - 4-Bit Arithmetic Logic Unit | 49 |
| A.19 74241 - Octal 3-State Buffer/ Line Driver/ Line Receiver | 49 |
| A.20 4011 - Quad 2-Input NAND Gate | 50 |

Lab Guidelines

- Make sure that integrated circuits (ICs) are inserted properly. Integrated circuits need power in order to open and close the transistors that implement the gates they contain. Generally, the pin in the upper-right corner is called VCC as seen in figure below. This is where you connect the source voltage. The pin in the lower-left corner is where ground voltage is applied. You can refer to Appendix A for more information. Wrong connections of ICs may harm the electrical components!



An Example Integrated Circuit

- You can reduce the risk of harming chips by **GENTLY** holding both sides when removing.
- Make sure that you place the ICs back to where you took after you are done with them. Misplaced ICs may cause troubles in the following sessions. They may be burned and become dysfunctional.
- **Never** take any ICs from other groups and **do NOT ALLOW** anyone take ICs from your group. Integrated circuits' packets will be checked after each session.
- If you think an IC is malfunctioned, do the followings before you call any lab assistant:
 1. Try the IC by placing into another place on the CADET
 2. Check VCC and ground connections
 3. Make sure that VCC and ground sockets are fed appropriately by the help of the LEDs on the right hand side of the CADET. You may want to use Voltmeter as well (VCC: 5V, ground: 0V).
 4. Connect the necessary sockets of 8 switches into the input pins of the IC.
 5. Make sure that the voltages of input pins of the IC are provided correctly by the help of the LEDs like in the 3rd step.

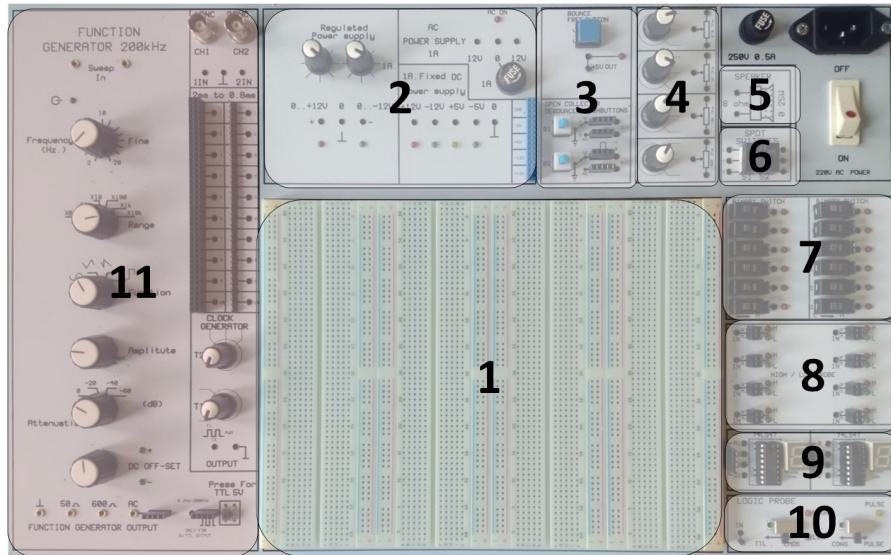
2 Lab Guidelines

6. Connect the output(s) of the IC to the LED(s).
 7. Observe the behavior of the LED(s) connected to output(s) of the IC for both states of the switches (0 and 1).
 8. After you performed all steps, if you still think the IC is not working properly, then you can politely ask the lab assistant to replace it with the new one.
- If you draw the circuit to a paper or via a program, such as Logisim, before the experiment, you are expected to face less problems during the experiment.
 - Try to use the cables with the same color for the same purposes. For example, blue cables can be used for the cables coming from the switches, and orange cables for the ones going to the LEDs. Furthermore, we suggest that red cables can be used for 5V and black cables for 0V. In this way, debugging of the circuit and correcting problems become easier.
 - If your circuit does not output properly, you may need to check every input and output with the help of a long cable. You can do it by connecting one side of the long cable to the socket you want to check and another side to a LED.

Introduction

CADET

The digital circuits you will design for the BLG 242E labs will be built on the CADET (Complete Analog/Digital Electronics Trainer) stations. Each CADET station contains a central bread-boarding area on which to lay out your circuit, as well as various built-in circuit accessories, such as multiple power supplies, a simple function generator, and logic LED indicators. The basic layout of the CADET station is shown below. Components of the CADET is labeled labeled 1 through 11 in the picture. We will briefly describe the functions that you will most likely need to be familiar with for the labs¹.



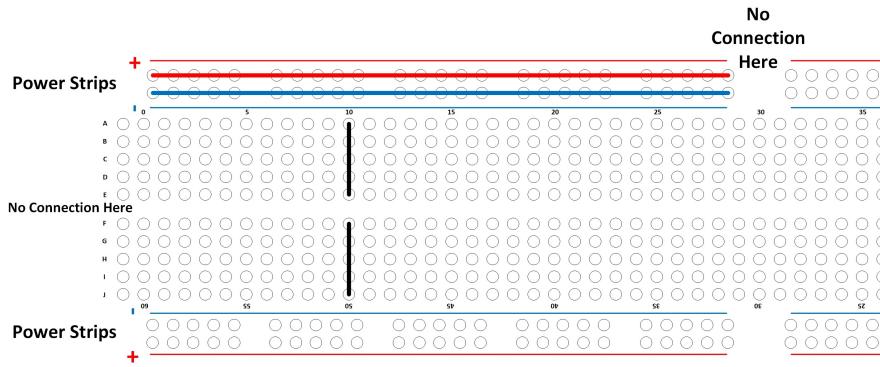
Layout of the CADET.

¹ Main reference for the CADET description can be accessed from the link below. However, please note that the referenced document is intended for a slightly different version of CADET and the explanations are modified accordingly.
<http://web.stanford.edu/class/ee121/handouts/lab2.pdf>

Components

1 Breadboard

There is a large white area in the center of the CADET full of holes called tie points or contact points. This is the breadboarding area as shown in the figure below. It is not connected to any circuitry beneath the surface of the trainer. This breadboarding area is where you will place your components (gates). There are actually four separate breadboards in the center of the trainer. Each breadboard is arranged as follows: vertical columns of five holes are the same point electrically (they are connected to each other), and they are shown with black solid lines in the figure. No column of tie points is connected to any other column, however (no horizontal connectivity). Moreover; horizontal rows of tie points named as power strips, are the same points electrically. The outer power strips are usually used to supply voltage and ground to the circuit while the inner vertical columns are used for signal connections². Note that the power strips above and below the center mounting screw are not connected to each other.



Interconnection of the holes.

This structure provides a convenient means of constructing experimental circuits. The breadboard has two arrays of holes separated by a long (blank) channel. An integrated circuit chip is placed over the central channel so its pins make connection with the first row of holes on either side of the channel. This leaves the four remaining holes in each column available to make connections to other points.

2 Power Supply

At the top of the breadboard area, the CADET features three different power supplies: regulated, fixed DC and AC. In the regulated area; the positive voltage (12V) can be adjusted from 0 V to +12 V and from 0 V to -12 V. In AC power supply, the positive 12V can be offered by using three outputs. In this lecture, you will use DC power supply with 5 V outputs. Fixed 5V or 12V are obtained by related outputs and ground. Do not forget to carry 0 V and 5 V to breadboard during experiments.

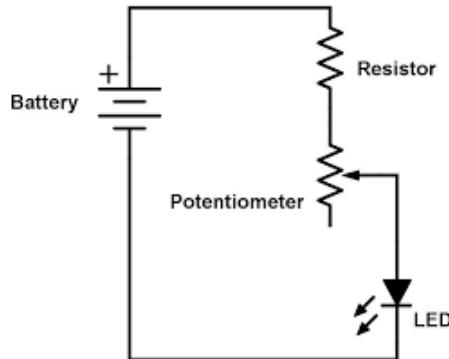
² <http://www.dejazzer.com/ece311/labs/station.html>

3 Debounced Pushbuttons

There are two pushbuttons on the top of the CADET. These are labeled open collector debounced pushbuttons because they consist of the physical mechanical switch with additional circuitry to eliminate the multiple switch closures normally found when operating mechanical switches. That is, most switch contacts actually bounce very briefly when closed. Even though this period of time is brief (a few milliseconds), digital circuitry is fast enough to falsely interpret this as several closures rather than just one. Thus the need to electronically debounce these switches. Each switch has eight tie points of two different types. Four of the points are marked by the letters NC meaning normally closed. These points are connected to ground when that pushbutton is in its unpressed position and become open (disconnected) when the button is pressed. Four of these points are marked by the letters NO meaning normally open (the exact opposite of normally closed). To wire the pushbutton as an active-low switch suitable for interfacing with digital circuitry. When the button is not pressed, the switch provides a digital high close to +5 V (there is a very small voltage drop across the pullup resistor). When the button is pressed, the switch output is a digital low (ground). Note that we can't bypass the pull-up resistor and connect the switch directly to +5 V; doing so would cause you to short power and ground when the switch is pressed and lead to a blown fuse.

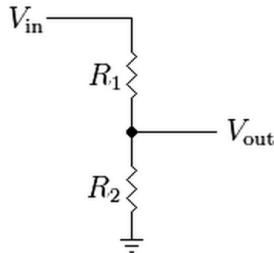
4 Potentiometers

Four potentiometers are provided on the CADET. The resistance values chosen (470, 1K, 10K, and 100K ohms) may be used in common circuit applications such as volume controls. All leads for both potentiometers are available and uncommitted.



An example circuit where the potentiometer is used.

An example circuit is given above. Here, the potentiometer is used as a voltage divider. Adjusting the resistance enables to change the voltage of the LED, and therefore, this changes its brightness. Voltage dividers are commonly used in electronic circuits. The simpler schematic of the voltage divider is given in the figure below.



$$V_{out} = V_{in} * \frac{R2}{R1 + R2} \quad (0.1)$$

Voltage Divider Circuit.

5 Loudspeaker

6 SPDT Switches

These two SPDT (Single Pole Double Throw) switches are intended for sending two alternative signals or voltages to an output. Two different voltages are given from pins 1 and 3. The output voltage is collected from pin 2. When a SPDT switch is in its down position, pin 2 is connected to pin 3. When it is in its up position, it is connected to pin 1.

7 Logic Switches

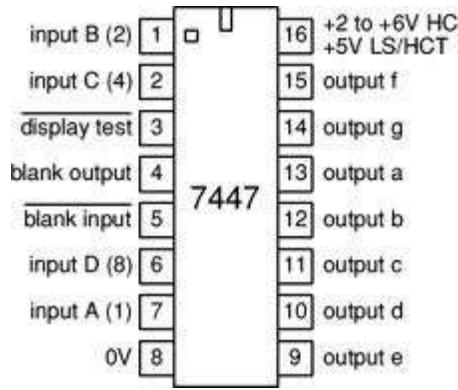
In the right part of the trainer are twelve sliding logic switches. These switches are not debounced. Each has two tie points on the connector block above the switches. When a logic switch is in its down position, it places its tie point on the connector block at logic 0 or ground. When a logic switch is in the up position, it places its tie point at logic 1. Just exactly what voltage logic 1 is depends on several factors. The output can be taken from both two pins near the switch.

8 Logic Monitor

On the right side of the trainer are eight LED logic indicators. These LEDs will often be used indicate the outputs of the circuits you build. They are also useful for debugging purposes. A small connector block allows these LEDs to be connected to the rest of the circuit being constructed. Each IN (input) pin has red and green LEDs. The red LEDs indicate a high or "1" logic level. The green LEDs indicate a low or "0" logic level. Moreover; if any led is not enabled, all are in high impedance.

9 Displays

The type of displays is the seven segment display which has four input pins labeled A, B, C and D. There are 2 seven segments displays where input pins are interpreted as binary coded decimal (BCD). In the CADET; the Integrated Circuit (IC) 7447 is used for decimal display. The pins can be seen in the figure below. Here; D is our MSB (Most Significant Bit) and A is our LSB (Least Significant Bit). Therefore; to display 39 by using both displays, the inputs should be as ($D_1C_1B_1A_1D_2C_2B_2A_2$) = (00111001).

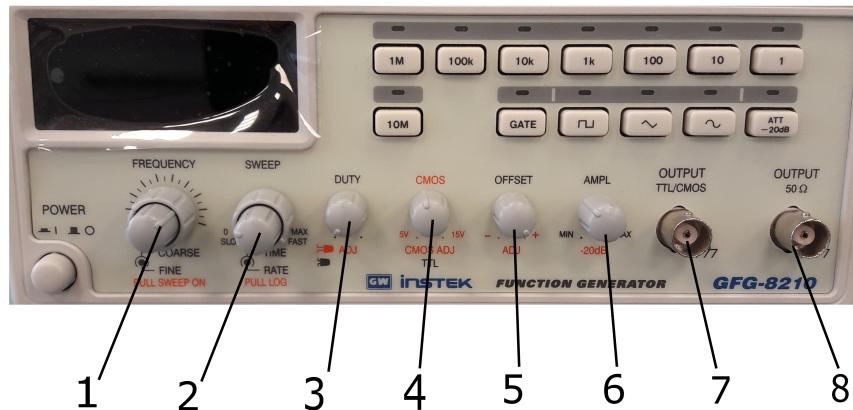


Pin Configuration of the 7447 IC.

10 Logic Probe

11 Function Generator

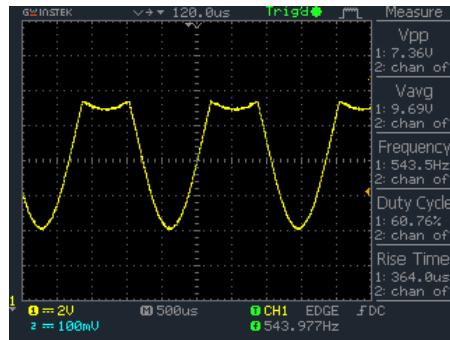
On the left side of the trainer is the function generator which can produce square, sine, and triangle waves of variable frequency and amplitude. It can also produce a TTL-compatible square wave suitable for use with digital circuitry via the outputs below. We will describe only this last feature of the function generator here. The speed or frequency of the TTL output square wave is adjusted by three other controls: frequency, range, amplitude, attenuation. Here, the frequency of your waveform is the number of Hertz (as indicated by the switch at the top left) found by multiplying the frequency slide adjustment by the decade range selection switch. For example, if the top left switch is in the KHz position, the sliding frequency adjustment is at the bottom (0.1), and the decade range selector is in the middle (x10) position, then the output is 0.1×10 KHz, or 1KHz. The lowest frequency available is 0.1 Hz, and the highest is 100 KHz. Note that the TTL output is not affected by the setting of the switch with the sine, triangular, and square wave symbols, nor by the amplitude sliding control (marked AMP). These are used by the other features of the function generator.



Function Generator.

In this lecture, the function generators (not being on CADETs) will be used as shown in the figure above. It generates square, sine, and triangle waves of variable frequency and amplitude. The Sweep knob (2) is used to start the auto sweep operation; the upper frequency limit is determined by the knob position. The Duty knob (3) enables to adjust the duty cycle of the waveform. The frequency knob (1) changes the waveform and the buttons on the top row are multipliers for the frequency.

There are two outputs can be taken at the bottom left: TTL/CMOS (7) and 50 Ohm outputs (8). In the TTL/CMOS mode the output is always a square wave. Offset and Amplitude knobs are not available here. When the CMOS knob (4) is pulled, it switches from TTL to CMOS. TTL output gives a 5V square wave while the CMOS output can be adjusted between 0.6V and 15V. Moreover; in 50 ohm mode, the output can be selected in one of the three waveforms: square, triangular and sinus via related knobs. Here, The amplitude knob (6) can be used to change the amplitude between 2.4V and 22V. When the knob is pulled, it will adjust a -20dB output. The ATT button can also be used for giving an additional -20dB to the output. The output can be generated as a DC offset using the offset knob (5). An example output of the wave clippings is observed as seen in below.



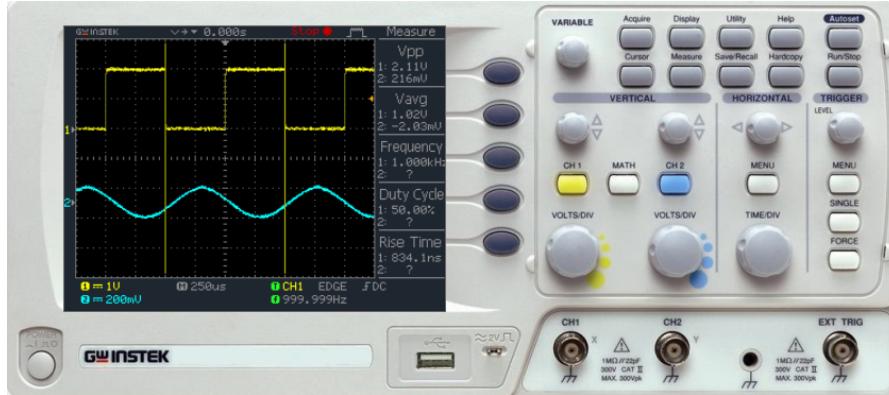
Wave clippings adjusting by DC offset.

The outputs should be carried function generator to CADET via crocodile cable as shown in below.



Crocodile Cable.

12 OSCILLOSCOPE



An example usage of the oscilloscope.

The oscilloscope shown in the figure above, is used for observing received waveforms. It has two channels: CH1 and CH2. Each output is connected to one, and the CH1 and/or CH2 is selected by using related button of the oscilloscope. To obtain output from CADET; the probe cable is used, it is exemplified in the figure below.



Probe cable.

There are several facilities on the oscilloscope. Run/Stop button is used to take a snapshot of the waves in a certain time. In order to refreshing the outputs on screen, the Autoset button is used. In the screen, the position knobs under the horizontal and vertical tabs are used to align the waveforms manually. Here; Vertical axis shows voltage and the horizontal axis shows time. Moreover; to observe the frequency and amplitude (Vpp), the measure button can be used. Moreover; for the same purpose, the square units on the screen is also helpful to calculate these metrics of the waves. For example; as an example given in the figure above, the voltage is observed at the bottom of the screen (1=1V and 2=200mV). These values show the voltage change of the wave in one square unit, vertically. The number of vertical squares of the wave is multiplied by the scale to obtain the wave's peak to peak voltage; whereas, the horizontal squares of a single wave is multiplied by the time scale to obtain frequency.

1

Learning CADET

1.1 Introduction

The aim of this experiment is to learn each CADET components, the function generator, and the oscilloscope.

1.2 Preliminary

- Revise CADET components, the function generator, and the oscilloscope given in the Introduction section of booklet.
- Review your electronic knowledge such as amplitude, frequency, period, peak to peak voltage, root mean square value.

1.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- Function generator
- Oscilloscope
- 74000 series ICs
 - 74xx¹04 - Hex Inverter

Fundamental informations (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

1.4 Experiment

1.4.1 Experiment - Part 1

Design general structure of CADET using power supply (0V-5V). You should use power-strips to distribute $V_{cc}(5V)$ and $Gnd(0V)$ through the CADET by using tie points. Show them in two logic monitors of CADET.

1.4.2 Experiment - Part 2

- A theorem is given as: $(a = a')$. Implement this function taking an input from a switch by using 7404 Hex Inverter and validate your implementation in two logic monitors for both input and output of the inverter.
- Repeat the same implementation by using SPDT switch.

1.4.3 Experiment - Part 3

- By using voltmeter, observe V_{cc} and Gnd voltages.
- By using potentiometer, build a resistance in 8 Kohm.

1.4.4 Experiment - Part 4

- Show "27" in two seven segment display by giving an input using 8 switches.

1.4.5 Experiment - Part 5

- By giving TTL input from function generator, show the output into logic monitor. Observe the changes on output with different frequencies.
- Use oscilloscope to observe input on different following frequencies.
 - 27 kHz TTL
 - 5 V (Vpp), 1992 Hz CMOS
 - 1 V (Vmax), 500 Hz CMOS
 - 3.14 V (Vpp), 0.6 MHz triangular wave
 - 1.2 V (Vmax), sine wave with 2.5ms period
 - 0.7 V (Vpp), 0.001 GHz square wave
 - 1250 mV (Vpp), 45 Hz square wave

1.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Theoretical information used in the experiment
- For the enrichment of the report, materials such as tables, photos, scheme, and diagram
- What you learned in the experiment

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

2

Boolean Algebra

2.1 Introduction

The aim of this experiment is to recall the axioms and theorems of Boolean algebra and validate these axioms and theorems by using physical components in an experimental environment.

2.2 Preliminary

- Revise the axioms and theorems of Boolean algebra.
- Prove the given equalities below by using the axioms of Boolean algebra.
 - $a + a \cdot b = a$
 - $(a + b) \cdot (a + b') = a$
- Determine and prove the duals of the equalities defined above.
- Calculate the complementary expression (F') for the function F which is defined as follows ($F = a \cdot b + a' \cdot c$) by using De Morgan theorem and draw the logic circuit for both expressions (F and F').
- Simplify given logical function and draw the logic circuit.
 - $F(a, b, c, d) = \cup_1(1, 2, 5, 6, 9, 10, 13, 14)$

2.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
 - 74xx¹04 - Hex Inverters
 - 74xx¹08 - Quadruple 2-input Positive AND Gates
 - 74xx¹32 - Quadruple 2-input Positive OR Gates

Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

¹ "xx" has been used as a wildcard in this document. Instead of "xx"; S, LS, C, HC or HCT could be written on the ICs . These letters specify the inner structure of the logic gates. Although their inner structures may differ, their logic functionalities are the same.

2.4 Experiment

2.4.1 Experiment - Part 1

Design and implement the logic circuits for the given expressions below by using the necessary gates.

- $F_1(a, b) = a + a \cdot b$
- $F_2(a, b) = (a + b) \cdot (a + b')$

You should use the switches on the CADET as the inputs for the expressions and you should also use the LEDs to observe the output of the circuit you have implemented. Finally, after the implementation phase, validate correctness of your design.

2.4.2 Experiment - Part 2

A theorem is given as: $(a + a \cdot b = a)$. First, determine the dual of the given theorem and then, implement the functions for both sides of the dual theorem by using logic gates. Validate the truth of the theorem by comparing the changes in the outputs.

2.4.3 Experiment - Part 3

$F_3(a, b, c) = a \cdot b + a' \cdot c$ is given. First, determine the complement of the given function (F_3). Then, implement the circuit which realizes the complementary function (F'_3) . Validate your implementation by using the truth table.

2.4.4 Experiment - Part 4

A basic logical function (F_4) is defined as follows.

$$F_4(a, b, c, d) = \cup_1(1, 2, 5, 6, 9, 10, 13, 14)$$

First, simplify given logical function and implement the simplified expression using logic gates. Validate your circuit by observing the outputs for each possible input.

2.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams of the expressions which were implemented during this experiment.
- Function tables (or truth tables) of the implemented expressions.

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

3

Combinational Logic Circuits

3.1 Introduction

The aim in this experiment is to find the expression with the lowest cost for combinational logic circuits and implement them.

3.2 Preliminary

1. Find all prime implicants of the function F below.

- a) using Karnaugh diagram
- b) using Quine-McCluskey method

$$F(a, b, c, d) = \cup_1(0, 3, 5, 7, 11, 12, 13) + \cup_\phi(1, 8, 15)$$

Also create the prime implicant chart of the function F and find the expression with the lowest cost. Cost criteria is 2 units for each variable and 1 unit for each complement. Draw the lowest cost expression using AND, OR, and NOT gates.

2. Design and draw the same function using only NAND and NOT gates.
3. Design and draw the same function using a single 8:1 multiplexer and NOT gates.
4. Design and draw the function below using a single 3:8 decoder, OR, and NOT gates.

$$F_1(a, b, c) = a' \cdot c' + b \cdot c$$

$$F_2(a, b, c) = a' \cdot b' \cdot c' + a \cdot b$$

3.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
 - 74xx00 - Quadruple 2-input Positive NAND Gates
 - 74xx04 - Hex Inverters
 - 74xx08 - Quadruple 2-input Positive AND Gates
 - 74xx10 - Triple 3-input Positive NAND Gates
 - 74xx11 - Triple 3-input Positive AND Gates
 - 74xx27 - Triple 3-input Positive NOR Gates
 - 74xx32 - Quadruple 2-input Positive OR Gates

- 74xx138 - 3:8 Decoder
- 74xx151 - 8:1 Multiplexer

Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

3.4 Experiment

3.4.1 Experiment - Part 1

Build the circuit you have designed in Preliminary Question 1 using the ICs. Fill up a truth table to evaluate your implementation.

3.4.2 Experiment - Part 2

Build the circuit you have designed in Preliminary Question 2 using the ICs. Fill up a truth table to evaluate your implementation.

3.4.3 Experiment - Part 3

Build the circuit you have designed in Preliminary Question 3 using the ICs. Fill up a truth table to evaluate your implementation.

3.4.4 Experiment - Part 4

Build the circuit you have designed in Preliminary Question 4¹ using the ICs. Fill up a truth table to evaluate your implementation.

3.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams of the expressions which were implemented during this experiment.
- Function tables (or truth tables) of the implemented expressions.
- Discuss the output of your circuits for undetermined (ϕ) inputs. Why did you get these outputs?

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

¹ Notice that, the given decoder IC (**74xx138**) in the experiment works in **negative logic** mode. Thus, you need to re-consider your design for the output logic accordingly.

Binary Arithmetic

4.1 Introduction

In this experiment, you will implement logic circuits for arithmetic operations on signed and unsigned binary numbers. In addition, you will use the Arithmetic Logic Unit (ALU) integrated circuit to carry out basic operations.

4.2 Preliminary

- Recall signed and unsigned addition for binary numbers in 2s complement notation.
- Recall signed and unsigned subtraction for binary numbers in 2s complement notation.
- Recall what carry, borrow and overflow mean, when they occur and how are they interpreted.
- Study the function table and pin descriptions of 74xx181, 4-Bit Arithmetic Logic Unit.

4.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

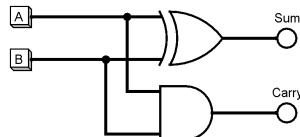
- C.A.D.E.T. (**C**omplete **A**nalogue **D**igital **E**lectronic **T**rainer)
- 74000 series ICs
 - 74xx08 - Quadruple 2-input Positive AND Gates
 - 74xx32 - Quadruple 2-input Positive OR Gates
 - 74xx83 - 4-bit Binary Full Adder
 - 74xx86 - Quadruple 2-input Positive Exclusive Or (XOR) Gates
 - 74xx174 - Hex D-Type Flip-Flops
 - 74xx181 - 4-Bit Arithmetic Logic Unit

Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

4.4 Experiment

4.4.1 Experiment - Part 1

A half adder circuit is given below. Implement and test the circuit to draw the truth table of the half adder. Use switches for inputs and LEDs for outputs.



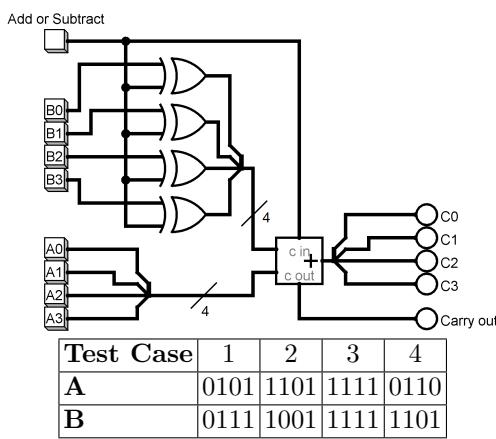
Attention: After you have completed this part, do not disassemble the circuit since you will reuse it in the next part.

4.4.2 Experiment - Part 2

Attach one more input (for C_{in}) and additional gates to your circuit in Part 1 to obtain a full adder. Test the circuit and draw the truth table.

4.4.3 Experiment - Part 3

Using a 4-bit full adder (74xx83) implement the circuit below.



For each test cases above, carry out the following tests.

- Calculate the result of $A + B$. Interpret inputs as **unsigned**. Draw a table with the following columns and add your results.

| | | | | |
|---|---|-------|------------------|-------------------|
| A | B | Carry | Result in Binary | Result in Decimal |
|---|---|-------|------------------|-------------------|
- Calculate the result of $A + B$. Interpret inputs as **signed**. Draw a table with the following columns and add your results.

| | | | | | |
|---|---|----------|-------------|------------------|-------------------|
| A | B | Overflow | Result sign | Result in Binary | Result in Decimal |
|---|---|----------|-------------|------------------|-------------------|
- Calculate the result of $A - B$. Interpret inputs as **unsigned**. Draw a table with the following columns and add your results.

| | | | | |
|---|---|--------|------------------|-------------------|
| A | B | Borrow | Result in Binary | Result in Decimal |
|---|---|--------|------------------|-------------------|
- Calculate the result of $A - B$. Interpret inputs as **signed**. Draw a table with the following columns and add your results.

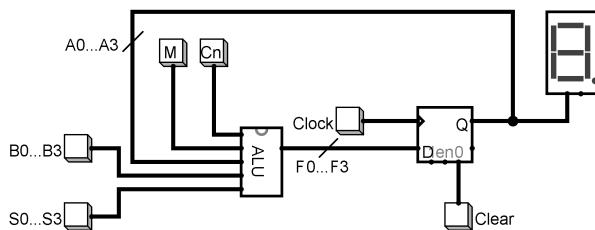
| | | | | | |
|---|---|----------|-------------|------------------|-------------------|
| A | B | Overflow | Result sign | Result in Binary | Result in Decimal |
|---|---|----------|-------------|------------------|-------------------|

4.4.4 Experiment - Part 4

Using a 4-Bit Arithmetic Logic Unit (74xx181) and D type flip-flops (74xx174) implement the circuit below. Use ttl for the clock input to prevent multiple entries. Also, use the SPDT switches for clear and M , and give ground (0 V) to C_n .

Interpreting ALU (74xx181) datasheet

In the datasheet of ALU, arithmetic subtraction and addition are defined as *minus* and *plus* operations. Here; $A + B$ and $A.B$ shows logic OR and AND operations. Please also be careful on that F outputs of ALU are in negative logic. Please use additional inverter for Q outputs of D type flip-flop before observe them in leds.



Carry out the following operations one by one. Add a row to the table for each operation.

- $A \leftarrow 0$
- $A \leftarrow A + 1$
- $A \leftarrow 5$
- $A \leftarrow A + 3$
- $A \leftarrow A + 2$
- $A \leftarrow A - 4$
- $A \leftarrow A \oplus 6$
- $A \leftarrow A \times 2$

| Inputs | | Outputs | |
|--------|-----|-------------------|---------------------------------|
| A | B | $S_3 S_2 S_1 S_0$ | $M C_n F_3 F_2 F_1 F_0 C_{n+4}$ |

Table 4.1: Input & Output Details

4.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams of the expressions which were implemented during this experiment.
- Your results as tables for each part of the experiment.

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

TTL and CMOS Characteristics

5.1 Introduction

Static and dynamic characteristics of TTL and CMOS gates are going to be examined in this experiment. Furthermore, similarities and differences among these semi-conductor technologies (TTL & CMOS) are going to be interpreted. The behavioral characteristics of a logic gate could be defined by its voltage and current values. These characteristics of TTL and CMOS logic gates could vary because of the differences in their internal structure. In the following parts of the experiment, you are going to examine the characteristics of the logical gates rather than their logical functions.

5.2 Preliminary

- Examine the data-sheets for the following gates.
 - 74LS00 - TTL NAND gate
 - 4011 - CMOS NAND gate
- Refresh your knowledge on the main differences between TTL and CMOS technologies and the internal structures of electronic digital circuits.
- Refresh your knowledge on basic electricity (current, resistance, voltage, Ohm's law etc.) and signals (frequency, period etc.).

5.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- Integrated Circuits
 - 74xx00 TTL NAND Gates
 - 4011 CMOS NAND Gates
- Oscilloscope
- 100Ω Resistor

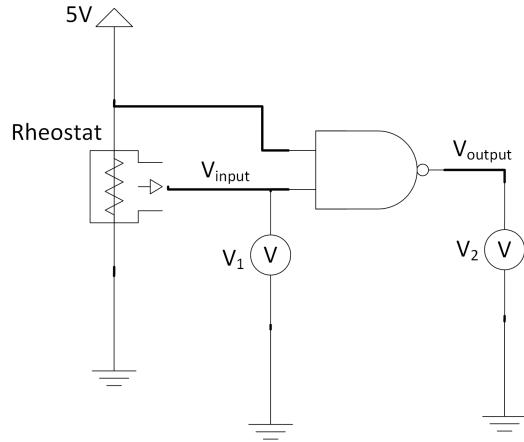
5.4 Experiment

5.4.1 Static Characteristics of TTL and CMOS NAND Gates

In the following parts of the experiment, you should fill in the given characteristic tables (Tables 5.1, 5.2) by observing the voltages in the circuits you have conducted.

5.4.1.a Output State Switching Characteristics in the Idle Mode

Idle Mode characteristic of a gate can be defined as the function $V_{output} = f_1(V_{input})$ when there is not any load at the output of the gate.



Implement the circuit given above by using TTL and CMOS gates separately. After the implementation phase, change the input voltage of the gate (V_{input}) between 0V to 5V by using the rheostat and observe the value of output voltage of the NAND gate (V_{output}) and fill in the tables below.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---------------------------------------|---|---|---|---|---|---|---|
| $\frac{V_{input}}{V_{output}}$ (Volt) | | | | | | | |
| $\frac{V_{output}}{V_{input}}$ (Volt) | | | | | | | |

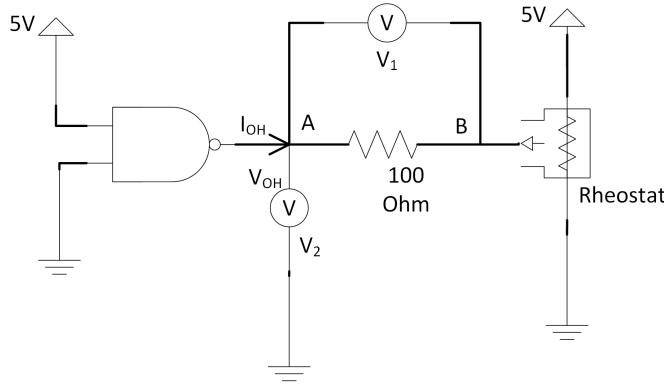
Table 5.1: Switching Characteristics of TTL

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---------------------------------------|---|---|---|---|---|---|---|
| $\frac{V_{input}}{V_{output}}$ (Volt) | | | | | | | |
| $\frac{V_{output}}{V_{input}}$ (Volt) | | | | | | | |

Table 5.2: Switching Characteristics of CMOS

5.4.1.b V_{OH} - I_{OH} Characteristics

Characteristic of V_{OH} - I_{OH} can be defined as the function $V_{OH} = f_2(I_{OH})$ when the output state of the gate is Logic-1. Main objective of this experiment is to determine the maximum possible value of the output current (I_{OH}) which the gate could provide without changing its output state to Logic-0. To this end, you need to implement the circuit given below by using both TTL and CMOS gates separately.



After you conclude the implementation phase, change the output voltage of the gate (V_{OH}) between $0V$ to $5V$ by using the rheostat and observe the voltage (V_{AB}) between the points A and B; and fill the given tables (5.3, 5.4) below.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|---|---|---|---|---|---|---|
| V_{AB} (Volt) | | | | | | | |
| V_{OH} (Volt) | | | | | | | |
| I_{OH} (mA) | | | | | | | |

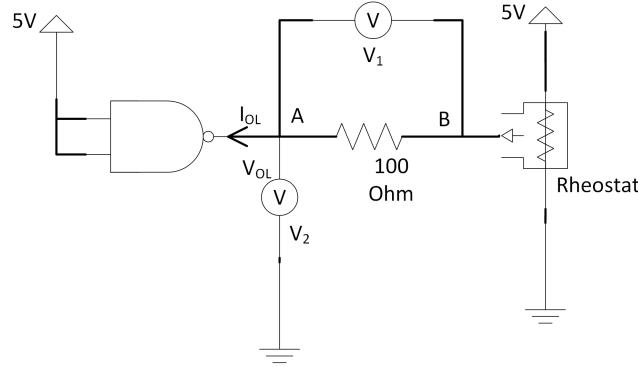
Table 5.3: V_{OH} - I_{OH} Characteristics of TTL

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|---|---|---|---|---|---|---|
| V_{AB} (Volt) | | | | | | | |
| V_{OH} (Volt) | | | | | | | |
| I_{OH} (mA) | | | | | | | |

Table 5.4: V_{OH} - I_{OH} Characteristics of CMOS

5.4.1.c V_{OL} - I_{OL} Characteristics

Characteristic of V_{OL} - I_{OL} can be defined as the function $V_{OL} = f_3(I_{OL})$ when the output state of the gate is Logic-0. Main objective of this experiment is to determine the maximum possible value of the current (I_{OL}) toward the output which the gate could consume without changing its output state to Logic-1. To this end, you need to implement the circuit given below by using both TTL and CMOS gates separately.



After you conclude the implementation phase, change the output voltage of the gate (V_{OL}) between $0V$ to $5V$ by using the rheostat and observe the voltage (V_{AB}) between the points A and B; and fill the given tables (5.5, 5.6) below.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|---|---|---|---|---|---|---|
| V_{AB} (Volt) | | | | | | | |
| V_{OL} (Volt) | | | | | | | |
| I_{OL} (mA) | | | | | | | |

Table 5.5: V_{OL} - I_{OL} Characteristics of TTL

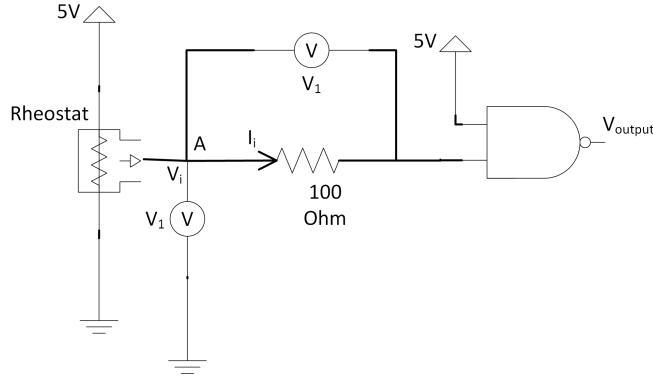
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|---|---|---|---|---|---|---|
| V_{AB} (Volt) | | | | | | | |
| V_{OL} (Volt) | | | | | | | |
| I_{OL} (mA) | | | | | | | |

Table 5.6: V_{OL} - I_{OL} Characteristics of CMOS

5.4.1.d V_i - I_i Characteristics

Characteristic of V_i - I_i can be defined as the function $V_i = f_4(I_i)$ when there is not any load at the output of the gate. This characteristic specifies the input current which the gate could consume or provide. When the logic gates are connected sequentially, there will be a flow of current between outputs and inputs of the logic gates. This current has to be measured in order to determine how many separate gates could be connected to the output of a logic gate without violating the operating boundaries of a logic circuit.

In order to measure the input current of a logic gate, you should implement two distinct circuits by using TTL and CMOS NAND gates. The design which you should implement is given below.



After you conclude the implementation phase, change the input voltage of the gate (V_i) between 0V to 5V by using the rheostat and observe the voltage (V_{AB}) between the points A and B; and fill the given tables (5.7, 5.8) below.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|---|---|---|---|---|---|---|
| V_{AB} (Volt) | | | | | | | |
| V_i (Volt) | | | | | | | |
| I_i (mA) | | | | | | | |

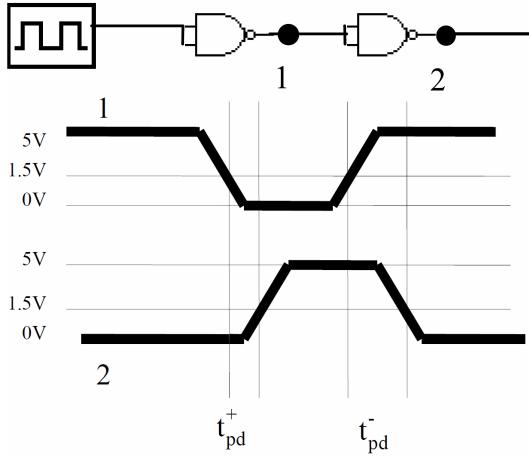
Table 5.7: V_i - I_i Characteristics of TTL

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------|---|---|---|---|---|---|---|
| V_{AB} (Volt) | | | | | | | |
| V_i (Volt) | | | | | | | |
| I_i (mA) | | | | | | | |

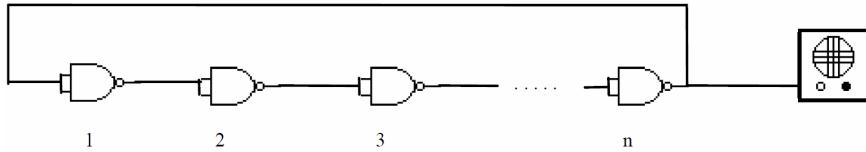
Table 5.8: V_i - I_i Characteristics of CMOS

5.4.2 Dynamic Characteristics of TTL and CMOS NAND Gates

Delay of a logic gate involves values t_{pd}^+ and t_{pd}^- which are visualized in the figures below.



Implement the circuit below for both TTL and CMOS gates. Here, there is no waveform input to circuit. n should be an odd number. Delay of a logic gate will be calculated using the oscilloscope to measure the oscillation period of the signal. Total delay of a gate is equal to $t_{pd} = t_{pd}^+ + t_{pd}^- = \frac{T}{2*n}$ where T is the measured period. Moreover, the output signal should be in triangle wave form. Please explain the reason of it.



5.4.3 Power Consumption of TTL and CMOS NAND Gates

In this part, you will monitor the power drawn by TTL and CMOS gates on varying frequencies. Apply a signal with a frequency between $0Hz$ to $1Mhz$ to all inputs of the gate and calculate the power drawn by the gate to fill in the tables 5.9 and 5.10. In that way you will obtain the $P_D = G(f)$ relation.

Remember that the power a gate draws is equal to $P_D = I_{CC} * V_{CC}$. You can calculate I_{CC} by using the voltage drop on a 100Ω resistor which links V_{CC} and the integrated circuit.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------------|---|---|---|---|---|---|---|
| Frequency (Hz) | | | | | | | |
| Power (watt) | | | | | | | |

Table 5.9: Power Consumption in TTL

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------------|---|---|---|---|---|---|---|
| Frequency (Hz) | | | | | | | |
| Power (watt) | | | | | | | |

Table 5.10: Power Consumption in CMOS

5.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams of the expressions which were implemented during this experiment.
- Your results as tables and line charts for each part of the experiment.

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

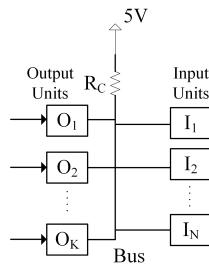
Data Bus Implementation in Digital Systems

6.1 Introduction

In the following experiment, a data bus is going to be implemented by using three-state buffers and ICs with open collector outputs.

6.2 Basic Principles

Buses are frequently used in digital systems in order to reduce the cost of the physical connections. For example, a digital system which consist of N distinct units requires $N \cdot (N - 1)/2$ physical connections to create fully connected network. Since providing distinct physical lines (wires) for each connection is an expansive approach, sub-set of digital elements communicates through the shared bus in time shared manner. A bus can be implemented as a "wired OR" circuit by using ICs with open collector outputs or it can be implemented by using 3-state buffers. Abstract design of the "wired OR" bus is given in the figure below.



Lower and upper bounds of the resistor in the given design can be calculated by using the following equations.

$$R_{C(min)} = [V_{cc} - V_{IL(max)}] / [I_{OL(max)} - I_{IL(max)} \cdot N]$$

$$R_{C(max)} = [V_{cc} - V_{IH(min)}] / [I_{OH(max)} \cdot K + I_{IH(max)} \cdot N]$$

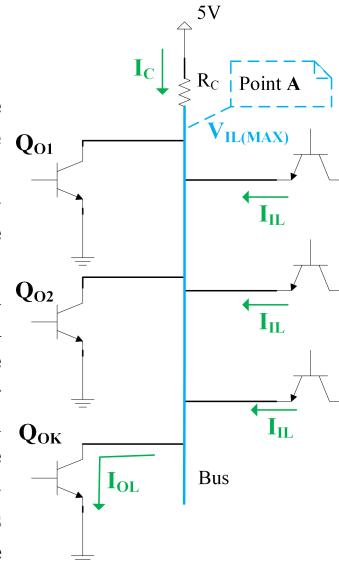
Input and output phases of the ICs with open collector have to be examined in order to grasp the behaviour of the data bus. During this experiment, we are going to use the transistors as switches. Thus, we are going to force the transistors to operate in saturation and cut-off modes only.

6.2.1 Assigning Logic-0 to the Common Bus

In order to acquire logic-0 in the common bus, at least one of the output transistors (Q_O) must be in the saturation mode. An exemplary scenario is given in the figure on the right. In this circuit, we assume that only the K^{th} transistor (Q_{OK}) is operating in the saturation mode, thus the current (I_{OL}) is flowing on it through the ground. Meanwhile, the other output transistors are in cut-off mode, and their currents are negligible. There is an upper bound ($V_{IL(max)}$) for the voltage in the point **A** in order to get logic-0 in the common bus. If the voltage becomes higher than this bound we can not guarantee to read the content of the bus as logic-0. The value of ($V_{IL(max)}$) is equal to $0.8V$ for the TTL-family units. When the logic value of the bus is equal to 0, the currents (I_{IL}) flow from the emitters of the input transistors through the bus. Minimum value of the resistor R_C can be determined by defining the flow equation for the bus. Current values for 74LS05 are given below.

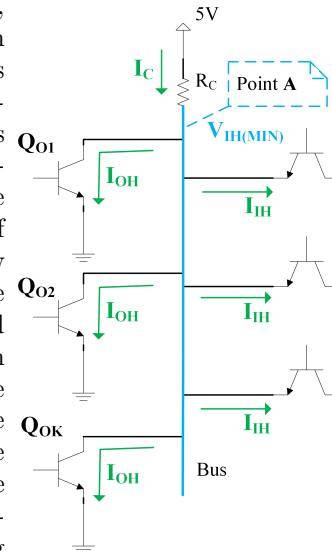
$$I_{OL(max)} = 8mA$$

$$I_{IL(max)} = 0.4mA$$



6.2.2 Assigning Logic-1 to the Common Bus

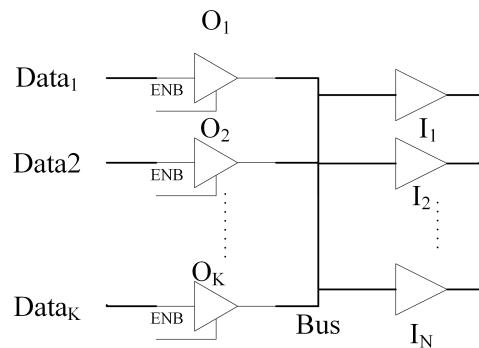
In order to acquire logic-1 in the common bus, all of the output transistors (Q_O) must be in the cut-off mode. An exemplary scenario is given in the figure on the right. In this circuit, all of the output transistors (Q_{OK}) is operating in the cut-off mode, thus the currents (I_{OH}) are flowing on them through the ground. Additionally, when the logic value of the bus is equal to 1, the currents (I_{IH}) flow from the bus through the emitters of the input transistors. There is an lower bound ($V_{IH(min)}$) for the voltage in the point A in order to get logic-1 in the common bus. If the voltage becomes lower than this boundary we may read the value of the bus as logic-0. The value of ($V_{IH(min)}$) is equal to 2.0V for the TTL-family units. Maximum value of the resistor R_C can also be determined by defining the flow equation for the bus. Current values for 74LS05 are given below.



$$I_{OH(max)} = 100\mu A$$

$$I_{IH(max)} = 20\mu A$$

Buses can also be implemented by using 3-state buffers. There is no need for a resistor in this design. Output a non-active 3-state buffer will be in high-impedance. There has to be only one active 3-state buffer which runs the bus as an output. An example implementation of a bus by using 3-state buffers is given below.



6.3 Preliminary

- Study the datasheets of 74xx241 and 74xx05
- Refresh your knowledge on ICs with open collector outputs and 3-state buffers.
- Assume $N = 1$ and $K = 2$ and, calculate the $R_{C(max)}$, $R_{C(min)}$ for the given circuits in experiment parts 6.2.1 and 6.2.2.

6.4 Equipments and Integrated Circuits (ICs)

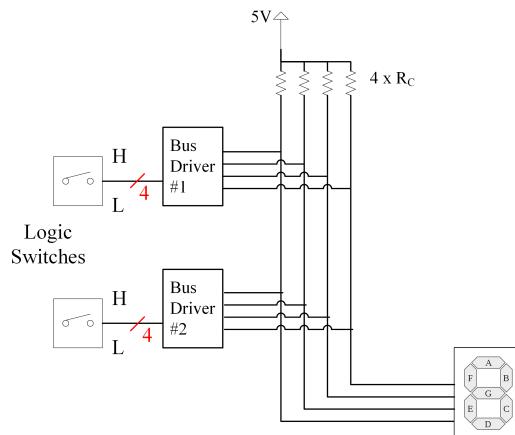
Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
 - 74xx241 Octal 3-State Buffer
 - 74xx05 Hex Inverters with Open-Collector Outputs
- Resistors

Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

6.5 Experiment

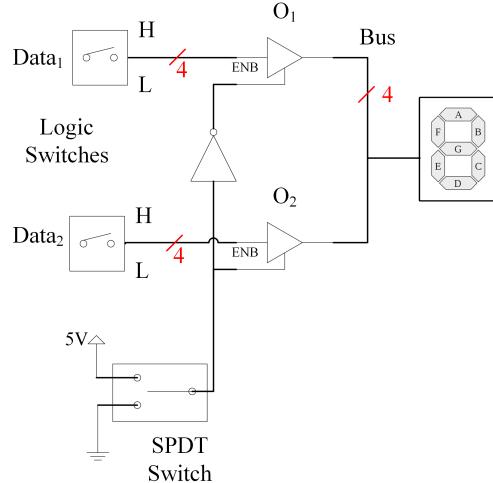
6.5.1 Experiment - Part 1



4 bit data bus with 2 drivers with open collector outputs

You are going to implement 4-bit data bus by using ICs with open collector outputs (given Hex Inverters). Use your calculations $R_{C(max)}$, $R_{C(min)}$ for $N=1$ and $K=2$ while building the circuit and use suitable resistance. In this circuit, each bit on the bus is shared by two different bus driver units. Here, please use Hex Inverters with Open-Collector Outputs for the bus driver units. While interpreting the circuit, please take attention that given inputs are going to be inverted before being transferred on bus. The basic design of the circuit is given in the Figure 6.1. After the implementation of the circuit, observe the output of the bus by keeping in mind the behavior of the "wired OR" circuit.

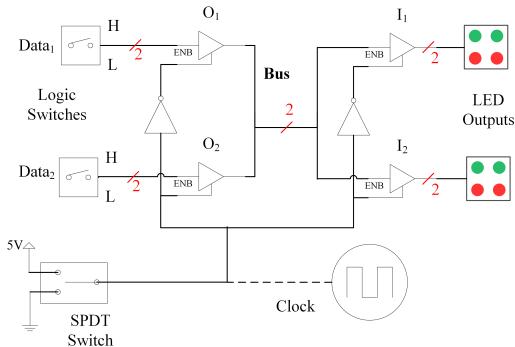
6.5.2 Experiment - Part 2



4 bit data bus with 2 drivers with 3-state buffers

In this part of the experiment, you need to implement a 4-bit bus by using 3-state buffers. The circuit diagram is given in the Figure 6.2. You should use logic switches on the CADET as data inputs and one of the SPDT switches for the enable inputs of the 3-state buffers.

6.5.3 Experiment - Part 3



2 bit data bus with 2 drivers and 2 readers

As the last part of experiment, you need implement the circuit given in the Figure 6.3. This circuit contains a 2-bit bus with two distinct outputs and inputs. First, you need to use SPDT switch for the enable pins of the 3-state buffers.

After validating the behaviour of the bus, you need detach SPDT switch and connect frequency generator instead for the enable pins. When you connect the frequency generator, observe the output of the circuit first for lower frequency and then increase the frequency and re-check.

6.6 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams of the expressions which were implemented during this experiment.
- Your results as tables for each part of the experiment.

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

Latches and Flip-flops

7.1 Introduction

In this experiment, you will implement and examine data storage elements: latches and flip-flops.

7.2 Preliminary

- Refresh your knowledge on how latches and flip-flops work.
- Design and draw the circuit to implement in each experiment part.
- Decide which input data must be loaded to the shift register for each case in 7.4.4.

7.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
 - 74xx00 - Quadruple 2-input Positive NAND Gates
 - 74xx02 - Quadruple 2-input Positive NOR Gates
 - 74xx04 - Hex Inverters
 - 74xx75 - Quadruple Bistable D Type Latches
 - 74xx165 - 8-Bit Parallel Input/Serial Output Shift Register
- Oscilloscope

Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

7.4 Experiment

7.4.1 Experiment - Part 1

Implement a SR type latch without an enable input. Use only NOR gates. Use switches for S and R inputs and LEDs for Q and Q_N outputs. Create a truth table for the latch by testing all possible input combinations. Using this truth table, write the characteristic function of the latch as $Q(t+1) = f(S, R, Q(t))$. Note how the latch behaves for disallowed inputs.

7.4.2 Experiment - Part 2

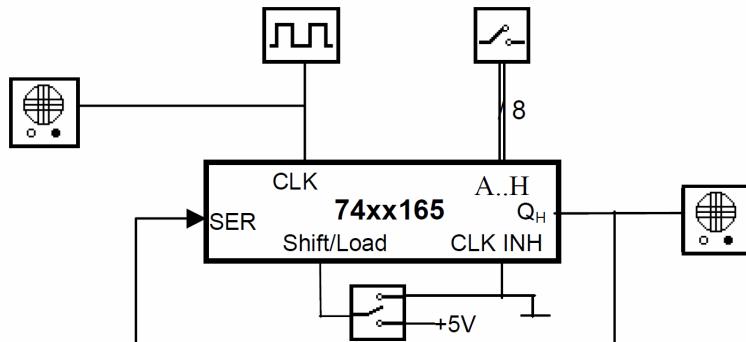
Implement a SR type latch with an enable input, C . Use only NAND gates. Use switches for S , R and C inputs and LEDs for Q and Q_N outputs. Create a truth table for the latch by testing all possible input combinations. Note how the latch behaves for disallowed inputs and how enable input effects the output.

7.4.3 Experiment - Part 3

Implement a negative edge triggered D type flip-flop using two D type latches and one inverter. Use a debounced pushbutton for the clock input, a switch for D and LEDs for Q and Q_N outputs. Show that the clock is only effective at falling edge.

7.4.4 Experiment - Part 4

Implement a pulse generator using a shift register. It should support variable pulse frequencies and durations. Build the circuit below and generate given signals. For each signal, observe both input and output using the oscilloscope and draw.



- with the 1/2 frequency of input
- with the 1/4 frequency of input
- with the 1/8 frequency of input
- with 1/3 pulse-gap duration rate
- with 1/7 pulse-gap duration rate

7.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams of the circuits which were implemented during this experiment.
- Your results as truth tables and discussions for the first 3 parts of the experiment.
- Your signal drawings and input values for the last part of the experiment.

Sequential Logic Circuits

8.1 Introduction

In this experiment, you will implement and analyze sequential logic circuits with the finite state machine model.

8.2 Preliminary

- Refresh your knowledge on how to design circuits with Mealy and Moore models.
- Analyze the circuit in 8.4.1 to create a state transition table and a state chart.
- Design and draw the circuit to implement in 8.4.2 with the minimum number of logic gates.
- Study how 74xx161 integrated circuit works. Draw the circuit to implement in 8.4.3.

8.3 Equipments and Integrated Circuits (ICs)

Following equipment and ICs are going to be used in the experiment.

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
 - 74xx04 - Hex Inverters
 - 74xx08 - Quadruple 2-input Positive AND Gates
 - 74xx32 - Quadruple 2-input Positive OR Gates
 - 74xx161 - Synchronous 4-Bit Binary Counter
 - 74xx174 - Hex D-Type Flip-Flops

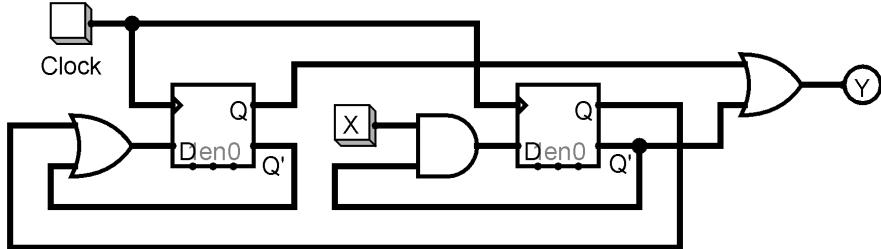
Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

8.4 Experiment

8.4.1 Experiment - Part 1

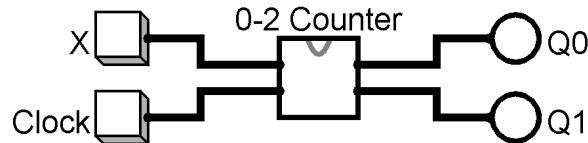
Build the circuit below. X input should be connected to a switch, while the clock should be given using a debounced pushbutton. Use LEDs to examine

the output Y as well as flip-flop states. Compare these values with the result of your preliminary analysis.



8.4.2 Experiment - Part 2

Build a 2 bit counter circuit that counts 0 to 2 in a circular way. Block diagram of the counter is given below. You should implement the part shown as a box. Here, input X is the counting direction. $X = 0$ means counting DOWN (2-1-0-2-1-0...) while $X = 1$ means counting UP (0-1-2-0-1-2-...).



After verifying that the circuit works correctly, test it for the undetermined state ($Q_1Q_0 = 11$). To do this, you should disconnect the flip-flops, load them with 1, send clock signal and reconnect them to the circuit.

8.4.3 Experiment - Part 3

Using the 74xx161 integrated circuit and other necessary gates, implement a counter that counts 0 to 5 in a circular way. Demonstrate the output in the seven segment display.

8.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Give the steps of your analysis for the circuit in 8.4.1.
- Give the steps of your design for the circuit in 8.4.2.
- Discuss the behaviour of your circuit in 8.4.2 for the undetermined state.
- Draw the circuit you implemented in 8.4.3 and explain how it works.

During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

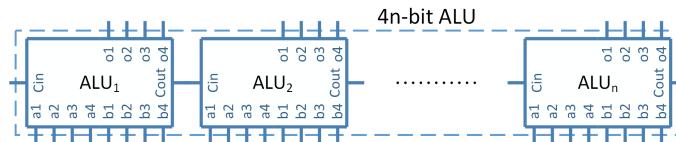
Design and Implementation of Serial Arithmetic Logic Unit

9.1 Introduction

Main objective of this experiment is to emphasize the use of sequential circuits (rather than combinational) in the realization of arithmetical and logical operations. In this manner, a serial arithmetic logical unit (sALU) is going to be designed and implemented.

9.1.1 Serial Arithmetic Logical Unit (sALU)

4-bit ALU (74xx181) was used in order to realize arithmetic and logical operations during the third experiment (Binary Arithmetic). As you could recall, this 4-bit ALU is a combinational circuit thus it does not require a clock signal to operate and it is capable of operating on 4-bit variables. By keeping this in mind, we could say that the size of variables could be easily extended by using additional number of ALUs in parallel as given in the Figure 9.1.



Extending bandwidth of an ALU in parallel

However, this extension dramatically increase the complexity (number of logic gates) of the circuit. In order to overcome this challenge, bandwidth of an ALU can be extended in a serial manner. To this end, memory units could be utilized to store the results of sequential inputs in order to provide wider operation capability. The complexity of the circuits could be easily reduced with the serial approach, however using memory units necessities the clock signal as an input. Thus, a logical or arithmetical operation requires multiple clock cycles in order to complete its execution. For example, we could extend the operation capability of an 4-bit ALU to 16-bit by using 16-bit D-type flip-flop but each operation requires 4 clock cycles before completion.

9.2 Preliminary

- Refresh your knowledge on how ALU works.
- The operation list of 1-bit ALU has been given below. Before the experiment, each group **should design a circuit** (ALU) which is capable to realize the given operations.
Each group **should bring the hard or soft copies** of their design. Groups which do not bring the requested documents are going to be considered **absent** for the experiment.

Table 9.1: Operation List

| Operations | RTL |
|---------------------|---------------------------------------|
| Addition with Carry | $C_{out}O \leftarrow A + B + C_{in}$ |
| Subtraction | $C_{out}O \leftarrow A + B' + C_{in}$ |
| XOR | $O \leftarrow A \oplus B$ |
| Clear | $O \leftarrow 0$ |
| Complement | $O \leftarrow A'$ |
| Two Complement | $O \leftarrow A' + C_{in}$ |
| Transfer B | $O \leftarrow B$ |

In this table, O is the output bit, A and B are input bits, C_{out} is carry output and C_{in} is carry input.

9.3 Equipments and Integrated Circuits (ICs)

- C.A.D.E.T.
- 74000 series ICs ¹

9.4 Experiment

9.4.1 Experiment - Part 1

In the first part of the experiment, you should implement and test your 1-bit ALU design.

9.4.2 Experiment - Part 2

You should extend the capability of your 1-bit ALU for 4-bit variables by using memory units. In this manner, you should implement 4-bit serial ALU by using 1-bit ALU which you have implemented in previous part of the experiment.

¹ The groups should decide the necessary ICs with respect to their designs.

9.5 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. Your report should also include the following materials:

- Circuits diagrams that were implemented during this experiment.
- Your test inputs and observed results as tables for each part of the experiment.

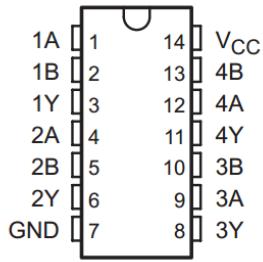
During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.

A

Data Sheets

A.1 7400 - Quadruple 2-input Positive-NAND Gates

Pin Configuration

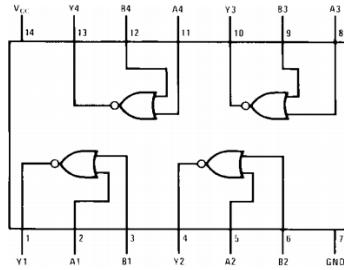


Function Table

| INPUTS | | OUTPUT |
|--------|---|--------|
| A | B | Y |
| H | H | L |
| L | X | H |
| X | L | H |

A.2 7402 - Quad 2-input Positive-NOR Gates

Pin Configuration



Function Table

$Y = \overline{A + B}$

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

H = HIGH Logic Level

L = LOW Logic Level

A.3 7404 - Hex Inverters

Pin Configuration

| | INPUT A | OUTPUT Y |
|---|------------|-------------|
| H | L | |
| L | H | |

Function Table

A.4 7405 - Hex Inverters with Open-Collector Outputs

Pin Configuration

Function Table

$$Y = \bar{A}$$

| Input | Output |
|-------|--------|
| A | Y |
| L | H |
| H | L |

H = HIGH Logic Level
L = LOW Logic Level

A.5 7408 - Quadruple 2-input Positive-AND Gates

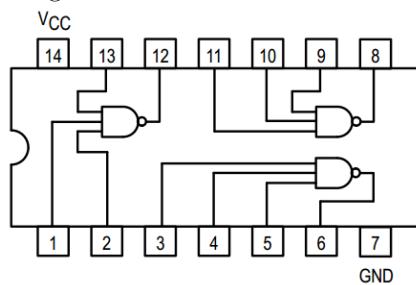
Pin Configuration

Function Table

| INPUTS | | OUTPUT |
|--------|---|--------|
| A | B | Y |
| H | H | H |
| L | X | L |
| X | L | L |

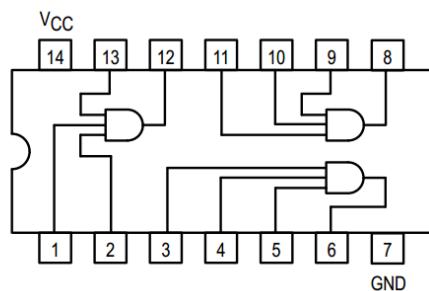
A.6 7410 - Triple 3-input NAND Gates

Pin Configuration



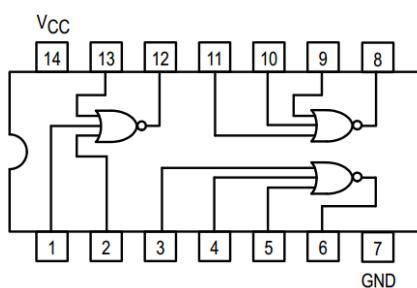
A.7 7411 - Triple 3-input AND Gates

Pin Configuration



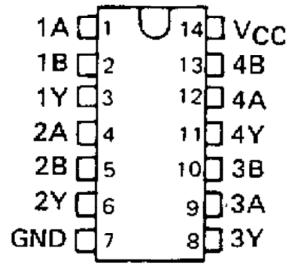
A.8 7427 - Triple 3-input NOR Gates

Pin Configuration



A.9 7432 - Quadruple 2-input Positive-OR Gates

Pin Configuration

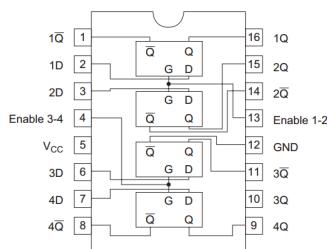


Function Table

| INPUTS | | OUTPUT |
|--------|---|--------|
| A | B | Y |
| H | X | H |
| X | H | H |
| L | L | L |

A.10 7475 - Quadruple Bistable Latches

Pin Configuration



Function Table

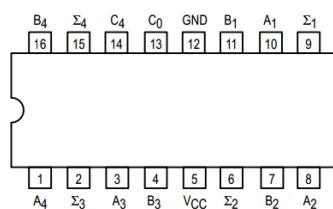
| Inputs | | Outputs | |
|--------|---|---------|-------------|
| D | G | Q | \bar{Q} |
| L | H | L | H |
| H | H | H | L |
| X | L | Q_0 | \bar{Q}_0 |

H; high level, L; low level, X; irrelevant

 Q_0 ; level of Q before the indicated steady-state input conditions were established. \bar{Q}_0 ; complement of Q_0 or level of \bar{Q}_0 before the indicated steady-state input conditions were established.

A.11 7483 - 4-Bit Binary Full Adder with Fast Carry

Pin Configuration



Function Table

FUNCTIONAL TRUTH TABLE

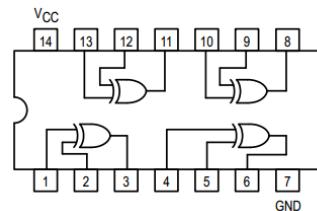
| C (n-1) | A _n | B _n | Σ_n | C _n |
|---------|----------------|----------------|------------|----------------|
| L | L | L | L | L |
| L | L | H | H | L |
| L | H | L | H | L |
| L | H | H | L | H |
| H | L | L | H | L |
| H | L | H | L | H |
| H | H | L | L | H |
| H | H | H | H | H |

C₁ — C₃ are generated internallyC₀ — is an external inputC₄ — is an output generated internally

A.12 7486 - Quad 2-Input Exclusive Or Gate

Pin Configuration

Function Table

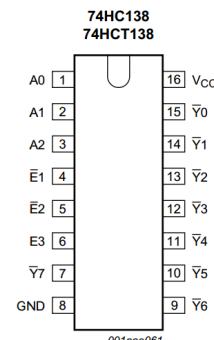


| IN | | OUT |
|----|---|-----|
| A | B | Z |
| L | L | L |
| L | H | H |
| H | L | H |
| H | H | L |

A.13 74138 - 3-to-8 decoder/demultiplexer

Pin Configuration

Function Table



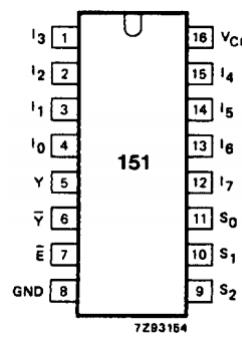
| Control | | | Input | | | Output | | | | | | | | |
|---------|----|----|-------|----|----|--------|----|----|----|----|----|----|----|---|
| E1 | E2 | E3 | A2 | A1 | A0 | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 | |
| H | X | X | X | X | X | H | H | H | H | H | H | H | H | |
| X | H | X | | | | | | | | | | | | |
| X | X | L | | | | | | | | | | | | |
| L | L | H | L | L | L | H | H | H | H | H | H | H | L | |
| | | | L | L | H | H | H | H | H | H | H | H | L | H |
| | | | L | H | L | H | H | H | H | H | L | H | H | |
| | | | L | H | H | H | H | H | H | L | H | H | H | |
| | | | H | L | L | H | H | H | L | H | H | H | H | |
| | | | H | L | H | H | H | L | H | H | H | H | H | |
| | | | H | H | I | H | I | H | H | H | H | H | H | |

[1] H = HIGH voltage level;
L = LOW voltage level;
X = don't care.

A.14 74151 - 8-Input Multiplexer

Pin Configuration

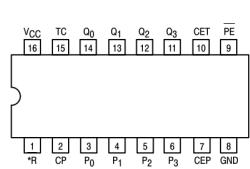
Function Table



| INPUTS | | | | | | | | | | | | OUTPUTS | |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|-----|
| \bar{E} | S_2 | S_1 | S_0 | I_0 | I_1 | I_2 | I_3 | I_4 | I_5 | I_6 | I_7 | \bar{Y} | Y |
| H | X | X | X | X | X | X | X | X | X | X | X | H | L |
| L | L | L | L | L | X | X | X | X | X | X | X | H | L |
| L | L | L | L | L | H | X | X | X | X | X | X | H | H |
| L | L | L | L | H | X | L | X | X | X | X | X | L | H |
| L | L | L | L | H | X | H | X | X | X | X | X | H | H |
| L | L | H | L | X | X | L | X | X | X | X | X | L | H |
| L | L | H | H | X | X | H | X | X | X | X | X | L | H |
| L | L | H | H | X | X | X | X | L | X | X | X | H | L |
| L | L | H | H | H | X | X | X | X | X | X | X | L | H |
| L | L | H | H | H | X | X | X | X | X | X | X | L | H |
| L | H | L | L | X | X | X | X | L | X | X | X | H | L |
| L | H | L | L | X | X | X | X | H | X | X | X | L | H |
| L | H | L | L | H | X | X | X | X | X | L | X | H | L |
| L | H | L | L | H | X | X | X | X | H | X | X | L | H |
| L | H | H | H | L | X | X | X | X | X | L | X | H | L |
| L | H | H | H | H | X | X | X | X | X | X | X | H | L |
| L | H | H | H | H | H | X | X | X | X | X | X | H | L |
| L | H | H | H | H | H | X | X | X | X | X | X | H | L |
| L | H | H | H | H | H | H | X | X | X | X | X | H | L |

A.15 74161 - BCD Decade Counter / 4-Bit Binary Counter

Pin Configuration

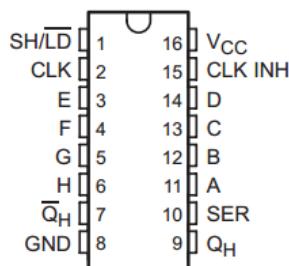


Function Table

| *SR | PE | CET | CEP | Action on the Rising Clock Edge ($\neg \bar{C}$) |
|-----|----|-----|-----|--|
| L | X | X | X | RESET (Clear) |
| H | L | X | X | LOAD (P_n Q_n) |
| H | H | H | H | COUNT (Increment) |
| H | H | L | X | NO CHANGE (Hold) |
| H | H | X | L | NO CHANGE (Hold) |

A.16 74165 - 8-Bit Parallel-Load Shift Register

Pin Configuration



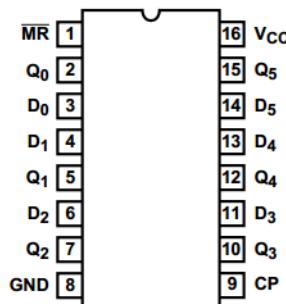
Function Table

| FUNCTION TABLE | | | |
|----------------|-----|----------|---------------|
| INPUTS | | FUNCTION | |
| SH/LD | CLK | CLK INH | |
| L | X | X | Parallel load |
| H | H | X | No change |
| H | X | H | No change |
| H | L | ↑ | Shift† |
| H | ↑ | L | Shift† |

† Shift = content of each internal register shifts toward serial output Q_H . Data at SER is shifted into the first register.

A.17 74174 - Hex D-Type Flip-Flop with Reset

Pin Configuration



Function Table

| INPUTS | | | OUTPUT |
|------------|----------|------------|--------|
| RESET (MR) | CLOCK CP | DATA D_n | Q_n |
| L | X | X | L |
| H | ↑ | H | H |
| H | ↑ | L | L |
| H | L | X | Q_0 |

H = High Voltage Level, L = Low Voltage Level, X = Irrelevant, ↑ = Transition from Low to High Level, Q_0 = Level Before the Indicated Steady-State Input Conditions Were Established

A.18 74181 - 4-Bit Arithmetic Logic Unit

Pin Configuration & Function Table

| | | | |
|----------------|----|----|-----------------|
| \bar{B}_0 | 1 | 24 | V _{CC} |
| \bar{A}_0 | 2 | 23 | \bar{A}_1 |
| S ₃ | 3 | 22 | \bar{B}_1 |
| S ₂ | 4 | 21 | \bar{A}_2 |
| S ₁ | 5 | 20 | \bar{B}_2 |
| S ₀ | 6 | 19 | \bar{A}_3 |
| C _n | 7 | 18 | \bar{B}_3 |
| M | 8 | 17 | \bar{G} |
| \bar{F}_0 | 9 | 16 | C_{n+4} |
| \bar{F}_1 | 10 | 15 | P |
| \bar{F}_2 | 11 | 14 | A=B |
| GND | 12 | 13 | \bar{F}_3 |

| | | | | Pin Names | Description |
|--------------------------------|--|--|--|-----------|-------------------------------------|
| $\bar{A}_0-\bar{A}_3$ | | | | | Operand Inputs (Active LOW) |
| $\bar{B}_0-\bar{B}_3$ | | | | | Operand Inputs (Active LOW) |
| S ₀ -S ₃ | | | | | Function Select Inputs |
| M | | | | | Mode Control Input |
| C _n | | | | | Carry Input |
| $\bar{F}_0-\bar{F}_3$ | | | | | Function Outputs (Active LOW) |
| A = B | | | | | Comparator Output |
| \bar{G} | | | | | Carry Generate Output (Active LOW) |
| P | | | | | Carry Propagate Output (Active LOW) |
| C_{n+4} | | | | | Carry Output |

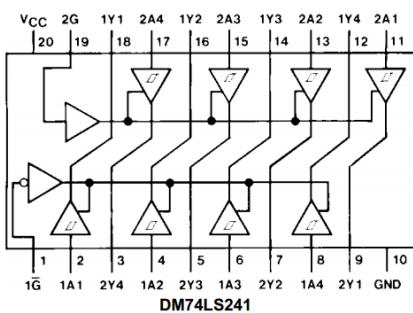
| Mode Select Inputs | | | | Active LOW Operands & F _n Outputs | | Active HIGH Operands & F _n Outputs | |
|--------------------|----------------|----------------|----------------|--|--|---|--|
| S ₃ | S ₂ | S ₁ | S ₀ | Logic (M = H) | Arithmetic (Note 2) (M = L) (C _n = L) | Logic (M = H) | Arithmetic (Note 2) (M = L) (C _n = H) |
| L | L | L | L | \bar{A} | A minus 1 | \bar{A} | A |
| L | L | L | H | $\bar{A}\bar{B}$ | AB minus 1 | $\bar{A} + \bar{B}$ | A + B |
| L | L | H | L | $\bar{A} + \bar{B}$ | AB minus 1 | $\bar{A}B$ | A + \bar{B} |
| L | L | H | H | Logic 1 | minus 1 | Logic 0 | minus 1 |
| L | H | L | L | $\bar{A} + \bar{B}$ | A plus (A + \bar{B}) | $\bar{A}B$ | A plus $\bar{A}\bar{B}$ |
| L | H | L | H | \bar{B} | AB plus (A + \bar{B}) | \bar{B} | (A + B) plus $\bar{A}\bar{B}$ |
| L | H | H | L | $\bar{A} \oplus \bar{B}$ | A minus B minus 1 | $\bar{A} \oplus B$ | A minus B minus 1 |
| L | H | H | H | $A + \bar{B}$ | A + \bar{B} | $\bar{A}\bar{B}$ | AB minus 1 |
| H | L | L | L | $\bar{A}B$ | A plus (A + B) | $\bar{A} + B$ | A plus AB |
| H | L | L | H | $A \oplus B$ | A plus B | $\bar{A} \oplus \bar{B}$ | A plus B |
| H | L | H | L | B | AB plus (A + B) | B | (A + \bar{B}) plus AB |
| H | L | H | H | A + B | A + B | AB | AB minus 1 |
| H | H | L | L | Logic 0 | A plus A (Note 1) | Logic 1 | A plus A (Note 1) |
| H | H | L | H | $\bar{A}\bar{B}$ | AB plus A | $A + \bar{B}$ | (A + B) plus A |
| H | H | H | L | AB | $\bar{A}\bar{B}$ minus A | $A + B$ | (A + \bar{B}) plus A |
| H | H | H | H | A | A | A | A minus 1 |

Note 1: Each bit is shifted to the next most significant position.

Note 2: Arithmetic operations expressed in 2's complement notation.

A.19 74241 - Octal 3-State Buffer/ Line Driver/ Line Receiver

Pin Configuration



Function Table

| Inputs | | | | Outputs | |
|--------|-----------|----|----|---------|----|
| G | \bar{G} | 1A | 2A | 1Y | 2Y |
| X | L | L | X | L | |
| X | L | H | X | H | |
| X | H | X | X | Z | |
| H | X | X | L | | L |
| H | X | X | H | | H |
| L | X | X | X | X | Z |

A.20 4011 - Quad 2-Input NAND Gate

Pin Configuration

