Gökhan Seçinti
Atakan Aral

Istanbul Technical University

Department of Computer Engineering

BLG 242E – Logic Circuits Laboratory

Experiments Booklet

Spring 2016

Version 1.8.1

Contents

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 15 in the picture. We will briefly describe the functions that you will most likely need to be familiar with for the labs[1].

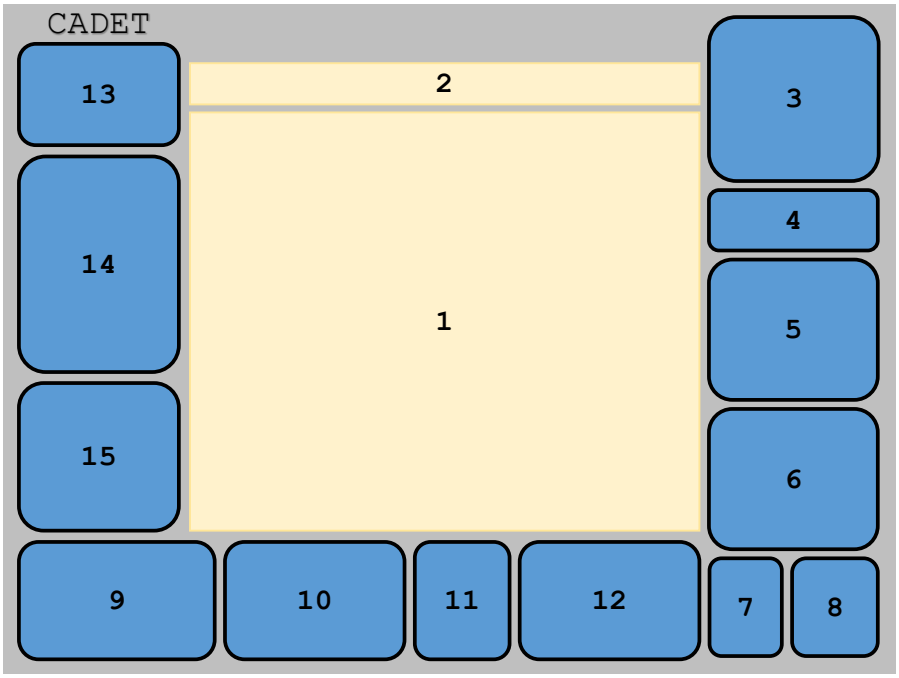


Fig. 0.1. Layout of the CADET.

[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. 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 three 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). No column of tie points is connected to any other column, however (no horizontal connectivity). Horizontal rows of 25 tie points, above and below the center mounting screw, are the same points electrically. Note that the vertical columns above and below the center mounting screw are not connected to each other.

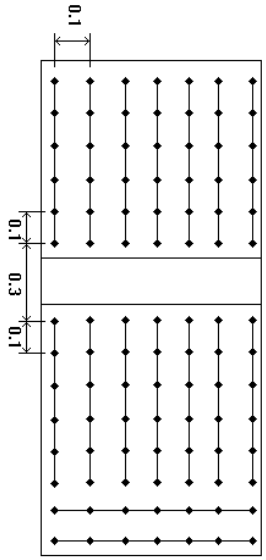


Fig. 0.2. Interconnection of the holes.

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. The outer horizontal rows are usually used to supply voltage and ground to the circuit while the inner vertical rows are used for signal connections[2].

2 Power Strips

The holes in the top 2 strips of the breadboard are horizontally connected to each other, however there is no vertical connectivity (between rows). They are typically used to distribute power supply to circuit components.

[2] http://www.dejazzer.com/ece311/labs/station.html

3 Power Supply

At the top of the breadboard area are two horizontal connector strips of breadboarding material. To their right are lines printed on the surface of the trainer indicating how they are connected to the various power supplies. Each entire row is connected to a separate power supply. The bottom row is connected to ground. The CADET features three power supplies: a fixed +5 V supply, a variable positive supply (+V), and a variable negative supply (-V). At the top of the trainer above the power connector strips are two voltage adjustment knobs for setting +V and -V. The positive voltage (+V) can be adjusted from roughly +1.3 V to +15 V and -V from -1:3V to -15 V. If necessary, power can also be taken directly from the read, blue, yellow, and black binding posts on the top of the trainer in addition to the power connector strip.

4 Logic Probe

5 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. There are eight red LEDs at the top of this section, and eight green LEDs at the bottom. The red LEDs indicate a high or 1" logic level. The green LEDs indicate a low or 0" logic level. The indicators work in pairs shown by the numbering from one to eight. Each pair has two tie points on the small connector block to the left. It does not matter which of the two tie points you use. To the top of the LEDs are two switches marked +5/+V and TTL/CMOS. For interfacing the LEDs to digital circuits, the +5/+V switch should be set to +5 and the TTL/CMOS switch should be set to TTL. This sets the threshold voltages for the indicators to be compatible with TTL-level inputs: if the input connected to a pair of LEDs is at a voltage of 2.2 V or higher, the red indicator will light, if the input voltage is 0.8 V or less, the green indicator will light, otherwise, neither LED will light. Setting the +5V/+V switch to +V allows you to vary the threshold voltage for the indicators according to the current value of the CADET board's adjustable +V power supply.

6 Displays

There are two types of displays on the trainer. First type is the seven segment display which has four input pins labeled A, B, C and D. There 2 seven segments displays where input pins are interpreted as binary coded decimal (BCD). For instance: $(A_1B_1C_1D_1A_2B_2C_2D_2) = (00111001)$ is displayed as 39. The other type of display is the bar graph. Each bar lights up by an increment of 0.5 V in the input.

7 Loudspeaker

8 BNC Connector

9 Debounced Pushbuttons

There are two pushbuttons on the left side of the trainer. These are labeled 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). The small switch diagram printed on the trainer helps illustrate this. 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.

10 Logic Switches

In the lower left part of the trainer are eight 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. In the upper left comer of this area is a switch labeled +5/+V. If this is in the +5 position, then a high or logic 1 for the switches is +5 V. If the voltage selection switch is in the +V position, then a high or logic 1 will be the voltage of the CADET board's +V adjustable power supply, which in turn depends on the settings of the +V adjustment knob at the top of the CADET board. For your experiments, you will usually use the switches as inputs to your digital circuitry. When this is the case, always make sure that the switch is set in the +5 position.

11 SPDT Switches

These two switches are intended for sending two alternative signals or voltages to an output. When a SPDT switch is in its down position, middle two pin columns are connected to left or right columns. When it is in its up position, middle two pin columns are connected to the other column.

12 Potentiometers

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

13 Digital Voltmeter

Digital voltmeter automatically detects appropriate input range and displays DC level of input signal. Four input ranges are $\pm 0 - 199.9mV$, $\pm 200mV -$

$1.999V$, $\pm 2V - 19.99V$ and $\pm 20V - 199.9V$. If the input voltage is less than 200 mV, it displays one digit in terms of mV. Otherwise, it displays voltage in terms of V.

14 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. We will describe only this last feature of the function generator here. The leftmost two tie points on the function generator connector block provide access to the TTL output. The speed or frequency of the TTL output square wave is adjusted by three other controls. One control is the frequency slide adjustment on the left, which ranges from 0.1 at the bottom to 1.0 at the top. Another control is the decade range selection switch at the top right side of this section. It has positions labeled 1, 10, and 100. The last control is at the top left part of this section, and is a two position switch labeled KHz and Hz. The frequency of your waveform is the number of Hertz or Kilohertz (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 x 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.

15 Pulse Generator

A short-cut for square wave generation, which will provide the clock signal needed for some of the experiments, is using the pulse generator. It generates TTL or CMOS square waves between frequencies of 1 Hz and 1 MHz.

1

Boolean Algebra

1.1 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.

1.2 Preliminary

- Revise the axioms and theorems of Boolean algebra.
- Prove the given equalities below by using the axioms of Boolean algebra.
  - $x + x \cdot y = x$
  - $(x + y) \cdot (x + y') = x$
- Determine and prove the duals of the equalities defined above.
- Calculate the complementary expression (F') for the function F which is defined as follows ($F = x + y \cdot z$) by using De Morgan theorem and draw the logic circuit for both expressions (F and F').
- Simplify the defined binary function below by using the axioms and theorems of the Boolean algebra.
  $F(a, b, c, d) = a'b'd + bc'd + ab'd' + bc'd + bcd'$

1.3 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
  - 74xx[1]04 - Hex Inverters
  - 74xx[1]08 - Quadruple 2-input Positive AND Gates
  - 74xx[1]32 - 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.

1.4 Experiment

1.4.1 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.

1.4.2 Experiment - Part 2

A theorem is given as: $(a = a + a \cdot b)$. 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.

1.4.3 Experiment - Part 3

$F_3(x, y, z) = x + y \cdot z$ 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.

1.4.4 Experiment - Part 4

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

$$F_4(a, b, c, d) = \cup_1(0, 2, 5, 7, 8, 10, 13, 15)$$

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

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:

- 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.

2

Combinational Logic Circuits

2.1 Introduction

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

2.2 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 and OR gates.
   $$F_1(a,b,c) = a' \cdot c' + b \cdot c$$
   $$F_2(a,b,c) = a' \cdot b' \cdot c' + a \cdot b$$

2.3 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
  - $74xx00$ - 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.

2.4 Experiment

2.4.1 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.

2.4.2 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.

2.4.3 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.

2.4.4 Experiment - Part 4

Build the circuit you have designed in Preliminary Question 4[1] using the ICs. Fill up a truth table to evaluate your implementation.

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.
- 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.

3

Binary Arithmetic

3.1 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.

3.2 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.

3.3 Equipments and Integrated Circuits (ICs)

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

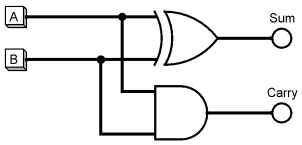
- C.A.D.E.T. (**C**omplete **A**nalogue **D**igital **E**lectronic **T**rainer)
- 74000 series ICs
  - $74xx08$ - Quadruple 2-input Positive AND Gates
  - $74xx32$ - Quadruple 2-input Positive OR Gates
  - $74xx83$ - 4-bit Binary Full Adder
  - $74xx86$ - Quadruple 2-input Positive Exclusive Or (XOR) 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.

3.4 Experiment

3.4.1 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.



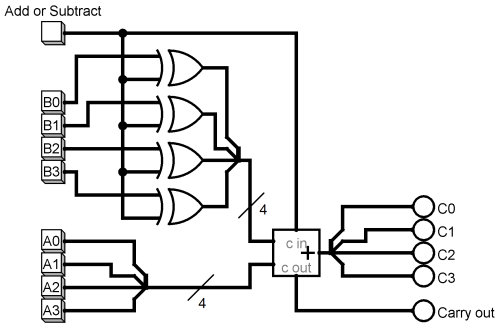
***Attention**: After you have completed this part, do not disassemble the circuit since you will reuse it in the next part.*

3.4.2 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.

3.4.3 Experiment - Part 3

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



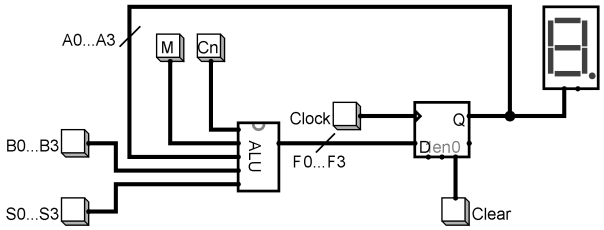
Test Case	1	2	3	4
A	0101	1101	1111	0110
B	0111	1001	1111	1101

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

3.4.4 Experiment - Part 4

Using a 4-Bit Arithmetic Logic Unit ($74xx181$) and D type flip-flops ($74xx174$) implement the circuit below. Use a debounced pushbutton for the clock input to prevent multiple entries. Also, since you will run out of logic switches, use the other debounced pushbutton for clear and SPDT switches for M and C_n. If some switches in your CADET do not work properly, you may connect B_3 bit to ground ($0V$).



Carry out the following operations one by one. Add a row to the table for each operation.
$A \leftarrow 0$
$A \leftarrow A + 1$
$A \leftarrow 5$
$A \leftarrow A - 1$
$A \leftarrow A + 3$
$A \leftarrow A + 2$
$A \leftarrow A - 4$
$A \leftarrow A \oplus 6$
$A \leftarrow A \times 2$

Inputs					Outputs	
A	B	$S_3S_2S_1S_0$	M	C_n	$F_3F_2F_1F_0$	C_{n+4}

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.
- 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.

4

TTL and CMOS Characteristics

4.1 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.

4.2 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.).

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)
- Integrated Circuits
  - 74xx00 TTL NAND Gates
  - 4011 CMOS NAND Gates
- Oscilloscope
- 100Ω Resistor

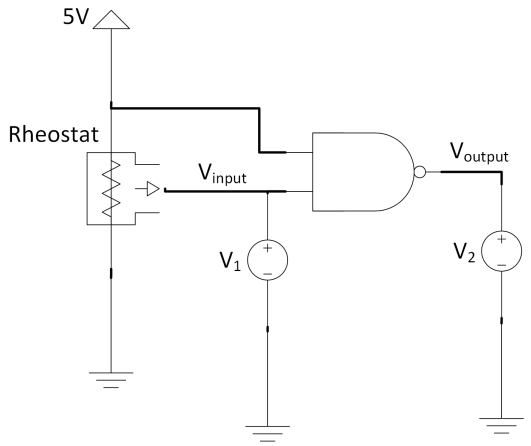
4.4 Experiment

4.4.1 Static Characteristics of TTL and CMOS NAND Gates

In the following parts of the experiment, you should to fill in the given characteristic tables (Tables 4.1, 4.2) by observing the voltages in the circuits you have conducted.

4.4.1.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
V_{input} (Volt)							
V_{output} (Volt)							

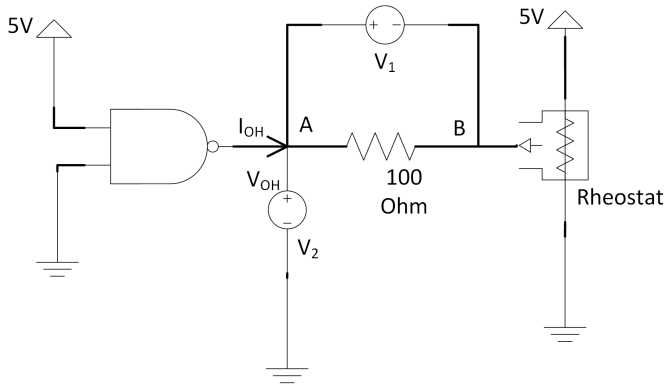
Table 4.1. Switching Characteristics of TTL

	1	2	3	4	5	6	7
V_{input} (Volt)							
V_{output} (Volt)							

Table 4.2. Switching Characteristics of CMOS

4.4.1.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 (4.3, 4.4) below.

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

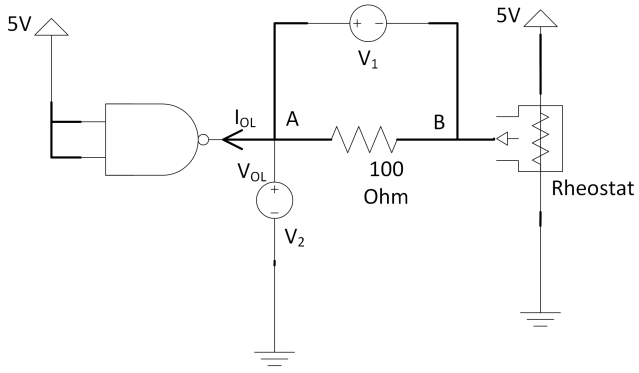
Table 4.3. V_{OH} - I_{OH} Characteristics of TTL

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

Table 4.4. V_{OH} - I_{OH} Characteristics of CMOS

4.4.1.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 (4.5, 4.6) below.

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

Table 4.5. V_{OL} - I_{OL} Characteristics of TTL

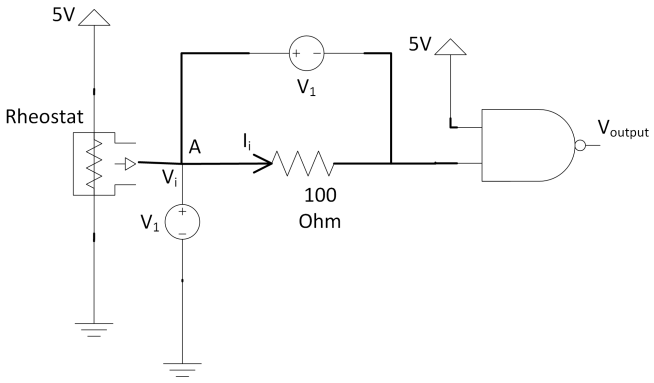
	1	2	3	4	5	6	7
V_{AB} (Volt)							
V_{OL} (Volt)							
I_{OL} (mA)							

Table 4.6. V_{OL} - I_{OL} Characteristics of CMOS

4.4.1.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 (4.7, 4.8) below.

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

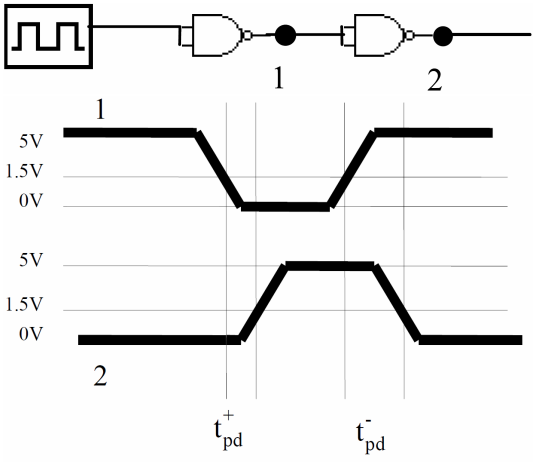
Table 4.7. V_i - I_i Characteristics of TTL

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

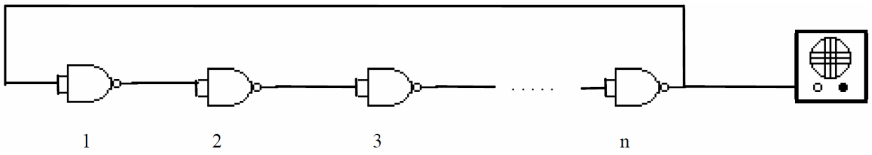
Table 4.8. V_i - I_i Characteristics of CMOS

4.4.2 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, 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.



4.4.3 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 4.9 and 4.10. 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 4.9. Power Consumption in TTL

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

Table 4.10. Power Consumption in CMOS

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 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.

5

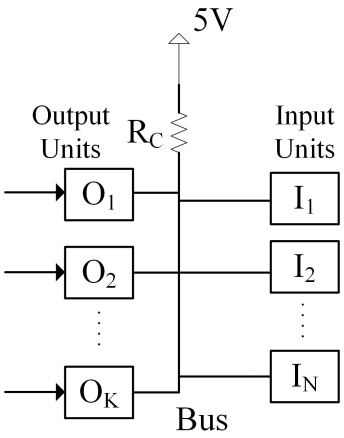
Data Bus Implementation in Digital Systems

5.1 Introduction

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

5.2 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.



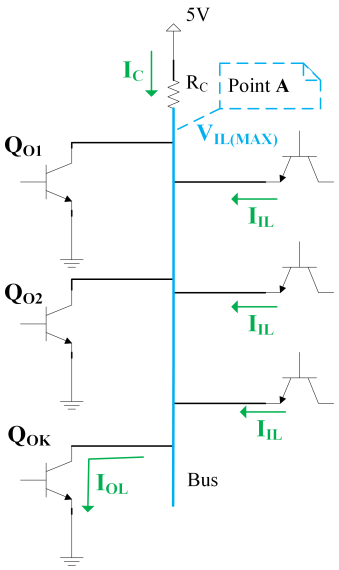
Lower and upper bounds of the resistor in the given design can be calculated by using the following equations.

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

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.

5.2.1 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.

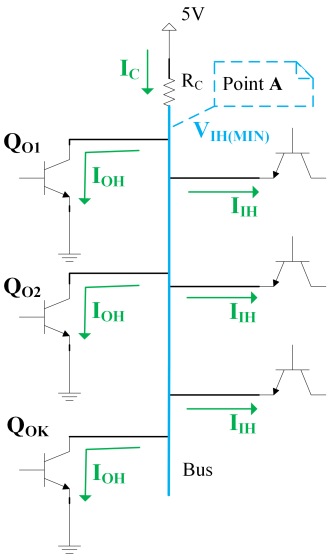


$$I_{OL(max)} = 8mA$$

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

5.2.2 Assigning Logic-1 to the Common Bus

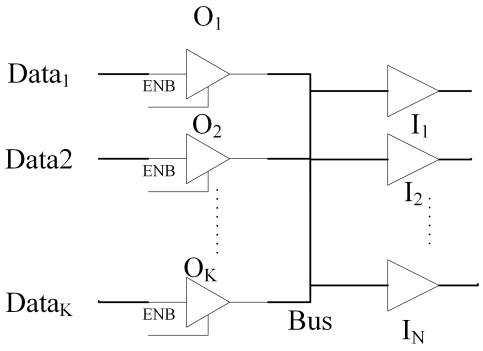
In order to acquire logic-1 in the common bus, all of the output transistors (Q_O) must be in the cut-off mode. An exemplary scenario is given in the figure on the right. In this circuit, all of the output transistors (Q_{OK}) is operating in the cut-off mode, thus the currents (I_{OH}) are flowing on 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 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.



5.3 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 5.2.1 and 5.2.2.

5.4 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
  - 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.

5.5 Experiment

5.5.1 Experiment - Part 1

You are going to implement 4-bit data bus by using ICs with open collector outputs. In this circuit, each bit on the bus is shared by two different bus driver units. The basic design of the circuit is given in the Figure 5.1. After the implementation of the circuit, observe the output of the bus by keeping in mind the behaviour of the "wired OR" circuit.

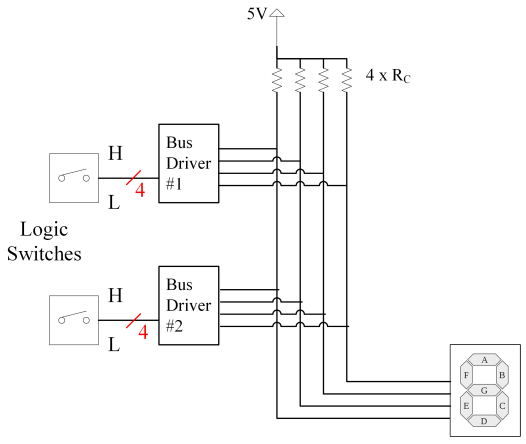


Fig. 5.1. 4 bit data bus with 2 drivers with open collector outputs

5.5.2 Experiment - Part 2

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 5.2. 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.

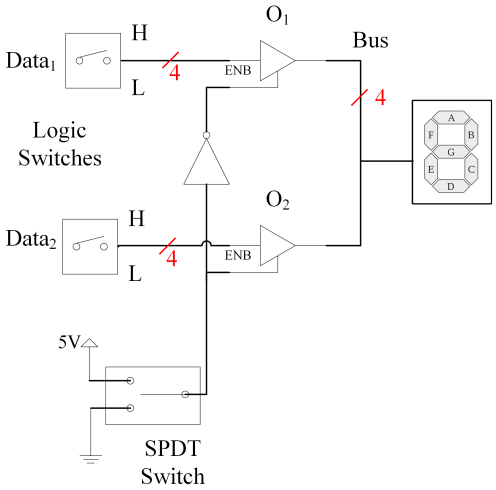


Fig. 5.2. 4 bit data bus with 2 drivers with 3-state buffers

5.5.3 Experiment - Part 3

As the last part of experiment, you need implement the circuit given in the Figure 5.3. 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.

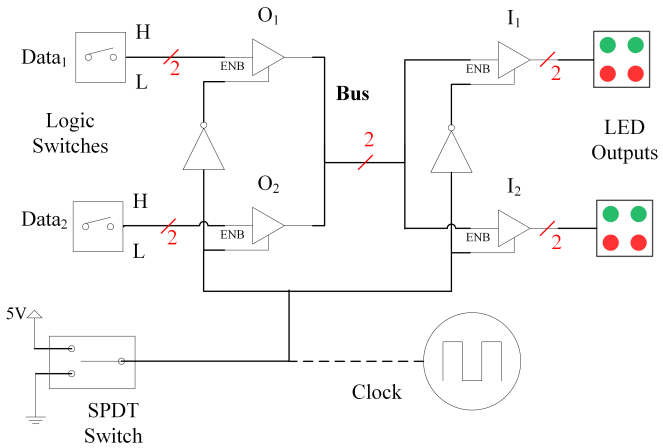


Fig. 5.3. 2 bit data bus with 2 drivers and 2 readers

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.

5.6 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.

6

Latches and Flip-flops

6.1 Introduction

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

6.2 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 6.4.4.

6.3 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
  - 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.

6.4 Experiment

6.4.1 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.

6.4.2 Experiment - Part 2

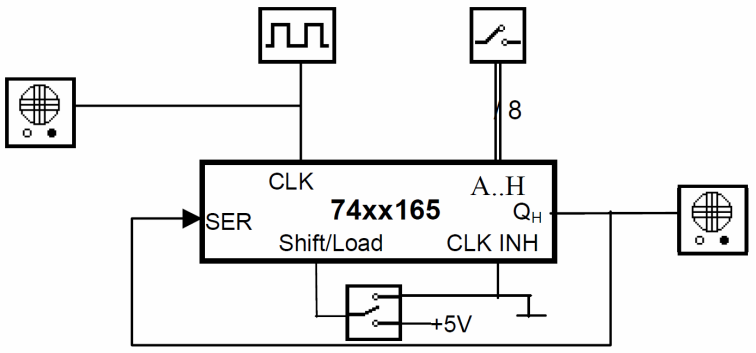
Implement a SR type latch with an enable input, C. Use only NAND gates. Use switches for S, R and C inputs and LEDs for Q and Q_N outputs. Create a truth table for the latch by testing all possible input combinations. Note how the latch behaves for disallowed inputs and how enable input effects the output.

6.4.3 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.

6.4.4 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

6.5 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.

7

Sequential Logic Circuits

7.1 Introduction

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

7.2 Preliminary

- Refresh your knowledge on how to design circuits with Mealy and Moore models.
- Analyze the circuit in 7.4.1 to create a state transition table and a state chart.
- Design and draw the circuit to implement in 7.4.2 with the minimum number of logic gates.
- Study how 74xx161 integrated circuit works. Draw the circuit to implement in 7.4.3.

7.3 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
  - 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

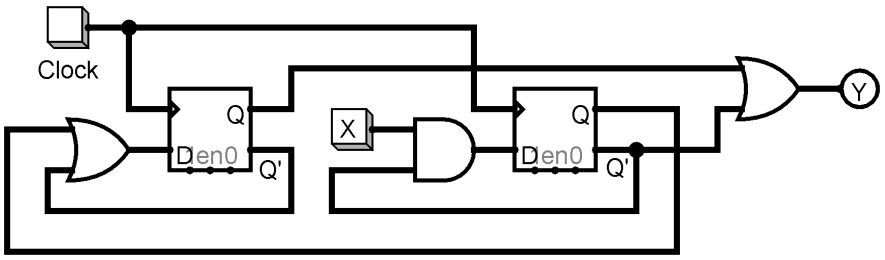
Fundamental information (function tables and pin configurations) of the ICs listed above are given in the Appendix A. You should also examine the data-sheets in order to acquire further information about these ICs.

7.4 Experiment

7.4.1 Experiment - Part 1

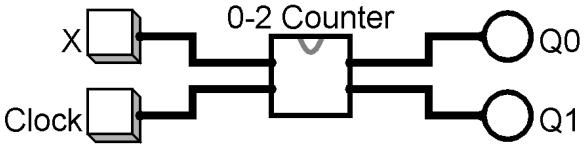
Build the circuit below. X input should be connected to a switch, while the clock should be given using a debounced pushbutton. Use LEDs to examine

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



7.4.2 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 up (0-1-2-0-1-2-...) while $X = 1$ means counting down (2-1-0-2-1-0-...).



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.

7.4.3 Experiment - Part 3

Using the $74xx161$ integrated circuit and other necessary gates, implement a counter that counts 0 to 9 in a circular way. Demonstrate the output in the seven segment display.

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:

- Give the steps of your analysis for the circuit in 7.4.1.
- Give the steps of your design for the circuit in 7.4.2.
- Discuss the behaviour of your circuit in 7.4.2 for the undetermined state.
- Draw the circuit you implemented in 7.4.3 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.

8

Design and Implementation of Serial Arithmetic Logic Unit

8.1 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.

8.1.1 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 8.1.

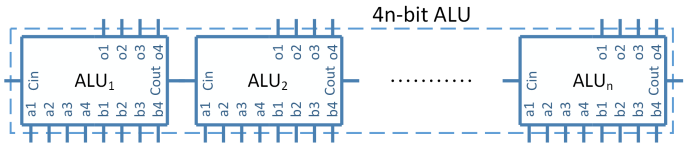


Fig. 8.1. Extending bandwidth of an ALU in parallel

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.

8.2 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 8.1. 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.

8.3 Equipments and Integrated Circuits (ICs)

- C.A.D.E.T.
- 74000 series ICs [1]

8.4 Experiment

8.4.1 Experiment - Part 1

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

8.4.2 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.

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:

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

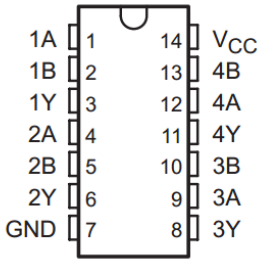
During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in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

A

Data Sheets

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

Pin Configuration

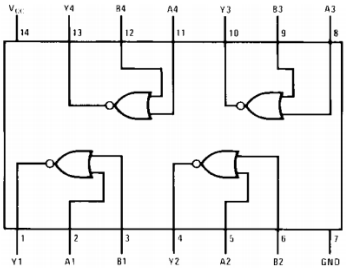


Function Table

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

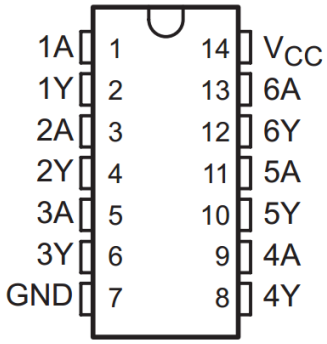
$$Y = \overline{A + B}$$

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

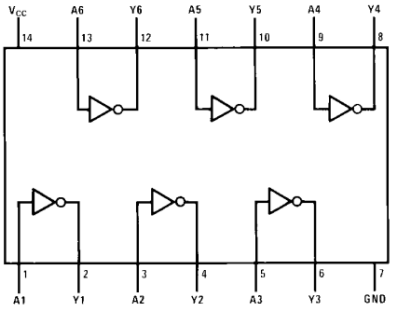


Function Table

INPUT A	OUTPUT Y
H	L
L	H

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

Pin Configuration



Function Table

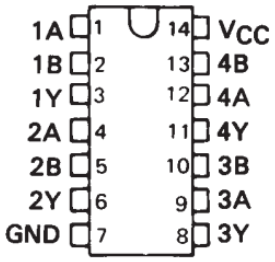
$Y = \overline{A}$

Input A	Output Y
L	H
H	L

H = HIGH Logic Level
L = LOW Logic Level

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

Pin Configuration

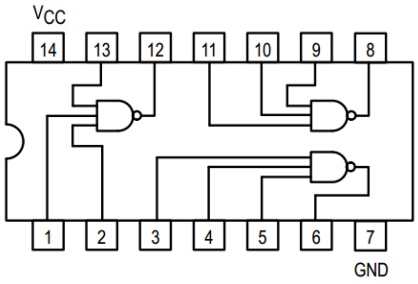


Function Table

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

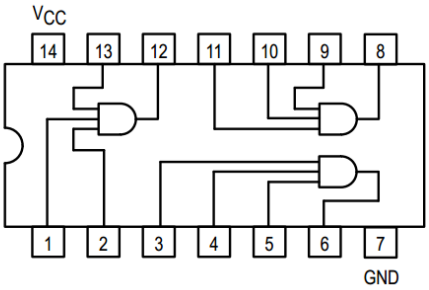
A.6 7410 - Triple 3-input NAND Gates

Pin Configuration



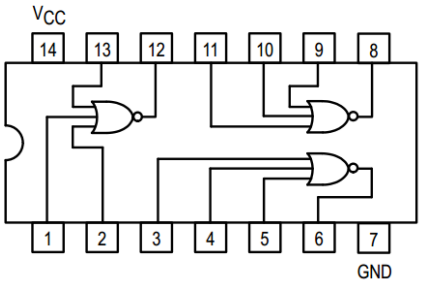
A.7 7411 - Triple 3-input AND Gates

Pin Configuration



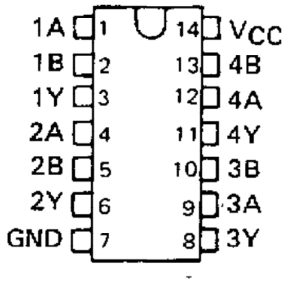
A.8 7427 - Triple 3-input NOR Gates

Pin Configuration



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

Pin Configuration

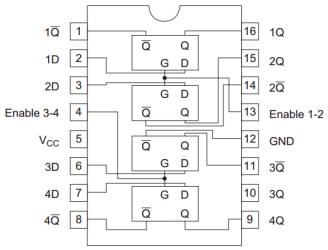


Function Table

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	$\overline{Q}$
L	H	L	H
H	H	H	L
X	L	Q_0	$\overline{Q}_0$

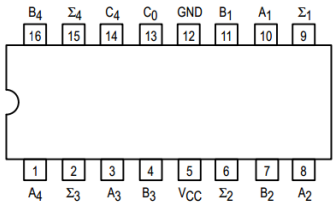
H; high level, L; low level, X; irrelevant

Q_0; level of Q before the indicated steady-state input conditions were established.

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

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

Pin Configuration



Function Table

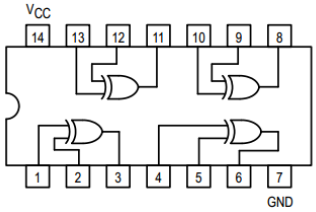
FUNCTIONAL TRUTH TABLE

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

C_1 — C_3 are generated internally
C_0 — is an external input
C_4 — is an output generated internally

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

Pin Configuration



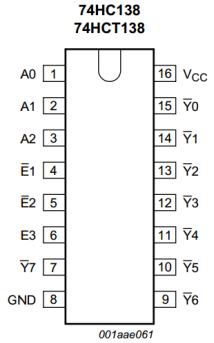
Function Table

TRUTH 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



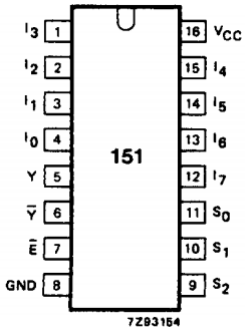
Function Table

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	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	L	H	L	H	H	H	H	H	H
			H	H	H	L	H	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

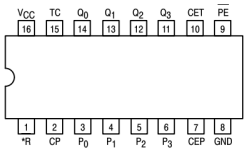


Function Table

INPUTS												OUTPUTS	
E	S_2	S_1	S_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	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	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	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	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
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	H	L	H

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

Pin Configuration

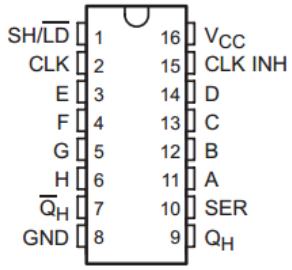


Function Table

*SR	PE	CET	CEP	Action on the Rising Clock Edge (⌐)
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

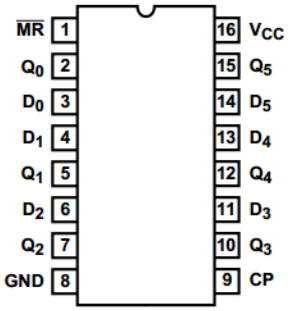
FUNCTION TABLE

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



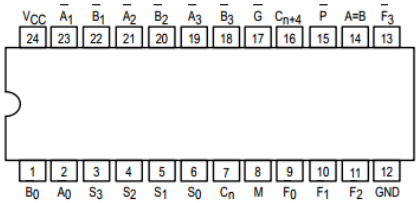
Function Table

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



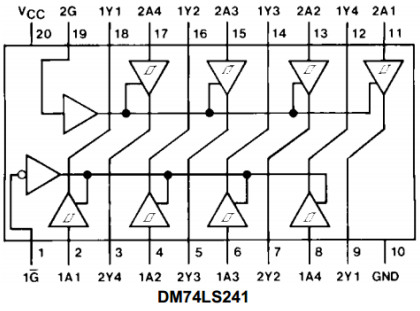
Function Table

MODE SELECT INPUTS				ACTIVE LOW INPUTS & OUTPUTS		ACTIVE HIGH INPUTS & OUTPUTS	
S_3	S_2	S_1	S_0	LOGIC (M = H)	ARITHMETIC** (M = L) (C_n = L)	LOGIC (M = H)	ARITHMETIC** (M = L) (C_n = H)
L	L	L	L	$\overline{A}$	A minus 1	$\overline{A}$	A
L	L	L	H	$\overline{AB}$	AB minus 1	$\overline{A+B}$	A + B
L	L	H	L	A + $\overline{B}$	AB minus 1	AB	A + B
L	L	H	H	Logical 1 minus 1		Logical 0 minus 1	
L	H	L	L	$\overline{A+B}$	A plus (A + B)	$\overline{AB}$	A plus AB
L	H	L	H	$\overline{B}$	AB plus (A + B)	B	(A + B) plus AB
L	H	H	L	$\overline{A \oplus B}$	A minus B minus 1	$\overline{A \oplus B}$	A minus B minus 1
L	H	H	H	$\overline{A+B}$	A + B	$\overline{AB}$	AB minus 1
H	L	L	L	AB	A plus (A + B)	$\overline{A+B}$	A plus AB
H	L	L	H	$\overline{A \oplus B}$	A plus B	$\overline{A \oplus B}$	A plus B
H	L	H	L	B	AB plus (A + B)	B	(A + B) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logical 0	A plus A*	Logical 1	A plus A*
H	H	L	H	AB	$\overline{AB}$ plus A	A + B	(A + $\overline{B}$) plus A
H	H	H	L	AB	AB plus A	A + B	(A + B) Plus A
H	H	H	H	A	A	A	A minus 1

L = LOW Voltage Level
H = HIGH Voltage Level
*Each bit is shifted to the next more significant position
**Arithmetic operations expressed in 2s complement notation

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

Pin Configuration



Function Table

Inputs				Outputs	
G	$\overline{G}$	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		Z

A.20 4011 - Quad 2-Input NAND Gate

Pin Configuration