OSU Mechanical Engineering

Smart Products Laboratory

Mixed Signal Hardware Design | Five

ME Course Number
Spring 2019

Table of Contents

List of Figures

1 Overview

The fifth lab is purely hardware based and will help develop the fundamental skills needed to implement mixed signal algorithms. This lab will combine the domains of analog and digital electronics to develop a basic smart detection system. In this lab you will be given three constant reference voltages and an unknown voltage signal. The goal is to identify which reference is closest in voltage to the unknown signal at any point in time.

2 Background

The information below should provide a basic introduction to the concepts needed to complete this lab.

2.1 Definitions

The key concepts for this lab are defined and demonstrated below

- **Digital Logic:** Digital logic is the category of electronics that completes operations based solely on the logic levels of HIGH and LOW.
    - Common elements for digital logic circuits are: AND, OR, and NOT
- **Analog Circuits:** any circuit element that works with a continuous signal
    - Common elements consist of op-amps, etc.
- **Truth tables:** a table that consists of all the possible combinations of a logic-based circuit.

The following examples and information will be helpful when deriving and building your solution circuit

2.1 Mixed Signal Design Case

Mixed signal circuits combine analog circuit elements, such as op amps, with digital circuit elements, such as logic gates, to derive logical decisions about physical signals. Additional elements such as diodes, resistors, and switches can be used to manipulate both digital and analog signals. As a smart product engineer, you may find yourself having to implement such circuits in disciplines including automation, robotics, manufacturing, vehicle engineering, or any embedded product requiring power management.

Shown below is a simple example of how a mixed signal circuit may be applied to monitor different physical variables of a chemical process. Here temperature and pressure transducers measure the chemical process and generate a proportional voltage signal. The process is designed so that these physical variables shall not exceed predetermined thresholds. If they do, an alert should be triggered. The following example will demonstrate how to break down this type of problem and solve it logically.

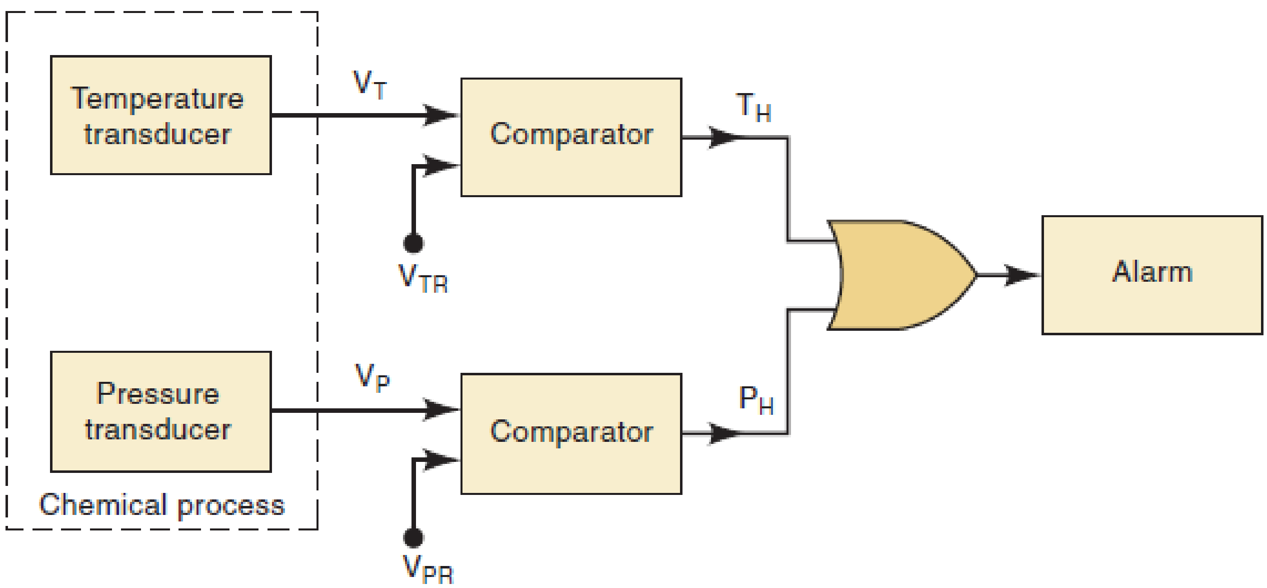


Figure 1 – Example Mixed Signal Logic Circuit

Beginning at the left side, the circuit is broken down algorithmically to demonstrate the underlying logic. From this algorithm it then becomes clear how to make a truth table.

Given a temperature signal, V_T, as well as a pressure signal, V_P, and given reference signals, V_{TR} and V_{PR}, let D_T, D_P be the digital states produced from the comparators of the temperature signals and the pressure signals, respectively. Let D_A be the digital state of the alarm.

//Comparator for temperature
IF $V_T > V_{TR}$
: $D_T = HIGH$
ELSE
: $D_T = LOW$
END
//Comparator for pressure
IF $V_P > V_{PR}$
: $D_P = HIGH$
ELSE
: $D_P = LOW$
END
//Digital OR element to compare the digital states D_T, D_P
IF $D_T == HIGH$ **OR** $D_P == HIGH$
: $D_A = HIGH$
ELSE
: $D_A = LOW$
END
//Alarm element
IF $D_A == HIGH$
: $Trigger\ Alarm!$
ELSE
: $Disable\ Alarm$
END

Table 1 – Truth table for chemical process example

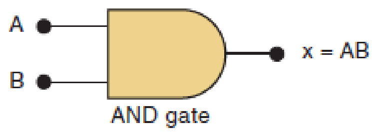
V_T	V_P	D_T	D_P	D_A	Alarm
$> V_{TR}$	$> V_{PR}$	1	1	1	Triggered
$> V_{TR}$	$< V_{PR}$	1	0	1	Triggered
$< V_{TR}$	$> V_{PR}$	0	1	1	Triggered
$< V_{TR}$	$< V_{PR}$	0	0	0	Disabled

2.2 Description of Common Mixed Signal Circuit Elements

AND

A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

(a)



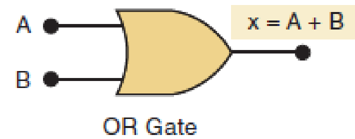
(b)

Figure 3 – AND logic gate and truth table

OR

A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(a)



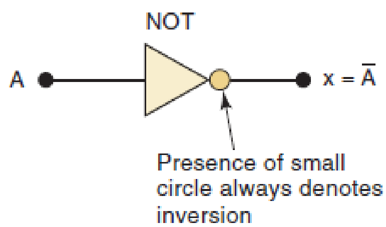
(b)

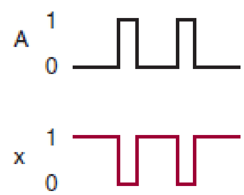
Figure 4 – OR logic gate and truth table

NOT

A	$x = \overline{A}$
0	1
1	0

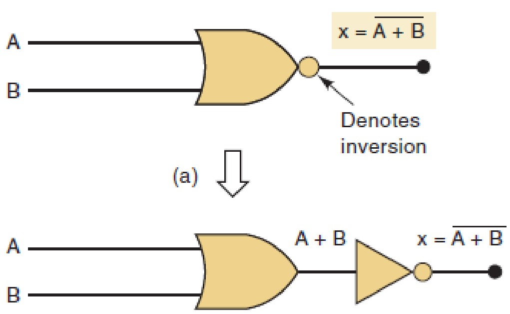
(a)





(c)

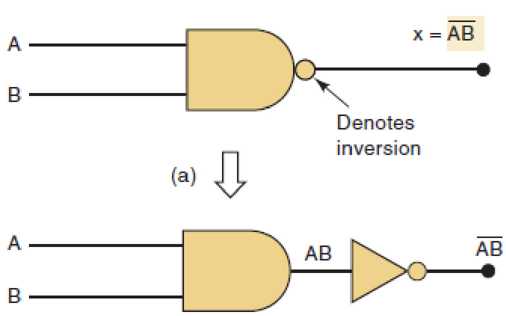
Figure 2 – NOT (inversion) logic gate and truth table



		OR	NOR
A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(c)

Figure 5 – NOR logic gate and truth table

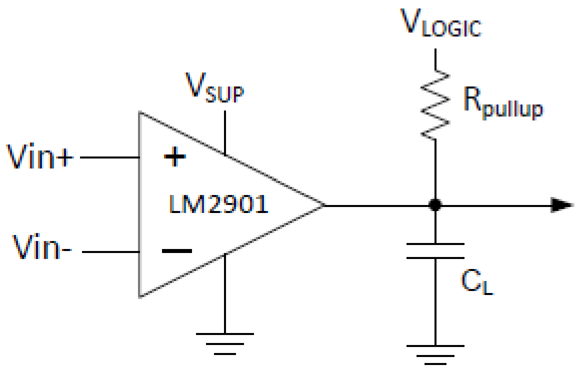


(b)

		AND	NAND
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

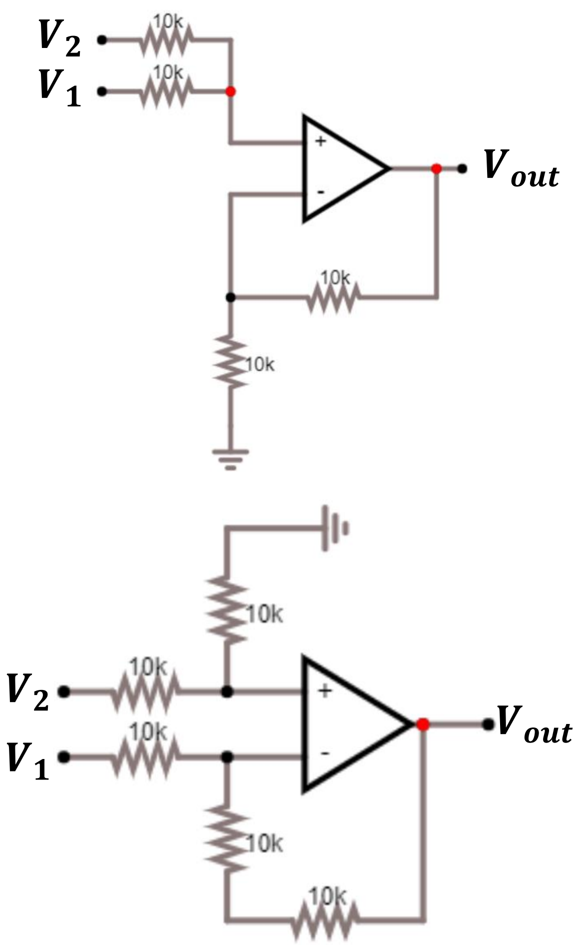
(c)

Figure 6 – NAND logic gate and truth table



IF $V_{in+} > V_{in-}$
: $D_T = V_{Logic}$
ELSE
: $D_T = 0$
END

Figure 8 – Example Comparator



Summation

$$V_{out} = V_2 + V_1$$

Subtractor

$$V_{out} = V_2 - V_1$$

Figure 7 – Summing and Subtracting Configurations

Example:
Goal is to turn on either the top or bottom LED while keeping the middle LED on.

Wrong without diodes	***Right with diodes***

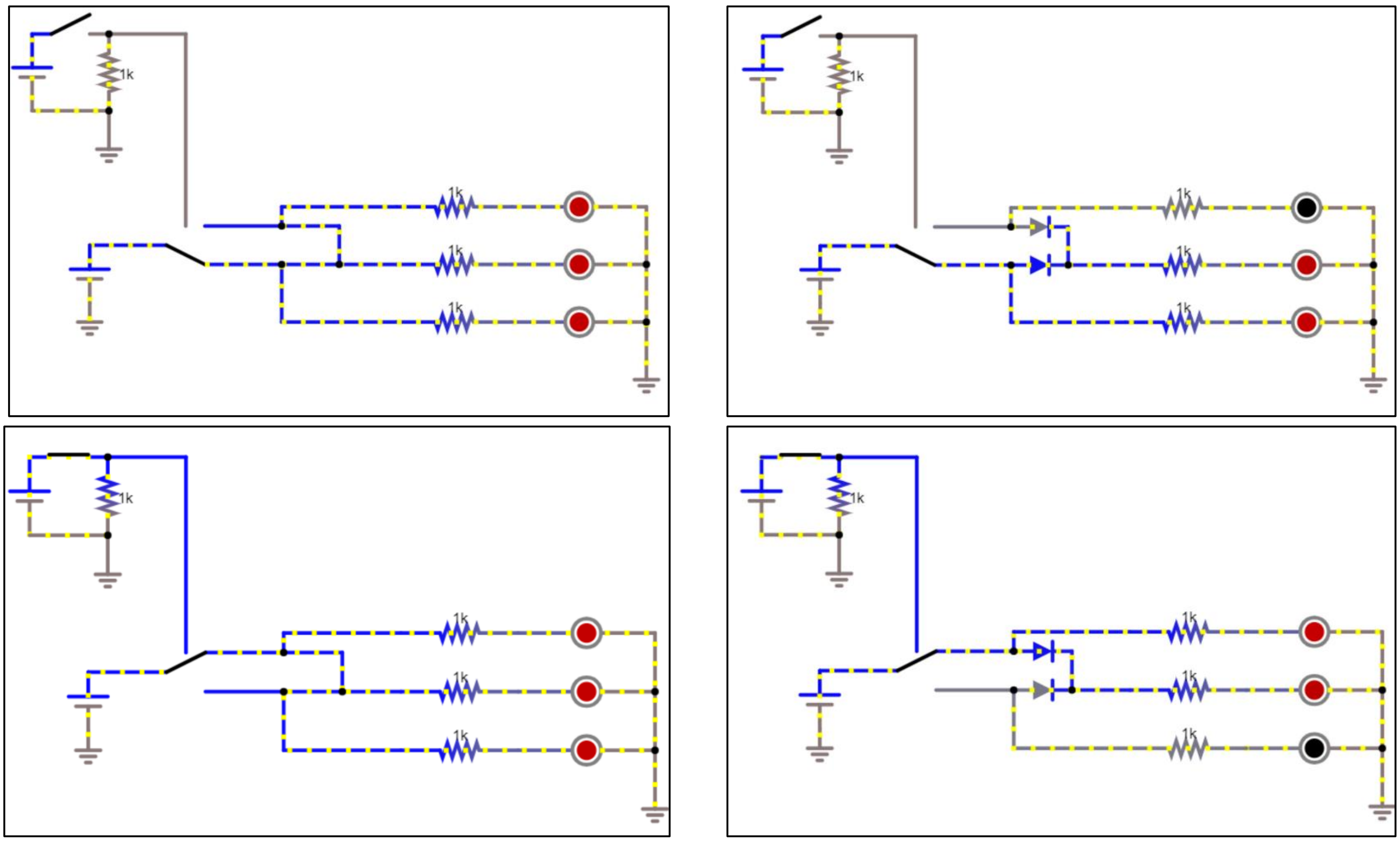


Figure 9 – How to properly use diodes to control switches

3 References & Resources

3.1 References

The following references may be helpful to complete the lab.

1. Op-amps: https://www.electronics-tutorials.ws/opamp/op-amp-building-blocks.html
2. Pull up resistors: https://www.electronics-tutorials.ws/logic/pull-up-resistor.html
3. Digital logic summary: https://www.electronics-tutorials.ws/logic/logic_10.html
4. More digital logichttps://www.electronics-tutorials.ws/category/logic
5. Truth tables summary: https://www.electronics-tutorials.ws/boolean/bool_7.html
6. Truth tables examples: https://www.electronics-tutorials.ws/boolean/bool_8.html

Additional resources for design:

7. http://e2e.ti.com/blogs_/b/analogwire/archive/2014/03/21/what-you-need-to-know-about-using-general-purpose-op-amps-at-low-voltage
8. http://www.ti.com/logic-circuit/overview.html
9. https://www.analog.com/media/en/training-seminars/tutorials/MT-214.pdf
10. https://www.analog.com/media/en/training-seminars/tutorials/MT-083.pdf
11. https://www.analog.com/media/en/training-seminars/tutorials/MT-088.pdf

4 Pre-lab

This section should be worked on prior to arriving in lab.

4.1 Procedure

Make sure you read this entire lab.

5 Laboratory

Complete the exercises and save all data and follow the procedure to turn in your work.

5.1 Requirements

1. The following IC chips are available for building your solution, the data sheets can be found at these links or in carmen:

Component	Link	Part Number
NOR	https://www.digikey.com/short/pt9c77	SN74HC02N
NAND	https://www.digikey.com/short/pth1f5	SN74HC00N
AND	https://www.digikey.com/short/pt9ccp	SN74HC08N
Inverter	https://www.digikey.com/short/pth1hp	CD40106BE
XOR	https://www.digikey.com/short/pth1rm	SN74HC86N
Switches	https://www.digikey.com/short/pt9c0n	CD74HC4053E
Comparators	https://www.digikey.com/short/pt9345	LM339N

Op Amp	https://www.digikey.com/short/pt93fn	LM324AN
Diodes	https://www.digikey.com/short/pt98cq	1N4001-G
10k Resistor	https://www.digikey.com/short/pt98hv	CF14JT10K0

2. You will also need the following to build and test your solution:
    a. Bread boards
    b. Wires
    c. Possibly capacitors
    d. DC power source for powering your electronics.
    e. Oscilloscope for probing and testing your digital logic.
    f. USB memory stick
    g. 3: LEDS
    h. 3: 100 Ohm– 250 Ohm resistors for your LEDS
    i. 4: 1k Ohm resistors
    j. 1: ~500 Ohm resistor (or something close)
    k. 1: ~2k Ohm resistor (or something close)

5.2 Procedure

1. When initially solving the problem its highly recommended that you use the free online circuit simulator at https://www.falstad.com/circuit/circuitjs.html. It is also recommended that you use this site to draw the schematic of your final solution.
2. It may be helpful to break down the problem into the following sections
    a. Analog signal processing – the part of the circuit that initially conditions all your signals to be later fed into the comparators. This includes the use of switches to direct voltage signals, diodes to direct current, resistors and op amps to manipulate multiple signals, and capacitors to steady constant signals.
    b. Analog-Digital interface – the part of the circuit which converts the conditioned signals into digital logic states. At this point you will have conditioned the signals to be able to compare which reference is closest to the unknown signal.
    c. Digital logic – the part of the circuit which deduces which reference is closest to the unknown signal and ultimately sets its corresponding LED high.
3. Use this breakdown to create an algorithm to solve the problem. In this algorithm make sure the use the naming convention given in the initial setup circuit in Figure 10
4. From this algorithm create a truth table.
5. From the truth table select the proper components that produce the same logic.
6. Just as you should incrementally test your software in small chunks, you should also incrementally build smaller parts of your circuit and test for proper operation.
7. **USE NO MORE THAN 5V AND 0.08 AMPS TO POWER YOUR CIRCUIT**
8. Make sure make proper use of diodes to control the direction of current.
9. When setting up the resistor network create three voltage dividers: in each case use the 1k Ohm resistor as the bottom resistor (the one that connects to ground). Follow the circuit in Figure 10 to setup the input and output circuits.

10. When testing your reference signals use the signal generator on the back of the oscilloscope.
11. When capturing data from your circuit use a USB memory stick to save all signals.
12. When finished with your circuit find the GTA for them to test out your solution to various waveforms.
13. When finished with the exercises in the next section, you will then generate a brief report and submit your work on Carmen. The submission will be in the form of a zip file and must conform to the following requirements:
    a. The zip file should be named as follows
        i. *SP19_lab05_Lastname_dotnumber.zip*
    b. Inside the zip files should be only the following
        i. One schematic of the final solution made with any software of your choosing. Hand drawn circuits will not be accepted. In pdf or png form.
        ii. One report in PDF format which will contain only the following in 6 pages or less,
            1. Date, Name, Class, and Lab number(s) at the top.
            2. Section one:
                a. Present and describe the algorithm you developed for your solution.
                b. Present your truth table and briefly describe how you used it to select the proper components.
                c. Describe which components you used and why you used them.
                d. Detail any other methodology you used to solve the problem.
                e. How did you use incremental testing to efficiently build your solution?
                f. If having to do it over again, do you think you could build the solution with less components? If so, explain.
            3. Section Two
                a. Include all graphics captured from your testing with the oscilloscope and briefly explain what you are testing for in each image.
                b. Present the measurements, graphics, and explanations from the exercises.
                c. Include a picture of your final built circuit.

5.3 Exercises

Create a network of analog signal operations and digital logic to solve the following problem. Given three constant reference voltages, $(V_{R_1}, V_{R_2}, V_{R_3})$, and an unknown voltage signal, V_S, determine which of the reference voltages are closest to the voltage of the unknown signal. You may use only the components presented in the hardware requirements section and may not use a microcontroller or software to solve this problem. The output of the circuit will result in turning one of the three digital indicator lines, $(D_{R_1}, D_{R_2}, D_{R_3})$ HIGH which in turn will light an LED. See **Error! Reference source not found.**, Figure 11, Figure 12, and Figure 13 for details.

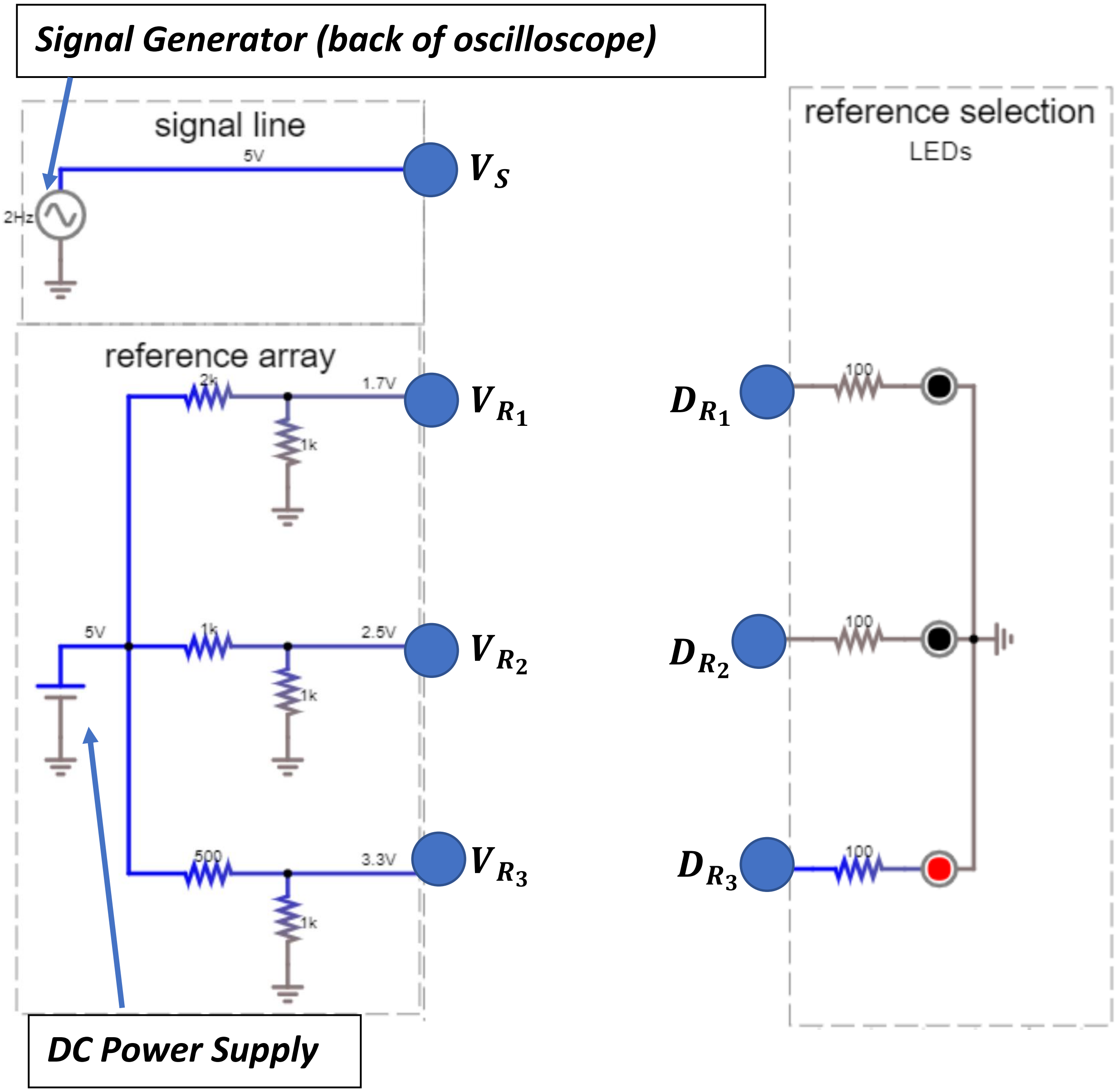


Figure 10 – Initial circuit setup for lab 5 and naming convention

When building your circuit use the digital and analog probes on the oscilloscope to capture your solution and measuring the following characteristics

1. Measure the time it takes for your circuit to switch digital indictor lines. For example, starting at the instant that V_S becomes closer to V_{R_1} than to V_{R_2} (as in bottom of Figure 11), how long does it take for D_{R_2} to go LOW and D_{R_1} to go HIGH. Capture this transition using analog probes.
2. Run your circuit through several frequencies of a sinusoidal signal. What is the minimum frequency of the signal line which causes multiple LEDs to illuminate at once? Probe and capture the digital indictor lines and describe the signals that are being produced. Explain why multiple LEDs are illuminated. What other microcontroller process causes this type of phenomena?

6 Rubric

Exercise	*Performance*	*Completeness*	*Cleanliness*	*Clarity*	*Total*
Circuit	/ 100	/ 100	/ 10	/ 10	/ 220
Report	/ 50	/ 100	-	/30	/ 180
Total	**/ 150**	**/ 200**	**/10**	**/40**	**/400**

1. **Performance**– Does your circuit produce the correct results and are your reported measurement correct.
2. **Completeness**– How much of the exercise did you complete.
3. **Cleanliness –** Of the completed portion of the exercise how well organized and structured is the work (i.e. is your circuit designed efficiently in a compact and clean manner, does your report flow from one idea to another)
4. **Clarity –** Of the completed portion of the exercise, how easy is it to infer what is occurring from the context of your work (i.e. did you explain your work/methodology in your report).

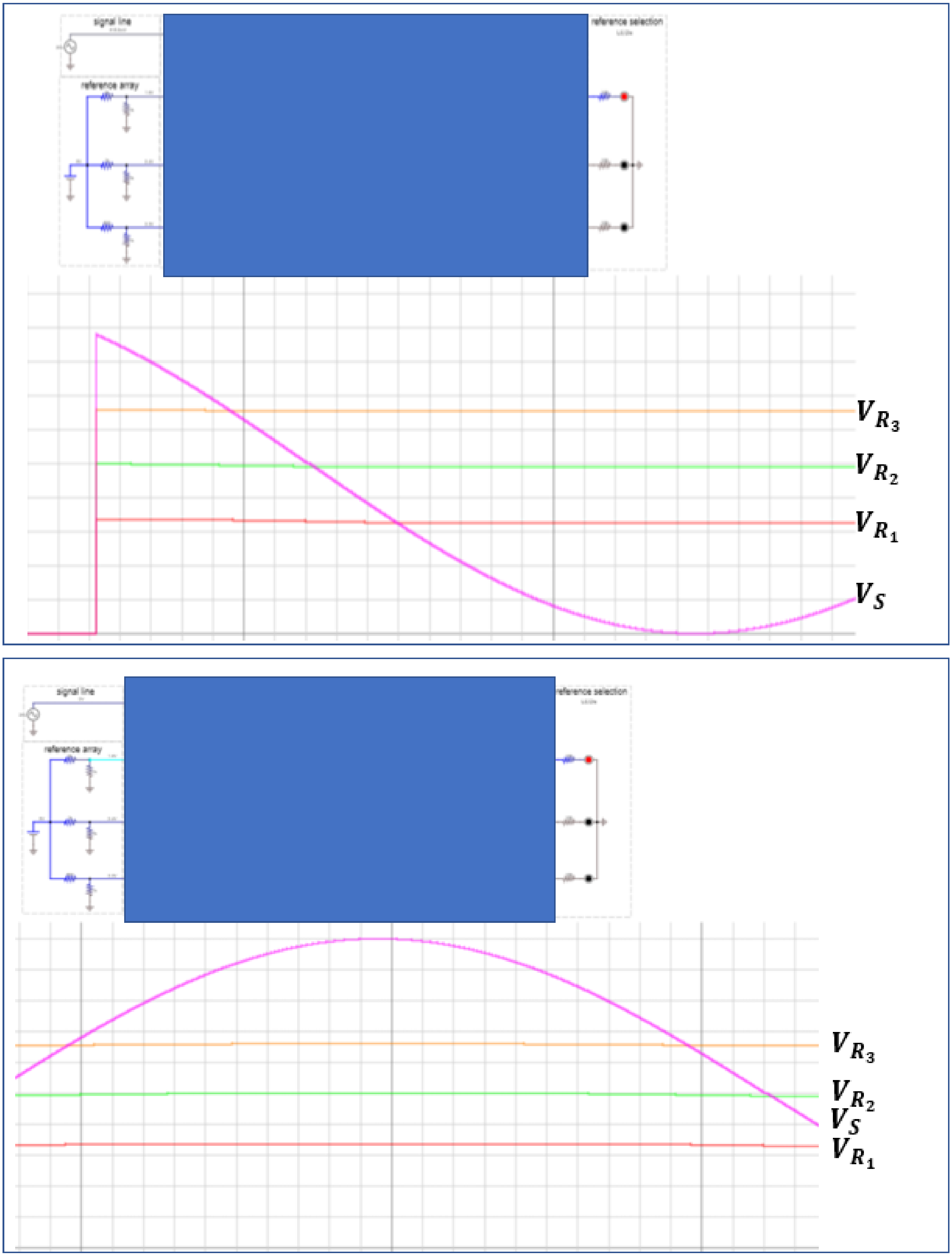


Figure 11 – Examples of Signal and References with D_{R_1} high

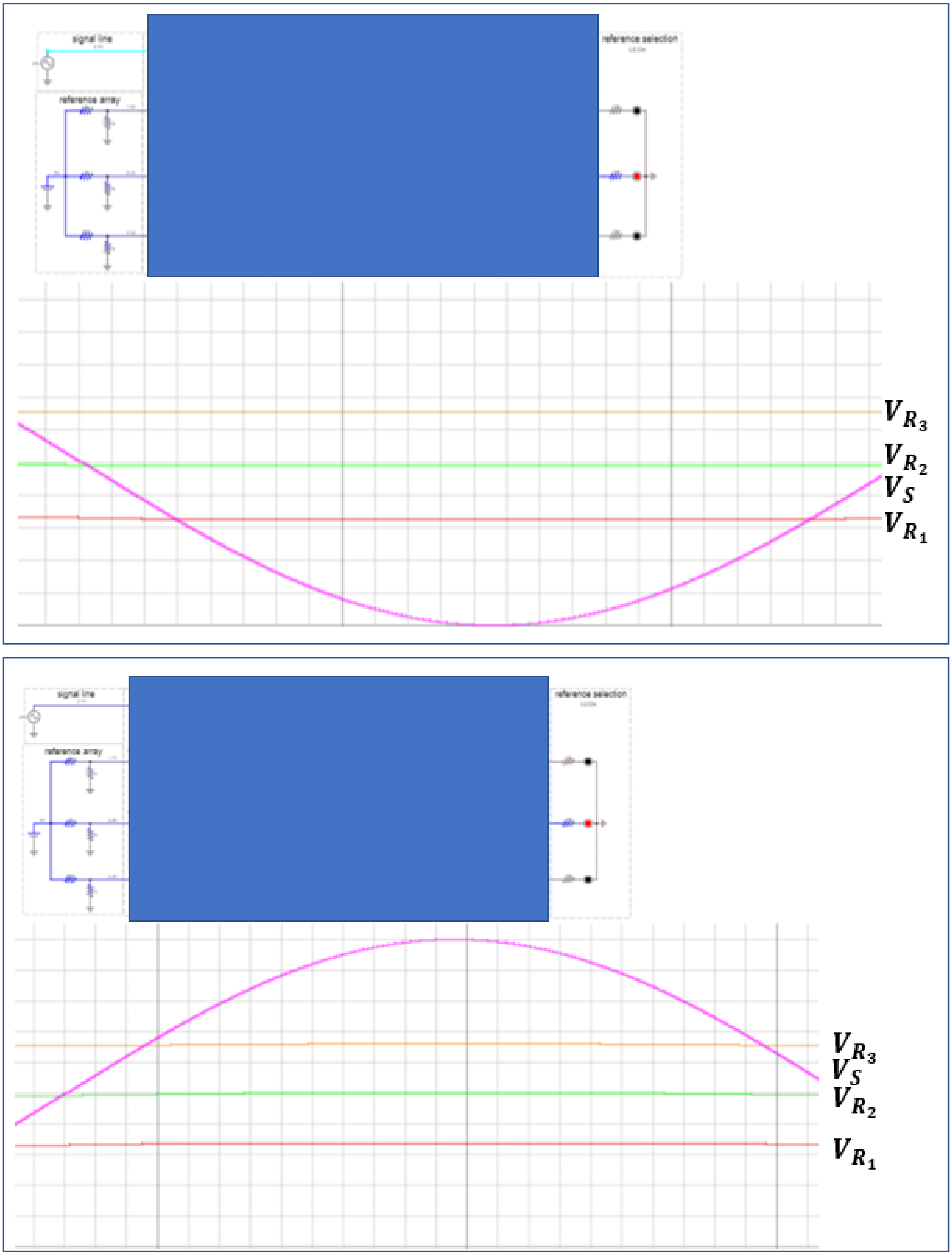


Figure 12 – Examples of Signal and References with D_{R_2} high

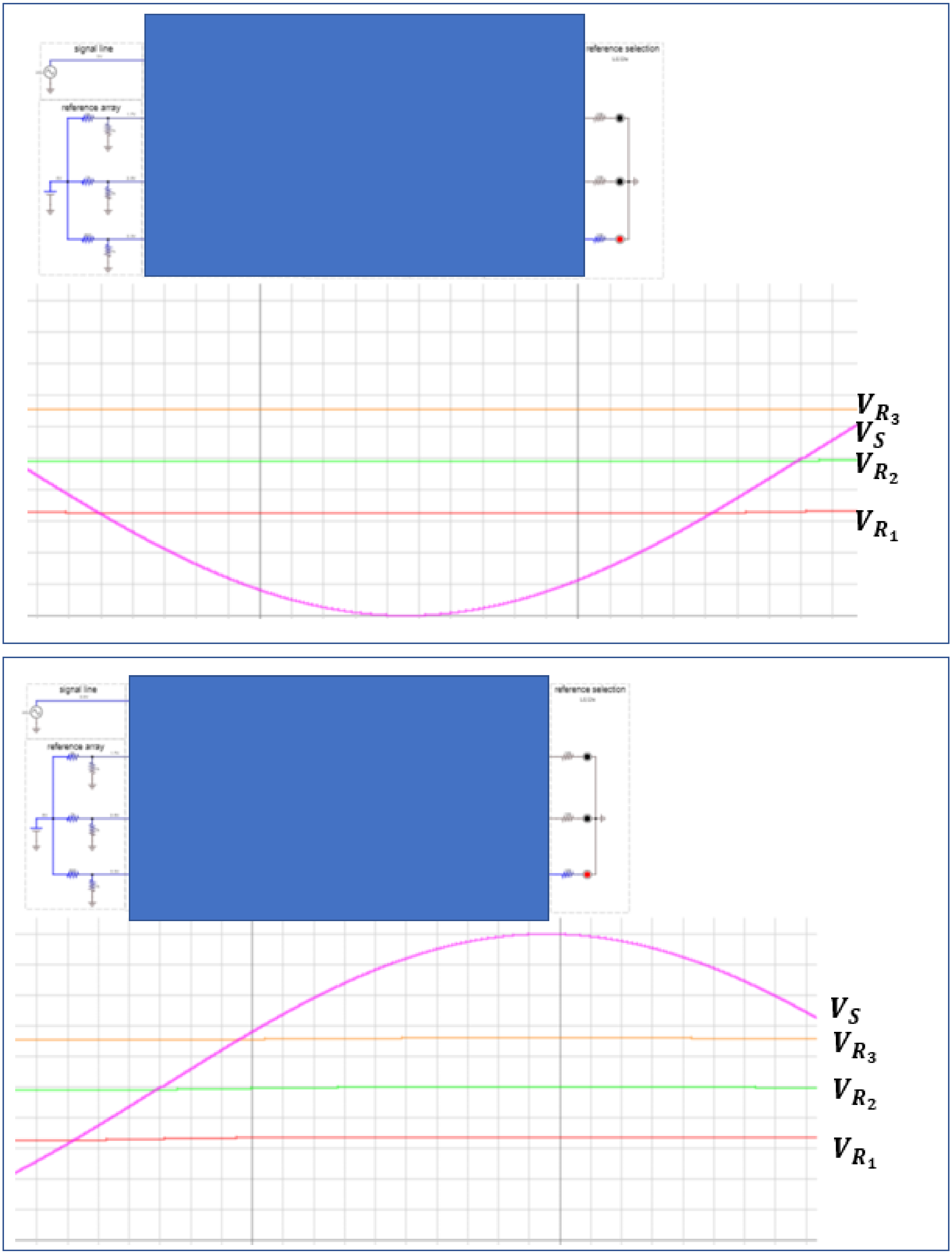


Figure 13 – Examples of Signal and References with D_{R_3} high

Appendix

$V_{IH} < V_{OH}$ ***and*** $V_{IL} > V_{OL}$ ***ensures proper system operation***

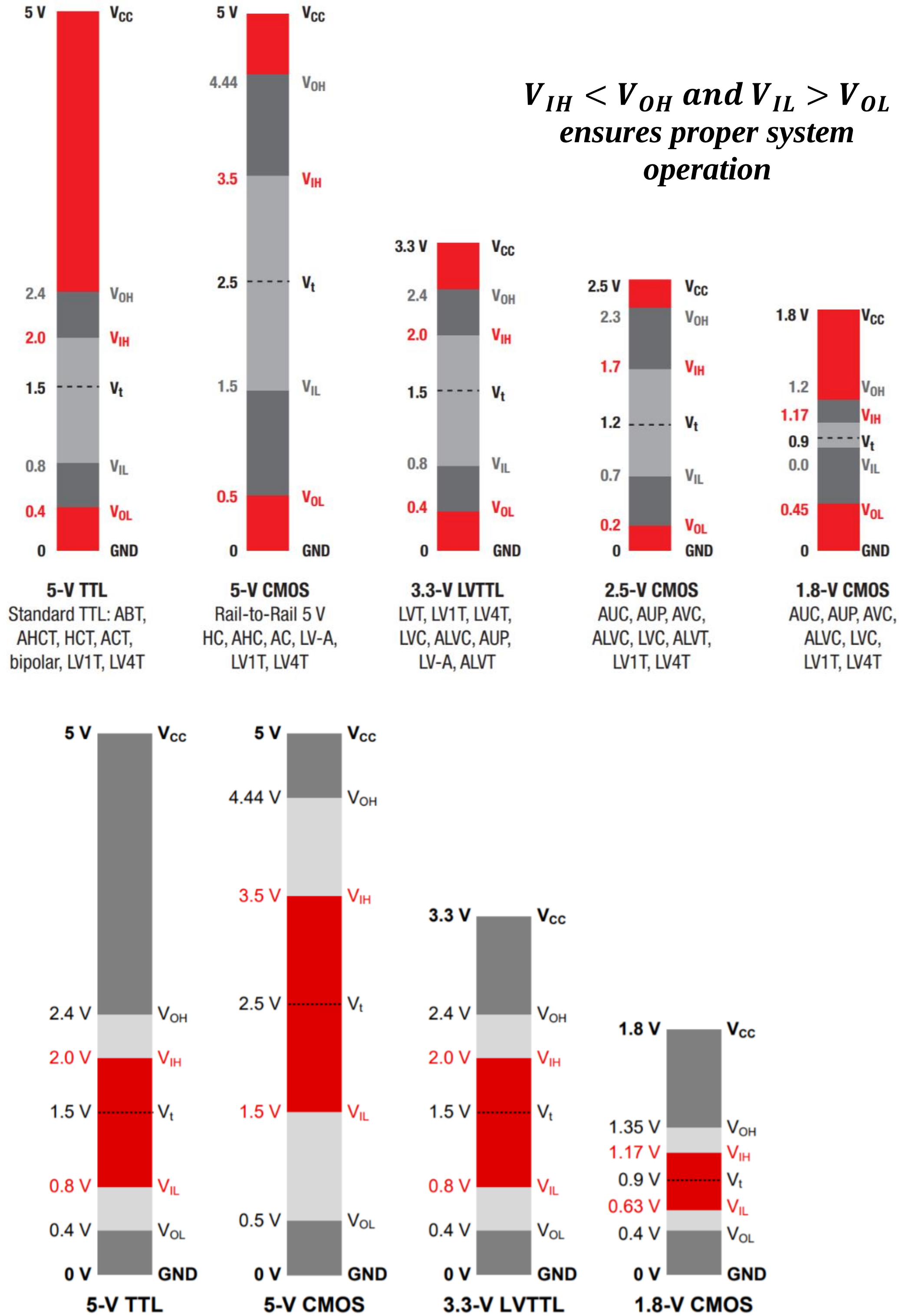


Figure 14 – Logic Level Thresholds