

# PACE Electrical Engineering

## Sequential Logic

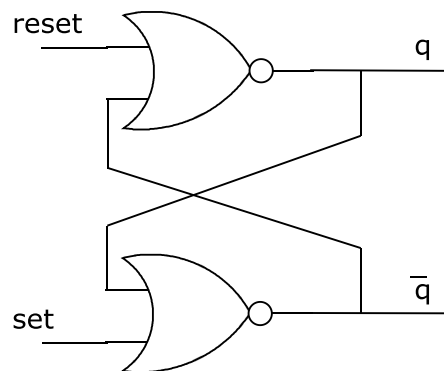
### Introduction

We have not used this term before, but all of the logic circuits we have studied so far are called **combinational logic** circuits. They all satisfy the following condition: The circuit output at any instant depends only on the circuit inputs at that instant. It does not matter what the inputs had been. If you know what the inputs are then you have all the information necessary to determine the output.

Not all circuits behave that way. For some their output depends on more than just their inputs at that instant. An example is the Quiz Game Response Light circuit. For that circuit, the output, the status (on or off) of the contestant lights, depends not just on the inputs (whether or not the contestant buttons are being pushed) at the moment, but on which button had been pushed first, and on whether either button had been pushed before. So, the Quiz Game Response Light is not a combinational logic circuit. It's a type of circuit called a **sequential logic** circuit.

### State

The output of sequential logic circuits depend not only on their inputs but also on the **state** the circuits were in when the inputs were applied. The state of a circuit is the value of internal signals<sup>1</sup>. The state usually depends in some way on previous inputs. This is achieved by feeding the output back to inputs of gates in the circuit. For example, look at the following circuit:



This circuit is called an RS latch. Let's see how the value of the outputs depends on more than just the inputs. We'll do that by showing that for the same set of inputs it's possible to get two different outputs. Analyzing a sequential logic circuit is tricky. To see why, let's start with inputs of  $\text{reset} = 0$  and  $\text{set} = 0$ . We can't determine what the outputs of each of the NOR gates are because we don't know the other input of either one. We have a problem. To resolve it we'll pick values for  $q$  and  $\bar{q}$  and check whether they stay the same when they are feed back into the inputs of the NOR gates.

We'll need to remember the truth table for a NOR gate. It's

<sup>1</sup> Recall that we refer to the inputs and outputs of logic gates as signals. These signals can have values of 0 or 1.

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

TABLE 1: TRUTH TABLE FOR NOR GATE

Let's set both  $q$  and  $\bar{q}$  to 0. Then both inputs to both NOR gates would be 0. Referring to the NOR gate truth table, the outputs are both 1, so  $q$  and  $\bar{q}$  are both 1, which is not 0. So, that doesn't work. Let's try  $q = 1$  and  $\bar{q} = 0$ . The top NOR gate would then have inputs of  $\text{reset} = 0$  and  $\bar{q} = 0$ , so its output,  $q$ , is 1. That's consistent with what we picked. For the bottom NOR gate, the inputs become  $\text{set} = 0$  and  $q = 1$ . So, its output is 0 and that too is consistent with what we picked for  $\bar{q}$ . We've found one possible combination of inputs and outputs, which is

reset	set	q	$\bar{q}$
0	0	1	0

If we change  $\text{reset}$  to 1, then  $q$  changes to 0 and  $\bar{q}$  changes to 1. Now, if we change  $\text{reset}$  back to 0,  $q$  stays at 0 because the inputs to the top NOR gate would be 0 and 1. Since neither  $\text{set}$  nor  $q$  change, then  $\bar{q}$  remains at 1. Now, we have the following possible combination of inputs and outputs:

reset	set	q	$\bar{q}$
0	0	0	1

Notice that the inputs are the same as in the previous table, but the outputs are different. The outputs depend not just on what the inputs are, but by what the outputs had been. In this case, the state of the circuit is specified by the value of its outputs.

## Quiz Game Response Light Circuit

The Quiz Game Response Light Circuit is a sequential logic circuit. Its ability to "remember" past inputs is essential to its functionality. If Button A is pressed first then light A goes on. If it is hit second, light A does not go on. The output depends on the sequence of inputs.

Part of the circuit is a D-Type flip-flop, which is in the 4013 IC. The flip-flop, as used in the circuit, allows the output to be maintained (that is, keeps on the light of the first contestant to press a button) until the circuit is reset. Additional feedback of the flip-flop outputs to the inputs of AND gates earlier in the circuit allows the action of one contestant pressing a button first to lock out the light of the other contestant of being lit.

We will learn more about how this circuit works later.