Department of Computer Engineering
Digital Hardware Systems
CpEA 2202 – Electronic Circuits

Laboratory Report

Laboratory Exercise No.:	10	**Date Performed:**	March 28, 2025
Laboratory Exercise Title:	Transistor-Transistor Logic (TTL)		
Name of Student:	Pace, Vaun Michael C. Parras, Paul Andrew F,	**Document Version:**	2

Part I

Transistor-Transistor Logic (TTL) is a type of digital circuit that uses bipolar junction transistors (BJTs) to handle both logic functions and amplification. TTL, which was invented in the 1960s, became a cornerstone technology in digital electronics due to its resilience, speed, and simplicity of integration. A typical TTL logic gate, such as a NAND gate, contains a multi-emitter input transistor that accepts several logic inputs. This is followed by a phase-splitter transistor, which conditions the signal, and a totem-pole output stage, which employs two transistors and a diode to provide fast switching speeds and adequate output drive. This topology allows TTL circuits to efficiently source and sink current, making them capable of directly driving other logic inputs or devices.

TTL has various advantages, including quick switching speeds and high noise immunity. It can withstand minor voltage fluctuations without generating logic mistakes, making it dependable in noisy conditions. Its fan-out capability—defined as the number of gate inputs that a single output can drive—is particularly noteworthy, enabling the construction of large digital systems with fewer interconnection constraints. Despite its higher power consumption than CMOS logic, TTL remains useful in timing-sensitive applications.

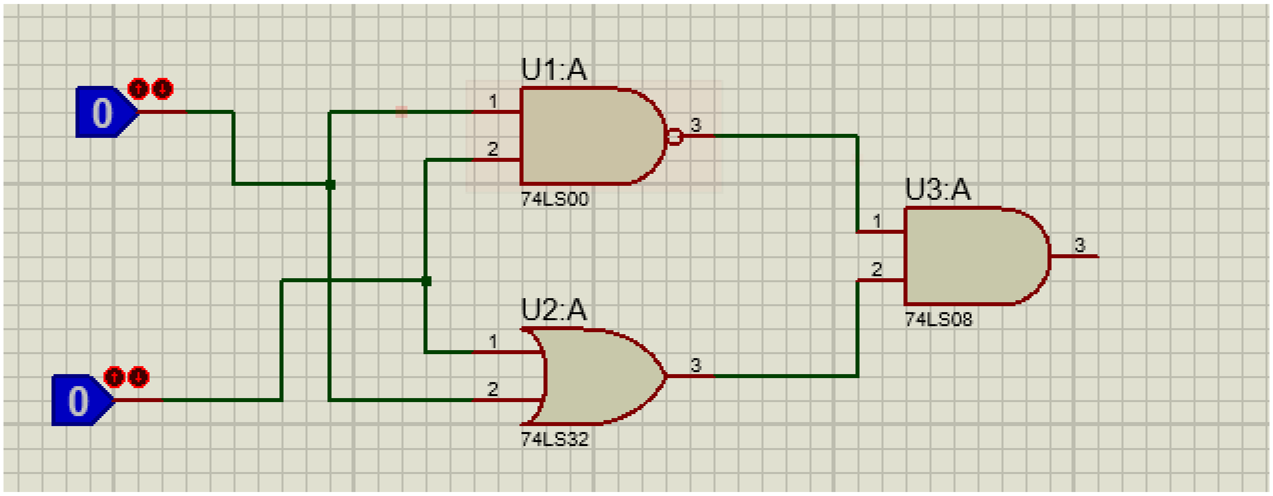
Over time, numerous TTL families have arisen to improve particular performance characteristics. For example, Schottky TTL (S-TTL) employs Schottky diodes to reduce transistor saturation, allowing for faster switching. Low-Power TTL (L-TTL) uses less energy but has a little slower speed, whereas High-Speed TTL (H-TTL) provides optimal speed for time-critical applications. These versions enable TTL to meet a variety of needs, including efficiency, performance, and circuit complexity.

Part IIa:

Table 2.1: Data for the Logic Gates

AND Gate				OR Gate			
A	B	Q	Vq (V)	A	B	Q	Vq (V)
not connected	not connected	0	0	not connected	not connected	0	0.0325
0	0	0	0.0001	0	0	0	0.0396
0	1	0	0.0374	0	1	1	5.03
1	0	0	0	1	0	1	5.03
1	1	1	2.526	1	1	1	5.03
XOR Gate				NAND Gate			
A	B	Q	Vq (V)	A	B	Q	Vq (V)
not connected	not connected	0	0.5276	not connected	not connected	1	5.036
0	0	0	0.5321	0	0	1	5.036
0	1	1	5.035	0	1	1	5.035
1	0	1	5.036	1	0	1	5.036
1	1	0	0.7	1	1	0	0.0232
NOT Gate							
A		Q	Vq (V)				
not connected		1	5.036				
0		1	5.036				
1		0	0.0056				

Part IIb:



A	B	Q	Vq (V)
not connected	not connected	0	0
0	0	0	0
1	1	1	4.02
0	0	1	3.89
1	0	0	1.37

Part IIc:

A	B	Q	Vq (V)
not connected	not connected	0	0.1355
0	0	0	0.130
1	1	1	2.404
0	0	1	2.11
1	0	0	0.1195

Part III. Observations

1. **Observe the data in Table 1. What can you conclude on the voltage output (VQ) and the logic output?**
    - Based on the data in Table 1, we may conclude that the voltage output (VQ) influences the circuit's logic output. When the voltage output is much larger than 0V (usually around 5V in TTL logic), the logic output is interpreted as 1 (HIGH). When the voltage approaches 0V, the logic output is regarded as 0 (low). As a result, determining the logic state produced by the gate is heavily dependent on the voltage level at its output.

2. **Notice that there are a combination of "not connected" inputs in Table 1. Observe the value of VQ. What can you infer on the output of a TTL logic gate when the inputs are not connected to either VCC or ground?**
    - Leaving TTL (Transistor-Transistor Logic) inputs unconnected, or "floating," might yield unpredictable and inaccurate results. This is because floating inputs can pick up electrical noise, resulting in erratic behavior in the circuit. To provide consistent and specified logic levels, employ pull-up or pull-down resistors, which offer a default high or low voltage level when there is no active signal.

3. **Analyze Figure 4 carefully. If an AND gate has a boolean function of Q = A.B, what is the logic function of the TTL circuit in Figure 4? Express answer as boolean function**
    - Q = A B
    - First gate input: AND Gate = (A * B) Invert into NAND Gate = ¬(A * B)
    - Second Gate Input: OR Gate = A + B
    - Final Gate combining the outputs of the two input using AND Gate Q = (NAND Output) * (OR Output) = ¬(A * B) * (A + B)

4. **From the data in Table 2, what is the circuit in Figure 4 in terms of its function?**
    - Based on the data in Table 2, the circuit in Figure 4 acts as an XOR (exclusive OR) gate. This is obvious from its truth table, which corresponds to the typical behavior of an XOR gate—producing a HIGH output only when one of the two inputs is HIGH. The circuit limits the output to a single HIGH state, confirming its XOR functioning while also potentially preventing signal conflicts or overloading in bigger logic systems.

5. **From the data in Part Iic, what is logic-0 and logic-1 in terms of voltage?**
    - In TTL (Transistor-Transistor Logic) circuits, a Logic 0 (LOW) voltage ranges from 0V to 0.8V, whereas a Logic 1 (HIGH) voltage ranges from 2V to 5V. Voltages between 0.8V and 2V are deemed unknown, which can result in unpredictable behavior.

References

[1] Electrical4U, "Transistor-Transistor Logic (TTL)," *Electrical4U*, [Online]. Available: https://www.electrical4u.com/transistor-transistor-logic-or-ttl/. [Accessed: May 2, 2025].

[2] TechTarget, "Transistor-transistor logic (TTL)," *TechTarget*, [Online]. Available: https://www.techtarget.com/whatis/definition/transistor-to-transistor-logic-TTL. [Accessed: May 2, 2025].