Project 1 — Gate-Level Implementation of toUpper()

Author: Gaurav Banepali
Course: CSC 211000 – Digital Design
Date: November 2025

1. Purpose

The purpose of this project is to design and implement a gate-level ASCII toUpper() converter using only primitive Verilog gates with fixed propagation delays. The converter transforms lowercase ASCII letters ('a'–'z') into their uppercase equivalents ('A'–'Z') by clearing bit 5. All non-lowercase ASCII values are passed through unchanged. Timing analysis is performed to evaluate the maximum operating speed of the circuit.

2. Design

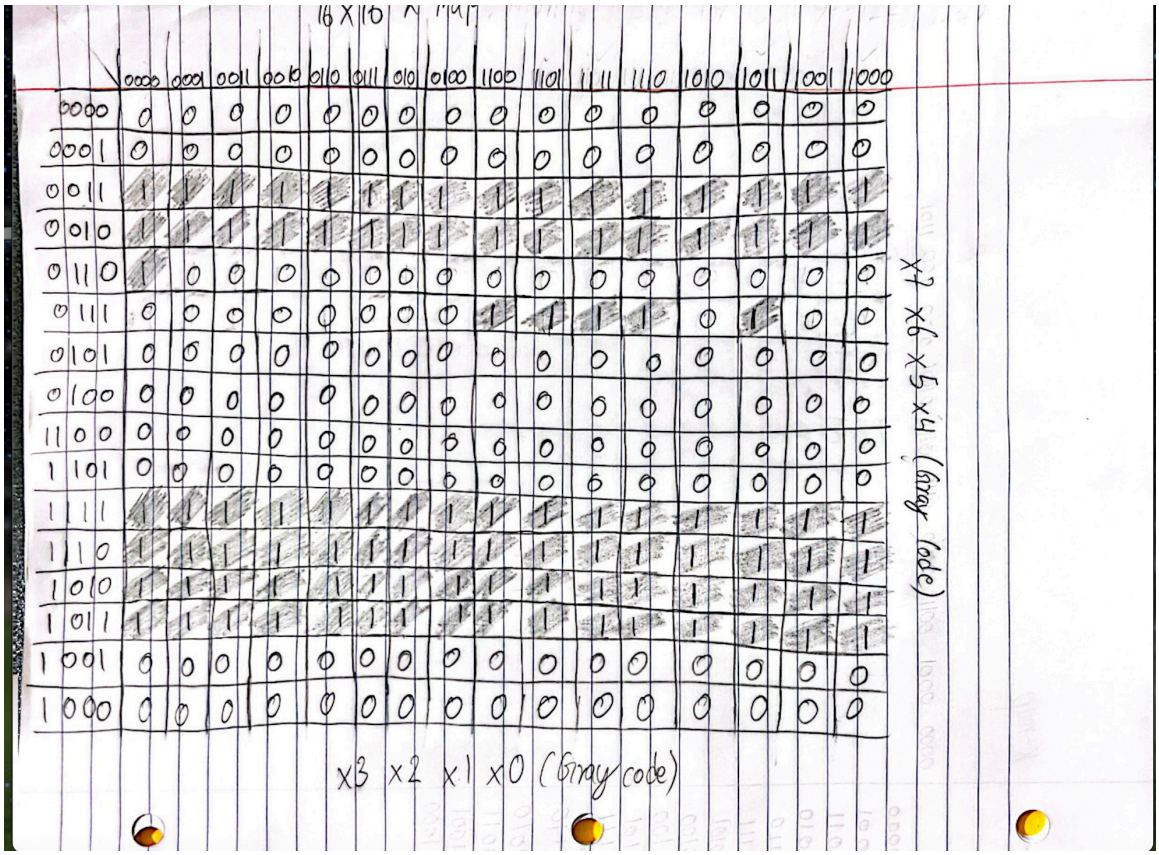
The circuit was constructed using only primitive Verilog gates: NOT, AND, OR, NAND, NOR, XOR, XNOR, and BUF. Each gate type uses a fixed delay to model real hardware behavior. Only one bit (x5) is modified; the remaining bits are passed through buffers to satisfy the primitive-only rule.

Gate Type	Delay (#)	Description
NOT	#5	Inverts a single bit
AND, OR	#10	Basic combinational gates
NAND, NOR	#12	Combined logic variations
XOR, XNOR	#15	Used for comparison and parity
BUF	#4	Buffers or stabilizes output

3. K-Map Analysis

A full 16×16 K-map was created using Gray-code ordering along both axes. Bit 5 must be forced to 0 for ASCII values 97–122 ('a'–'z'). The resulting map shows the active region where y5 transitions from 1 to 0.

Figure 1 — Hand-Drawn K-Map for y5:


16 x 16
x7 x6 x5 x4 (Gray code)
x3 x2 x1 x0 (Gray code)

4. Simulation Results

The circuit was simulated using Icarus Verilog (iverilog + vvp) with waveform inspection performed in GTKWave. Below are the waveform outputs at several input-change intervals.

Figure 2 — Output at 40 ns Interval (Correct Behavior)

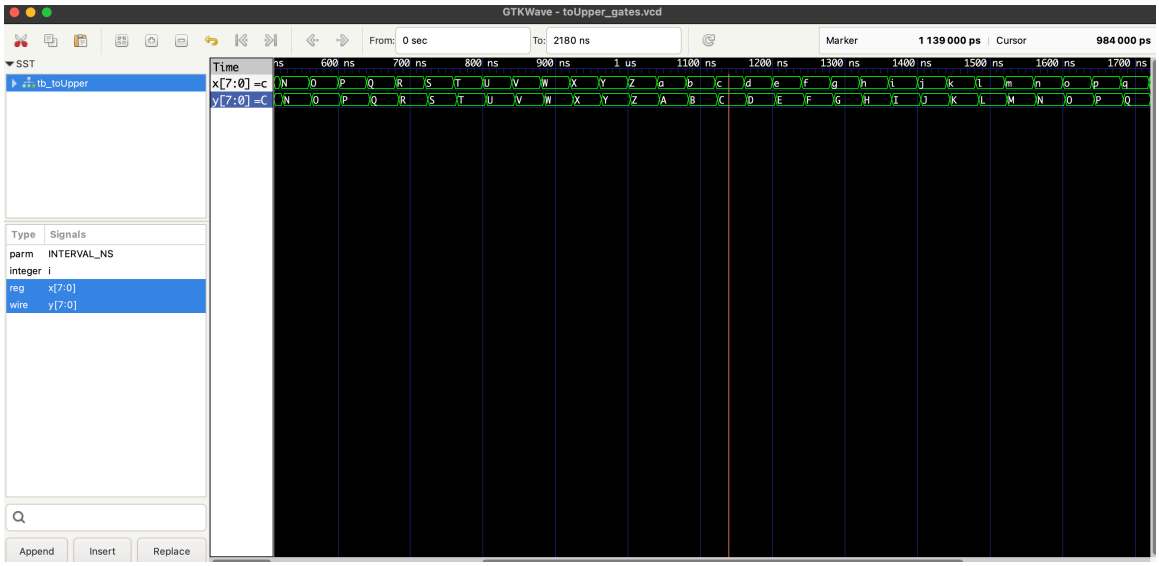


Figure 3 — Output at 10 ns Interval (Minimum Passing)

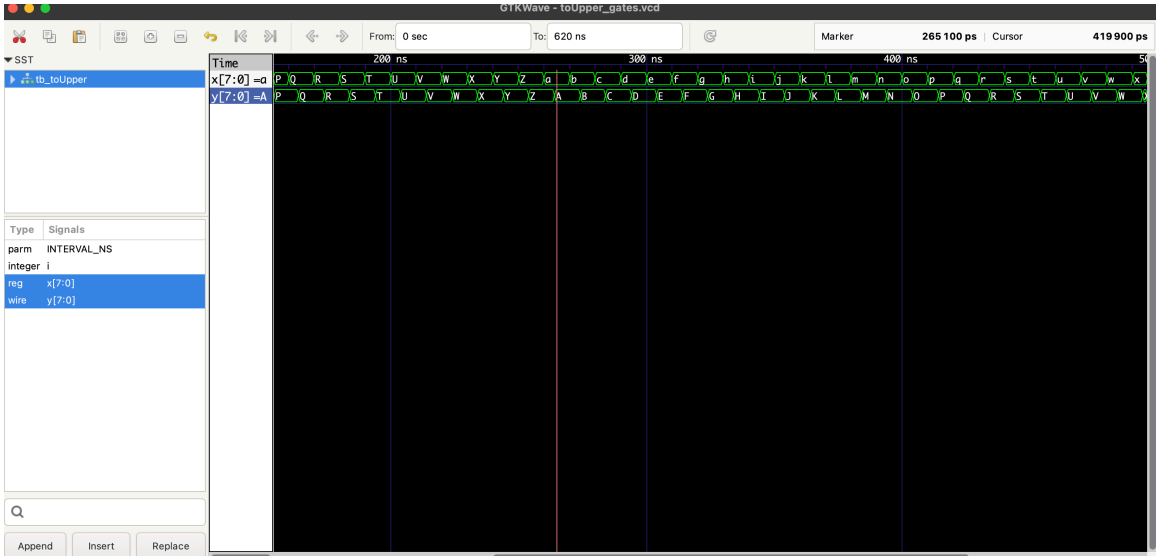
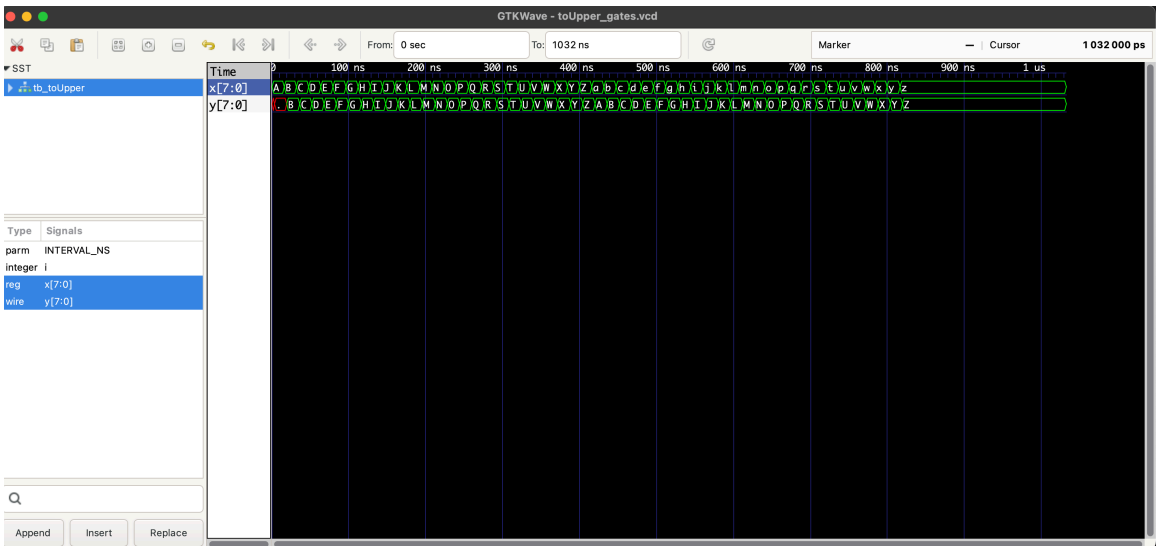


Figure 4 — Output at 8 ns Interval (Failing)



5. Stress Testing and Timing Analysis

To determine the timing limits of the gate-level implementation, the delay between input changes (INTERVAL_NS) was gradually reduced. At 40 ns, 20 ns, and 12 ns the output remains correct. At 10 ns the circuit still works correctly, making it the minimum safe operating interval. At 8 ns, unstable and incorrect outputs appear due to overlapping gate delays.

Interval (ns)	Behavior
40	Correct behavior
20	Correct behavior
12	Correct behavior
10	Minimum passing interval
8	Failing interval (unstable output)

6. Conclusion

The gate-level toUpper() module successfully implements ASCII case conversion using minimized logic. The K-map allowed y5 to be simplified into an efficient SOP form. Simulation and stress testing confirmed that the circuit is functionally correct and meets timing requirements down to 10 ns. Below this value, gate delays accumulate and cause

logic hazards, leading to incorrect outputs. Overall, the design satisfies the project requirements and provides clear insight into timing-driven hardware behavior.