ECE 310
Lab 7 Report: BCD Adder

Ohm Patel

November 9, 2025

Abstract

This report presents the design, implementation, and verification of a single-digit Binary Coded Decimal (BCD) adder with carry-in and carry-out functionality. The design was implemented in Verilog and tested both independently and as part of a provided four-digit structural BCD adder in Vivado. The adder performs a standard binary addition, applies correction for invalid BCD sums, and generates a carry-out when necessary. Simulation waveforms for both single- and four-digit configurations confirm correct operation, and synthesis results validate the expected combinational architecture.

1 Introduction and Background

Binary Coded Decimal (BCD) arithmetic represents each decimal digit (0–9) using a 4-bit binary code. Direct binary addition of BCD digits can produce invalid results greater than 9, requiring correction to maintain proper BCD format. This correction is achieved by adding 6 (0110_2) whenever the binary sum exceeds 9 or generates a carry-out.

The goal of this lab was to design a single-digit BCD adder capable of performing:

1. Binary addition of two 4-bit BCD digits and a carry-in.
2. Detection of invalid BCD results (> 9).
3. Addition of 6 to correct invalid codes and generate a proper carry-out.

The single-digit module was verified independently and also integrated into a provided 4-digit ripple structure to confirm carry propagation across multiple digits.

2 Design and Block Diagrams

The single-digit BCD adder performs a 4-bit binary addition between inputs A, B, and C_{in}. If the intermediate 5-bit sum exceeds 9, the circuit adds 6 (0110_2) to produce a valid BCD result. The fifth bit of the corrected sum becomes the carry-out (C_{out}). This design is fully combinational, requiring no clock or sequential logic.

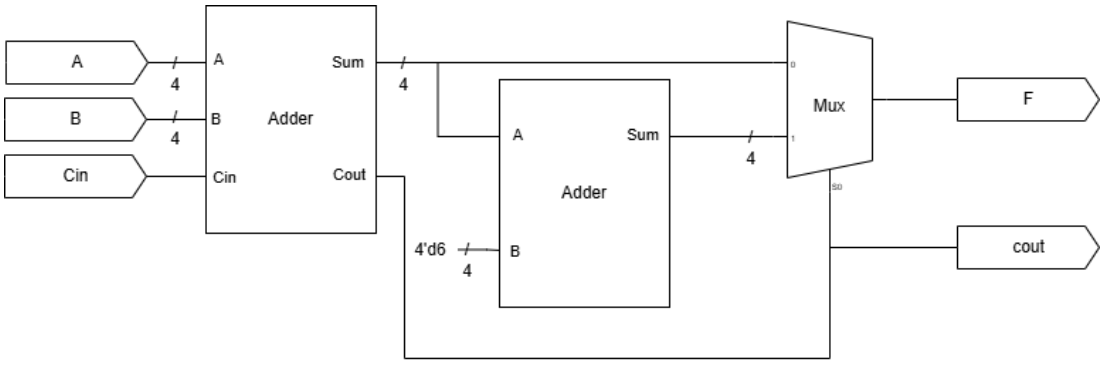

Figure 1: Block diagram of the single-digit BCD adder showing addition and correction logic.

For the 4-digit BCD adder, four single-digit modules are connected in series. The carry-out from each stage serves as the carry-in to the next digit, forming a ripple-carry configuration. This approach mirrors traditional binary adders while preserving valid BCD operation throughout.

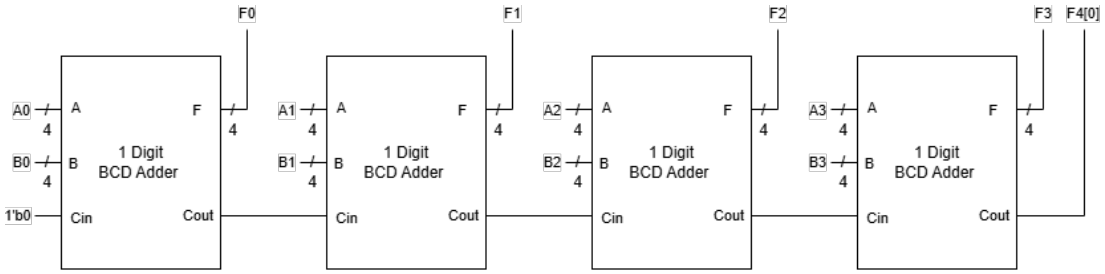

Figure 2: Block diagram of the four-digit BCD adder showing carry propagation across digits.

3 Verilog Implementation

Listing 1: Verilog implementation of the single-digit BCD adder

```
module combBCDadd_digit (
        input [3:0] A, B,
        input cin,
        output cout,
        output [3:0] F );

        // Your code goes here
        wire [4:0] sum;
        assign sum = A + B + cin;

        wire correction;
        assign correction = (sum > 9);

        wire [4:0] bcd_corrected;
        assign bcd_corrected = sum + (correction ? 5'b00110 : 5'b00000);

        assign F = bcd_corrected[3:0];
        assign cout = bcd_corrected[4];

endmodule
```

4 Testing and Verification

Testing was conducted using two provided testbenches:

- `combBCDadd_digit_tb.v` for the single-digit module.
- `combBCDadd_4d_tb.v` for the four-digit structural implementation.

The single-digit testbench verified correction logic and carry generation under multiple input combinations, including edge cases such as $A = 9$, $B = 8$, and $C_{in} = 1$. The four-digit testbench validated correct propagation of carries and consistent output representation across all four digits.

All simulations were run in Vivado with a 10 ns timescale. Since the modules are purely combinational, no clock was required.

5 Simulation Results

5.1 Single-Digit BCD Adder

When adding $A = 9$, $B = 8$, and $C_{in} = 0$, the sum of 17 triggered the correction logic, adding 6 to yield a valid BCD output of 7 with a carry-out of 1. The simulation waveform confirmed proper detection, correction, and output behavior.

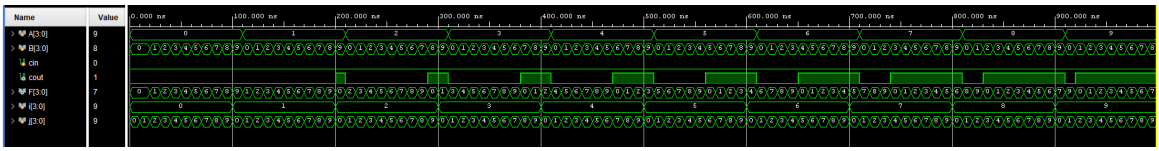

Figure 3: Simulation waveform of the single-digit BCD adder showing sum correction and carry-out generation.

5.2 Four-Digit BCD Adder

The 4-digit BCD adder was tested using two non-trivial inputs, verifying proper carry propagation and multi-digit summation. An example case, such as $A = 1234$ and $B = 5634$, produced a correct 5-digit BCD result of 6868, demonstrating that each stage's carry-out correctly influenced the next stage's input.

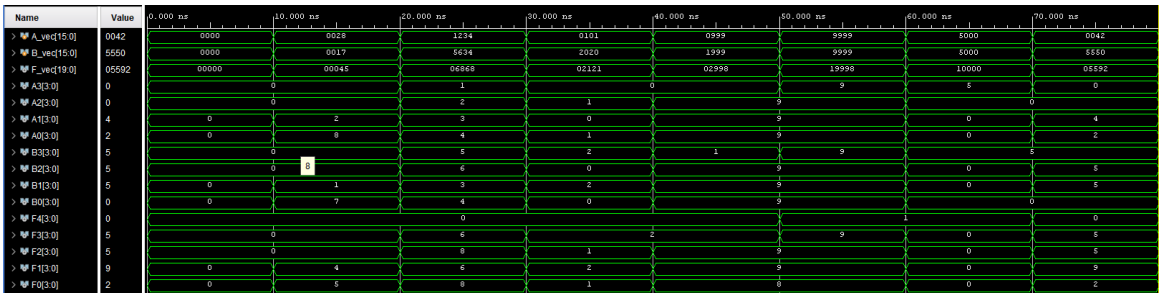

Figure 4: Four-digit BCD adder simulation showing accurate carry propagation and valid BCD output.

6 Synthesis Results and Discussion

Synthesis in Vivado confirmed a purely combinational architecture for both modules. The single-digit design consisted of a small set of adders and logic gates implementing the correction condition. The 4-digit structural version exhibited four interconnected instances of the single-digit block, reflecting the expected ripple-carry configuration.

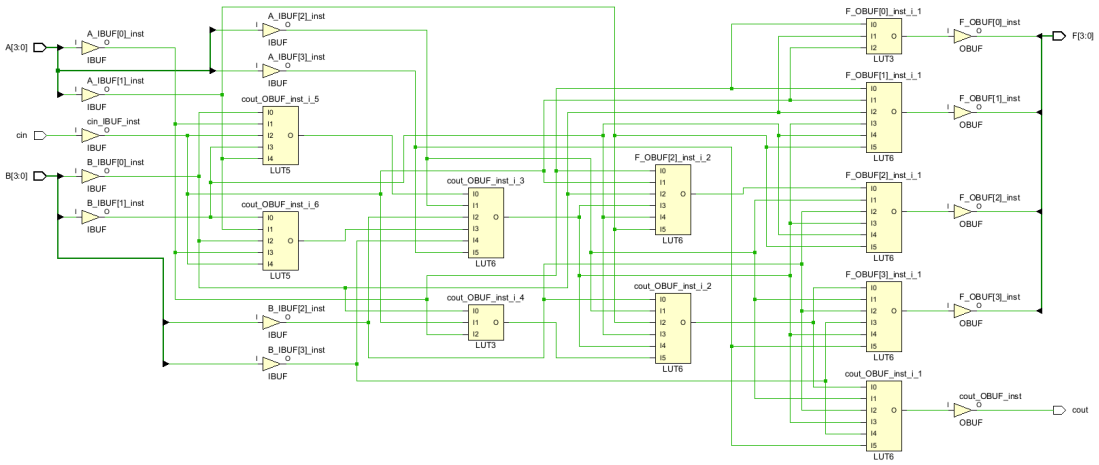

Figure 5: Synthesized RTL schematic of the single-digit BCD adder.

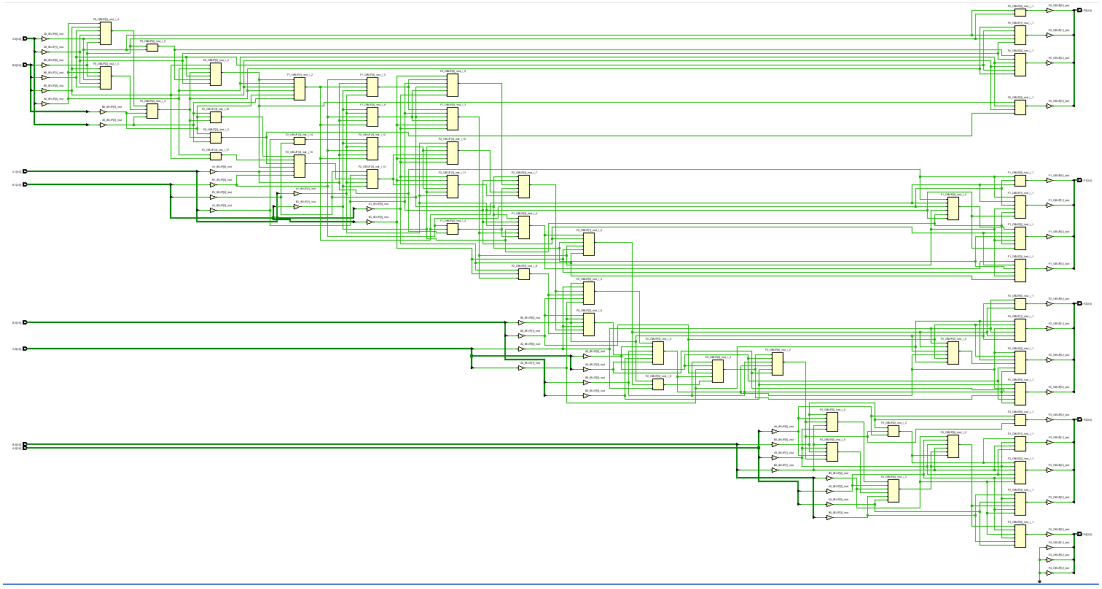

Figure 6: Synthesized RTL schematic of the four-digit BCD adder.

The synthesized hardware matched the theoretical design from the pen-and-paper diagram, showing identical behavior to the behavioral simulation. No timing issues or unexpected resource usage were observed. Both implementations met the lab's functional and structural expectations.

7 Conclusion

A single-digit BCD adder was successfully designed, implemented, and verified in Verilog, then tested within a four-digit structural configuration. Simulation confirmed correct arithmetic, carry generation, and correction logic across all tested cases. Synthesis validated a clean, combinational architecture consistent with the theoretical design. The work completed fulfills all lab deliverables and demonstrates a solid understanding of combinational arithmetic logic and modular Verilog design.

This report was typeset in LaTeX.