ECE 571:VLSI system design
Lab 4 -report (2024 spring)

Name: Mukesh Sekar
ID : 836763523

Name: Alec Maxwell
ID : 835704136

Name: Teague Mcphearson
ID : 835531187

GATENAME_SIZE

8-bit asynchronous ripple-carry adder/subtractor. The adder/subtractor has a mode control, a carry-in and a carry-out for cascading them to do 16-bit addition and subtraction in an asynchronous fashion.

Introduction

Imagine a digital circuit that can add or subtract 8-bit binary numbers. This is the core function of an 8-bit asynchronous ripple-carry adder/subtractor. Unlike some other adders, it operates asynchronously, meaning each bit position calculates its result independently without waiting for previous bits. This approach prioritizes simplicity over speed, making it suitable for cost-sensitive applications where immediate results are not crucial.

A key element within this adder/subtractor is the full adder. This building block takes three binary inputs: two numbers and a carry-in from a previous bit position. Its internal logic combines these inputs, generating a sum bit and a carry-out bit. These full adders are then connected in a chain-like structure, where the carry-out from one adder becomes the carry-in for the next.

For subtraction, the second number undergoes a bit-wise inversion before entering the chain. This effectively converts subtraction into an addition operation with the inverted number's two's complement. The carry-out bits ripple through the chain, meaning each adder waits for the carry from the previous one before calculating its own sum. This rippling effect introduces a slight delay compared to faster adders.

To handle 16-bit operations, two 8-bit adder/subtractor units are cascaded. The lower 8 bits of each number connect to one unit, while the upper 8 bits go to the other. The mode control signal, indicating addition or subtraction, and the carry signals are appropriately linked between these units. This cascading structure allows the combined circuit to perform 16-bit addition or subtraction, with the final carry-out bit indicating potential overflow or underflow (when the result exceeds the representable range). While the asynchronous design introduces some delay due to the carry rippling, it offers several advantages. Ripple-carry adder/subtractors are simpler to design and implement compared to faster alternatives, making them more cost-effective. This simplicity makes them suitable for situations where processing speed is not a critical factor, and cost efficiency is a priority.

Simulation setup and Transistor level schematic

Step1:
System Specification: Begin by defining the requirements and specifications of the system to be simulated. This involves identifying the desired functionality, input-output relationships, and any constraints or limitations.

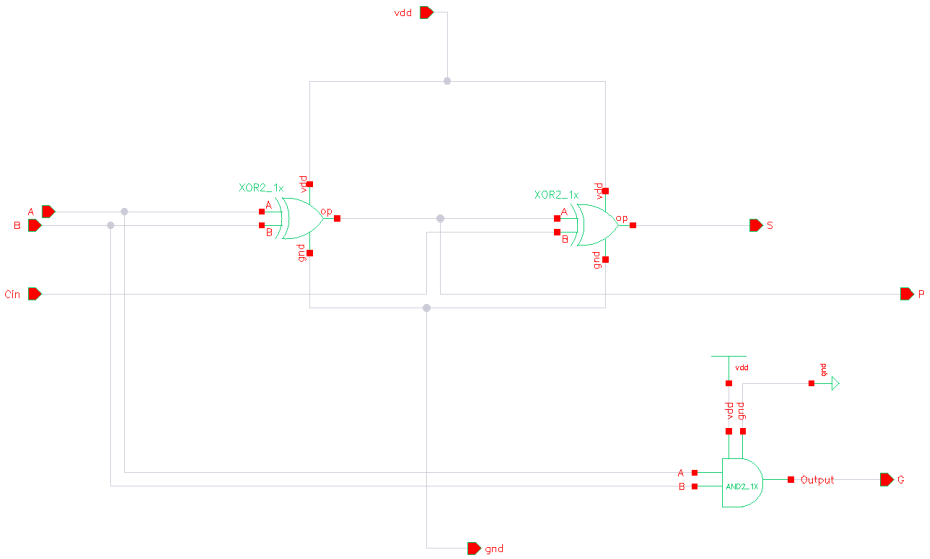
Step2:
Component Selection: Select the appropriate electronic components and devices needed to realize the desired functionality. Consider factors such as performance, compatibility, and availability when choosing components.

Step3:
Schematic Design: Create a schematic diagram illustrating the interconnections between the selected components. This schematic serves as a blueprint for the circuit layout and guides the simulation process.

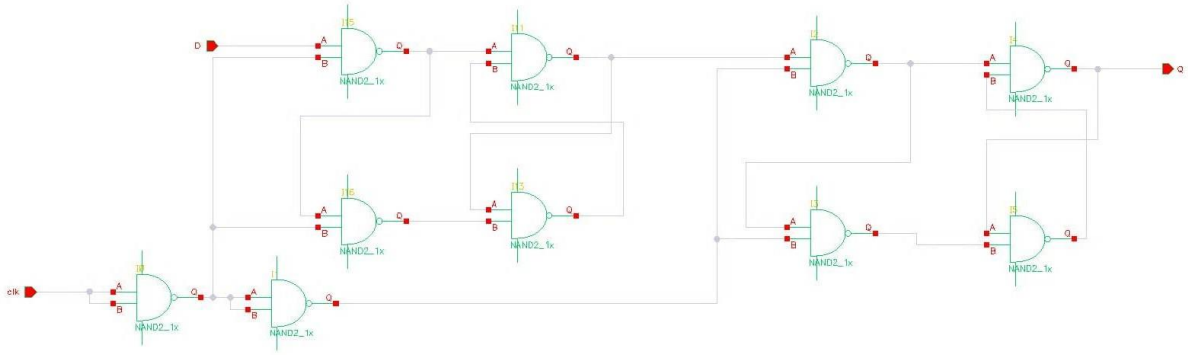
Step4:
Simulation Setup: Configure the simulation environment using a suitable software tool such as SPICE (Simulation Program with Integrated Circuit Emphasis). Define parameters such as simulation time, input signals, and component models.

Step5:
Simulation Execution and Analysis: Run the simulation to observe the behavior of the circuit over time. Analyze the generated waveforms and results to verify the circuit's performance against the specified requirements. Iterate and refine the design as needed based on the simulation outcomes.

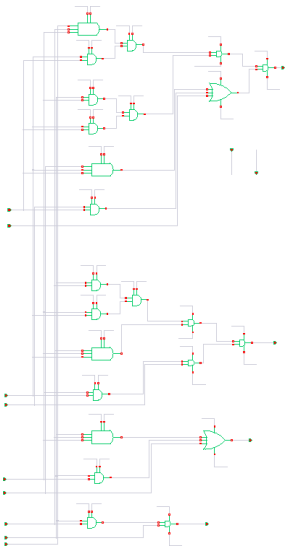
Full adder Schematic



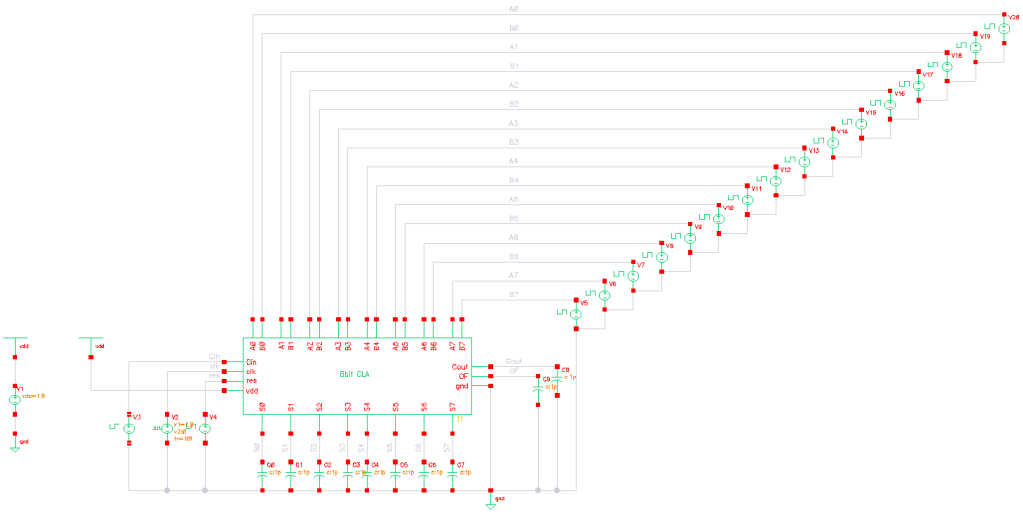
D-Flip Flop



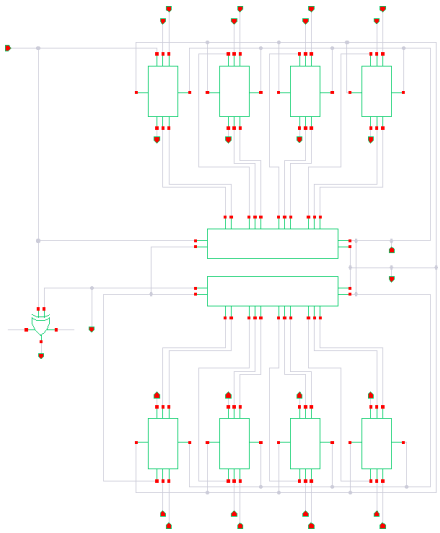
8-bit_CLA Schematic



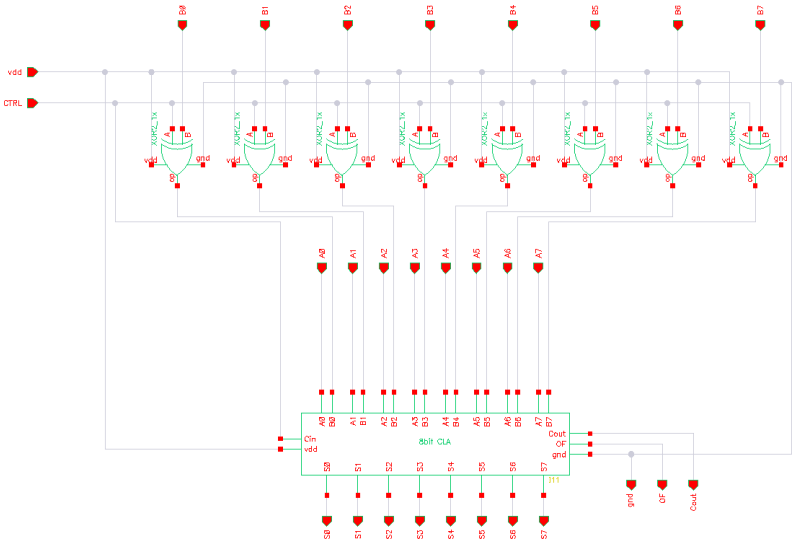
8-bit_CLA Test Bench



8-Bit Ripple carry adder Circuit

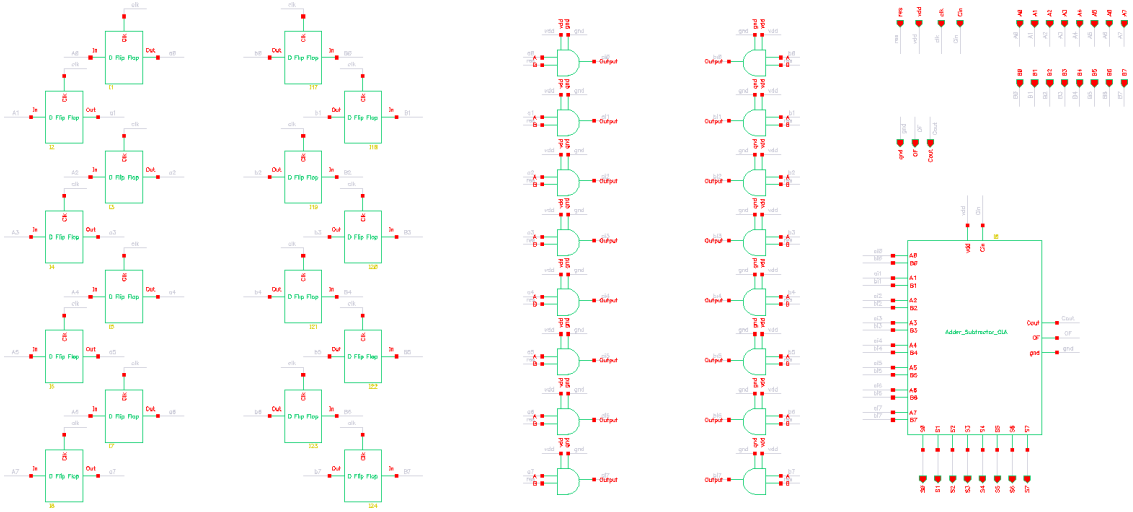


Adder and Subtractor Circuit

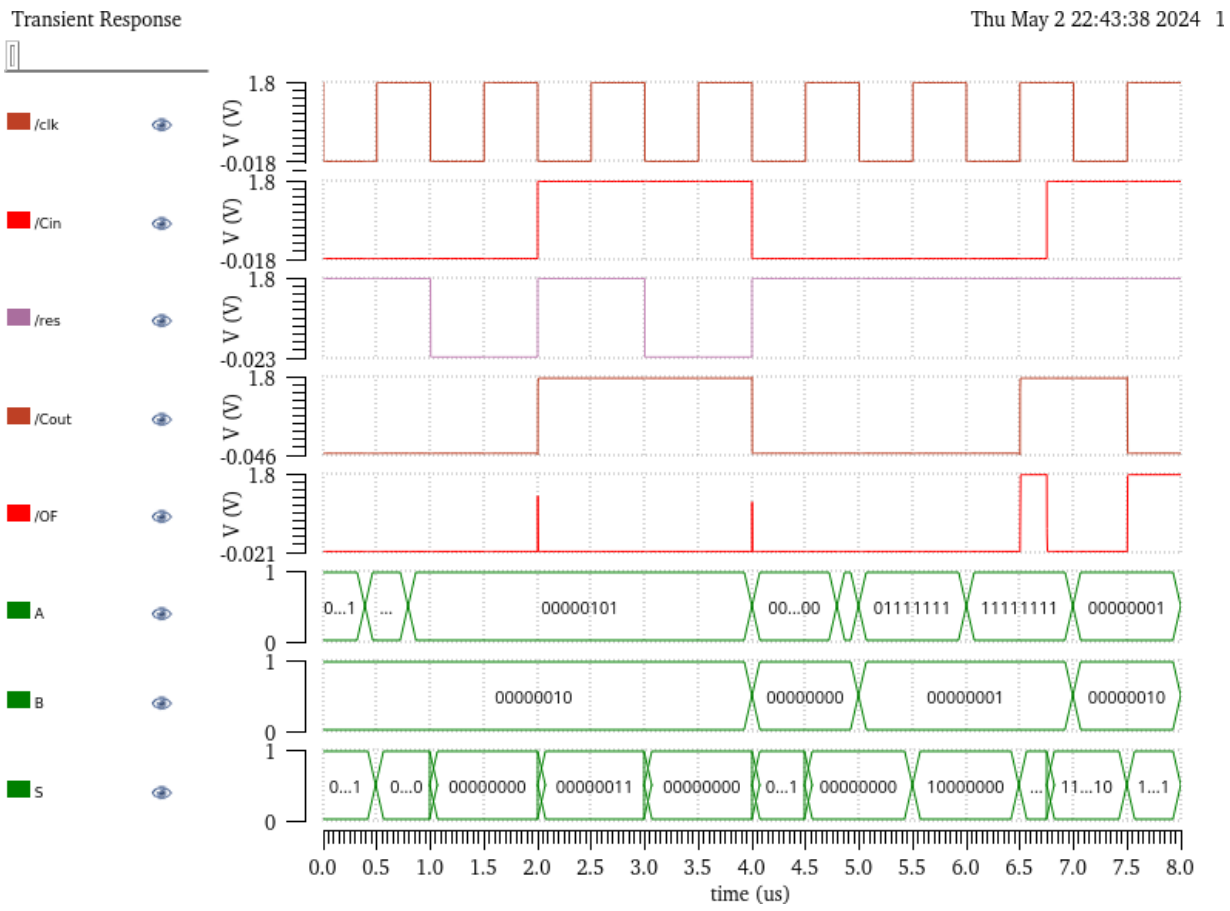


The 8-bit asynchronous ripple-carry adder/subtractor relies on interconnected full adders to perform binary addition, with each full adder handling three inputs to generate sum and carry-out bits. Subtraction functionality is achieved by introducing an XOR gate at the input to invert the bits of the second number, effectively obtaining its one's complement. When subtraction is required, this one's complemented second number is fed into the adder, along with the mode control signal, converting the operation into addition of the two's complement. This approach enables the circuit to perform both addition *and subtraction using the same hardware, enhancing its versatility and efficiency in processing binary numbers.*

8-Bit Ripple Carry Adder/Subtrator Circuit



Final Waveform Representing the Output of Adder / Subtractor

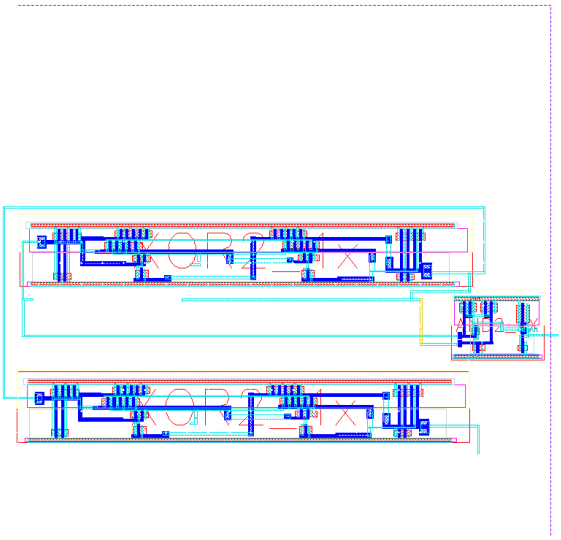


The waveform analysis provides insights into the behavior of an 8-bit ripple carry subtractor and adder through the examination of signal waveforms. In the top rows, signals such as 'rzk', 'rCin', 'rres', 'rCout', and 'rOF' exhibit rectangular waveforms, indicative of digital logic or control signals. These signals likely correspond to different stages or components within the subtractor and adder circuit, portraying operations like addition, subtraction, and overflow detection.

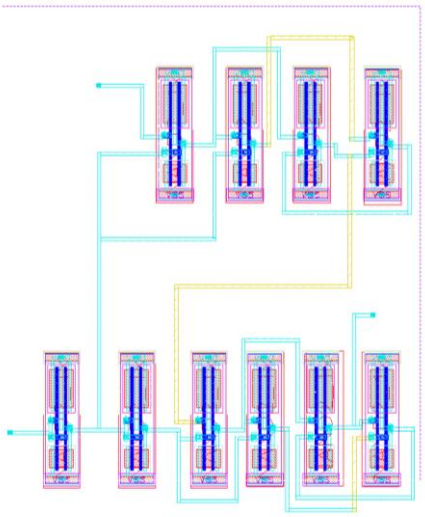
Meanwhile, the lower rows depict signals labeled 'A', 'B', and 's', showcasing binary waveforms with transitions between '0' and '1' values. These signals likely represent the input operands ('A' and 'B') and the output ('s') of the subtractor and adder. The transient behavior observed over a time span from 0 to 8 microseconds suggests dynamic changes in the circuit's state, possibly triggered by input stimuli or events.

Overall, the waveform analysis provides a glimpse into the operation of the 8-bit ripple carry subtractor and adder, highlighting the transient responses and digital data flow within the circuit. Further contextual information and detailed examination would be required to fully interpret the specific functionalities and behaviors observed in the waveforms.

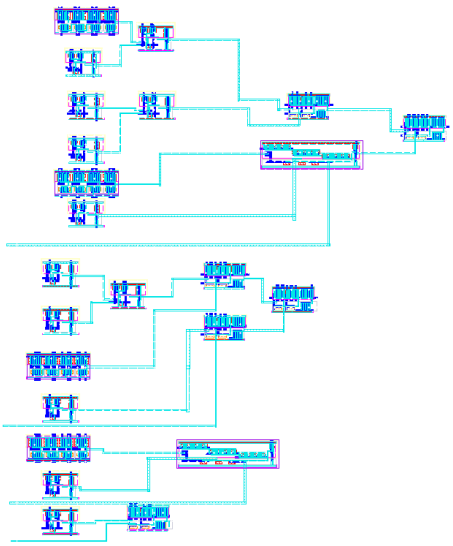
Layout Design



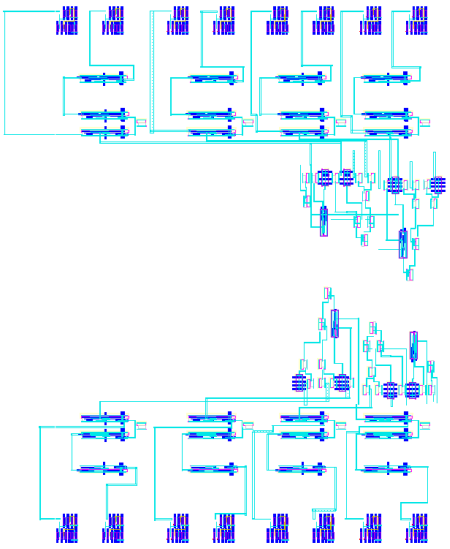
Full adder Layout



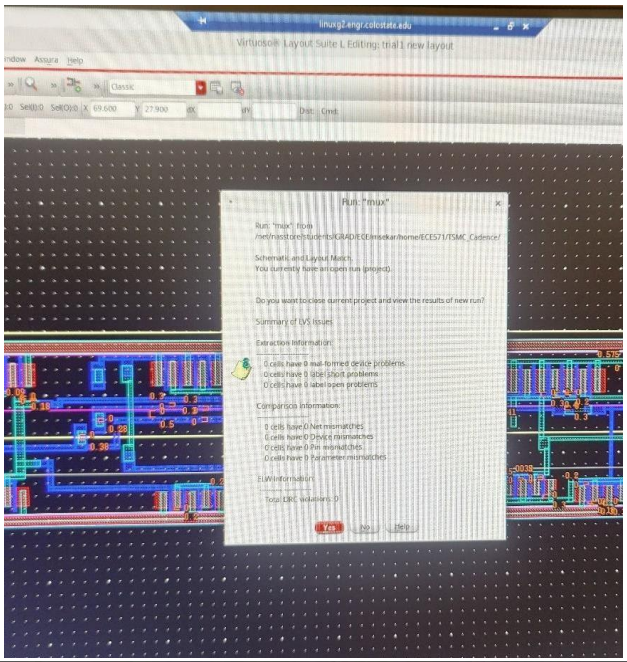
D-Flip Flop

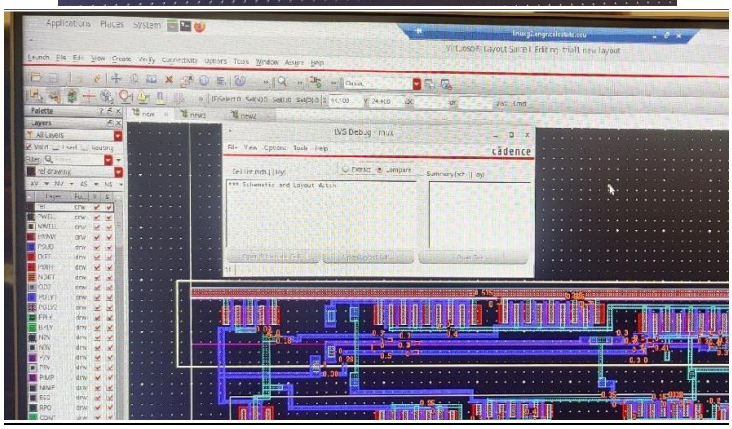


8-bit_CLA Layout



8-Bit Ripple Carry Adder/Subtrator Circuit Layout





In the pursuit of constructing a robust digital circuit, meticulous attention was devoted to each intricate component. Beginning with the foundational design of a Full Adder Layout, the circuitry expanded to include the essential functionality of a D Flip Flop Layout, serving as a stalwart guardian of clock pulses and overflow management. Seamlessly integrated within this framework were XOR gates, strategically positioned to augment the circuit's capabilities. As the design evolved, the synthesis of a 4-bit CLA Layout heralded a new level of sophistication, paving the way for the culmination of an 8-Bit Ripple Carry Adder/Subtractor Circuit Layout. Rigorous adherence to Design Rule Check (DRC) and Layout Versus Schematic (LVS) protocols ensured not only functional integrity but also manufacturing viability. This meticulous process yielded a finely tuned circuit, poised to deliver reliable performance across diverse computational tasks.

Conclusion

During the development phase of the 8-bit asynchronous ripple-carry adder/subtractor, integrating the XOR gate to enable subtraction initially faced challenges due to disparities in gate sizes and inaccuracies in node connections. Overcoming these hurdles necessitated meticulous adjustments and thorough refinement to ensure seamless functionality within the circuit. A significant complication arose from an oversight where not all inputs of the XOR gate were consistently linked to the power supply (VDD), resulting in inconsistencies in output behavior. This discrepancy manifested as an anomaly where an output of '1' indicated overflow, while '0' signified no overflow. To remedy this, D flip-flops were strategically introduced to regulate and stabilize the overflow detection mechanism. By rectifying gate sizes, ensuring proper node connections, and incorporating D flip-flops for overflow detection, the circuit achieved reliable performance, adeptly accommodating both addition and subtraction operations while effectively managing overflow conditions.

Furthermore, the layout initially encountered numerous errors during the Design Rule Check (DRC) and Layout Versus Schematic (LVS) stages. However, through diligent troubleshooting and meticulous attention to detail, these issues were successfully addressed, ultimately yielding a layout that passed the necessary validation checks and produced the desired output. This process underscored the importance of rigorous validation and iterative refinement in achieving a functional and reliable digital circuit design.