

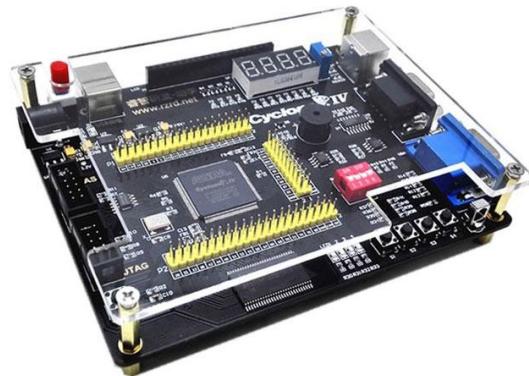
FPGA Implementation

Objective of FPGA Implementation

The objective of implementing the ALU design on an **FPGA** is to **validate the hardware functionality** of our Verilog-based ALU design through real-time operation and visualization. This **semi-custom implementation** demonstrates the transition from simulation to a working physical system and provides hands-on verification of design logic.

Target Platform and Tools

- **Target FPGA Board:** Intel **Cyclone IV EP4CE6E22**
 - **Design Tools Used:** Intel **Quartus Prime**
 - **Source Code:** Verilog HDL
- ▶ **Project files:** [Drive Link](#)



Implementation Details

- **Inputs:**
 - 4 LEDs on the FPGA are used as **selector inputs** to choose the desired ALU operation (in binary)
 - **Selectors** changes through the counter on Verilog code.
 - Two 4-bit operands **A** and **B** are assigned using input pins through Quartus (set in the Pin Planner).
- **Outputs:**
 - The ALU's output is displayed on the **7-segment display** of the FPGA.
 - The **operation mode** is selected based on the **most significant bit (MSB)** of the 4-bit selector (`SEL[3]`):
 - `SEL[3] = 0`: **Arithmetic operations** (e.g., addition, subtraction)
 - `SEL[3] = 1`: **Logical operations** (e.g., AND, OR, XOR)
 - **For arithmetic operations:**
 - 3 digits show the **magnitude** of the 4-bit result in **decimal**.
 - 1 digit indicates the **sign** (positive or negative) of the result.
 - **For logical operations:**
 - The result is displayed as a **4-bit binary** value using the **7-segment display**, with one digit per bit (or grouped as needed).

- **Functionality:**
 - The ALU performs operations such as addition, subtraction, AND, OR, etc., based on the selector input.
 - Real-time results are visible on the 7-segment display, enabling easy debugging and validation.

Verification and Testing

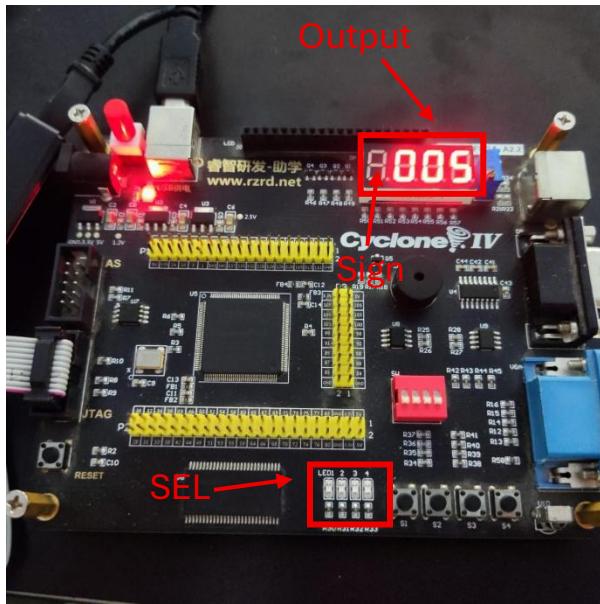
- The implemented design was verified through **physical testing on the FPGA board**.
- A demonstration video showing the ALU in operation is available here:

▶ **Demo Video:** [Drive Link](#)
- Additional screenshots and labeled images of selector configuration and output display will be included below.

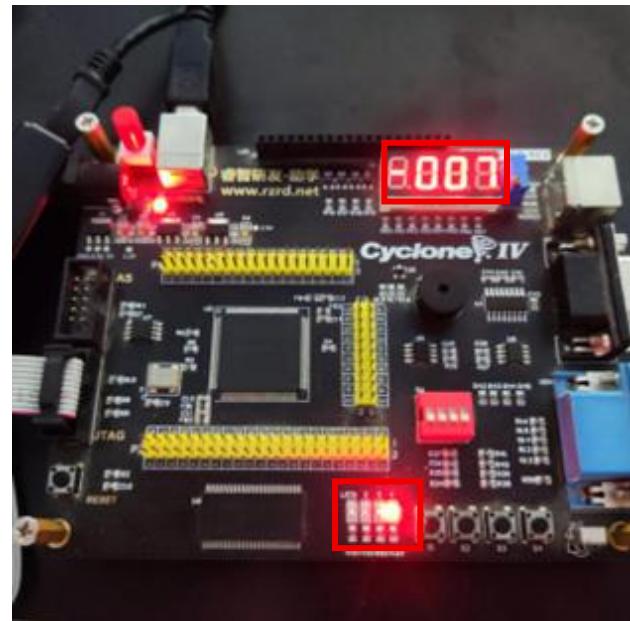
A=0100, B=1000

A=4, B=-8

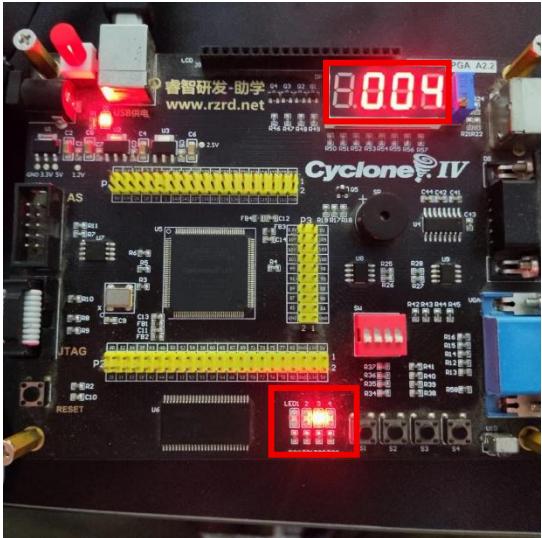
1. SEL=0000 (Increment A)



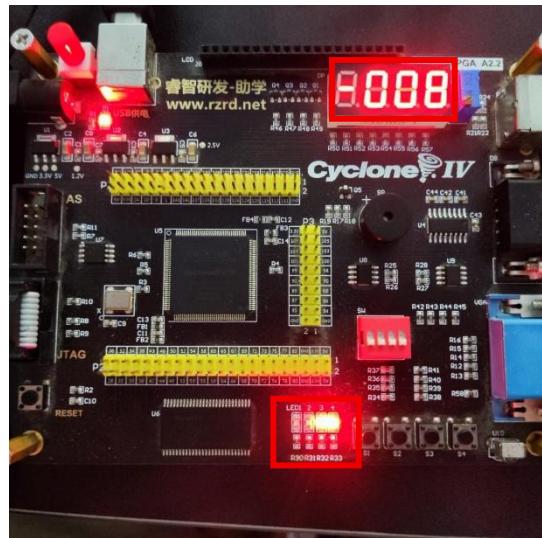
2. SEL=0001 (Increment B)



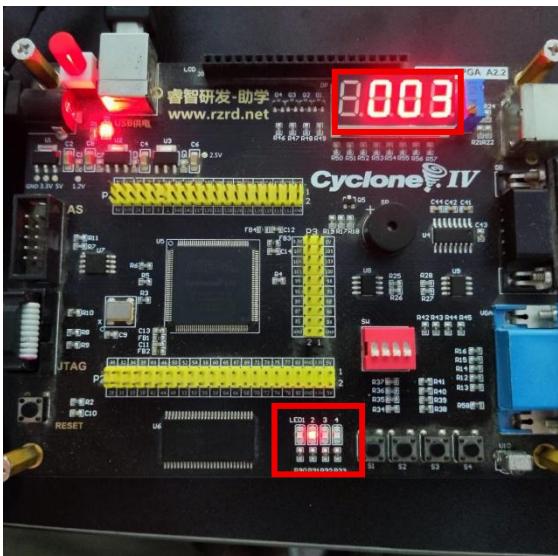
3. SEL=0010 (Transfer A)



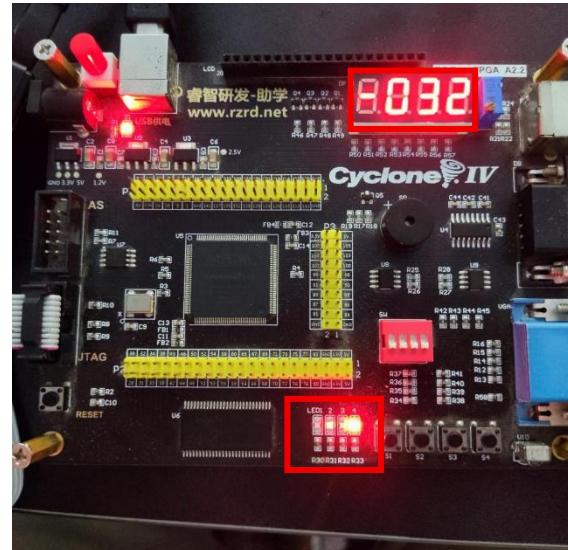
4. SEL=0011 (Transfer B)



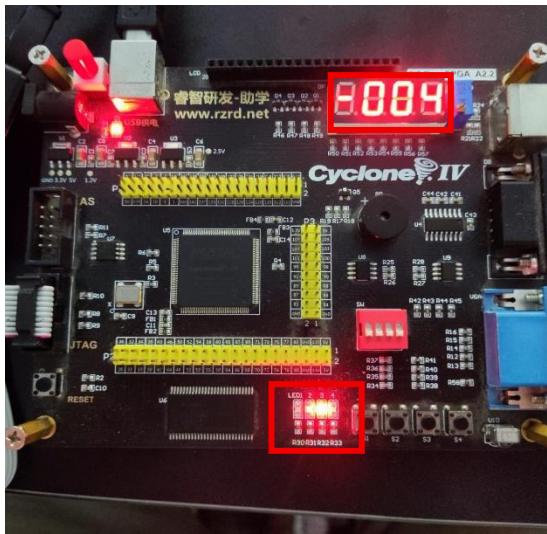
5. SEL =0100 (Decrement A)



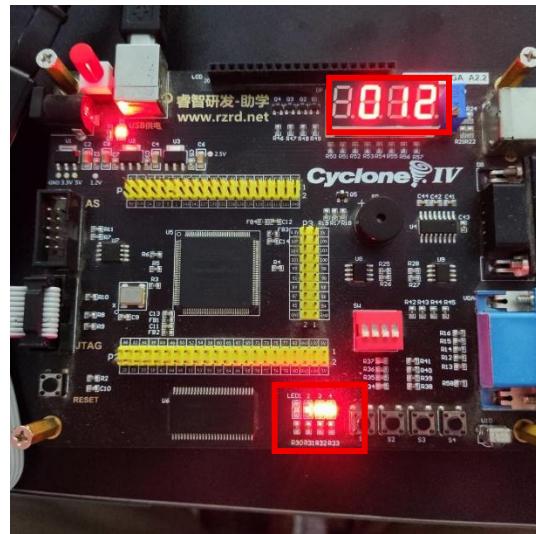
6. SEL= 0101 (Multiply A and B)



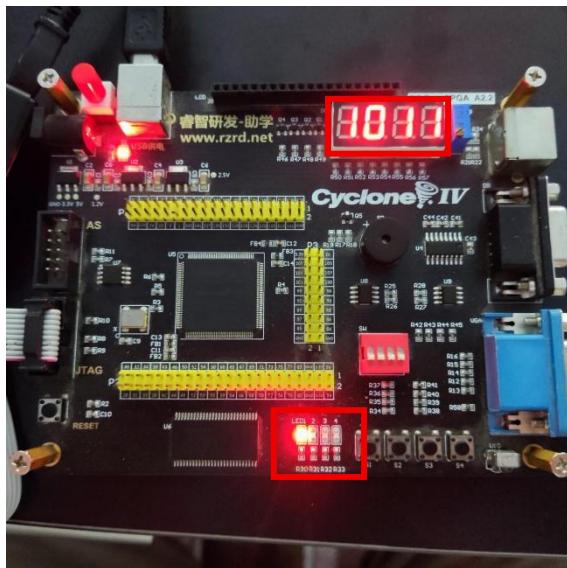
7. SEL=0110 (Add A and B)



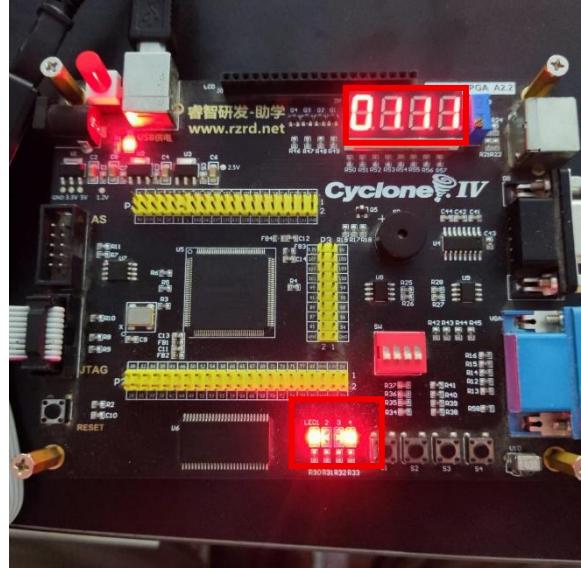
8. SEL=0111 (Subtract A and B)



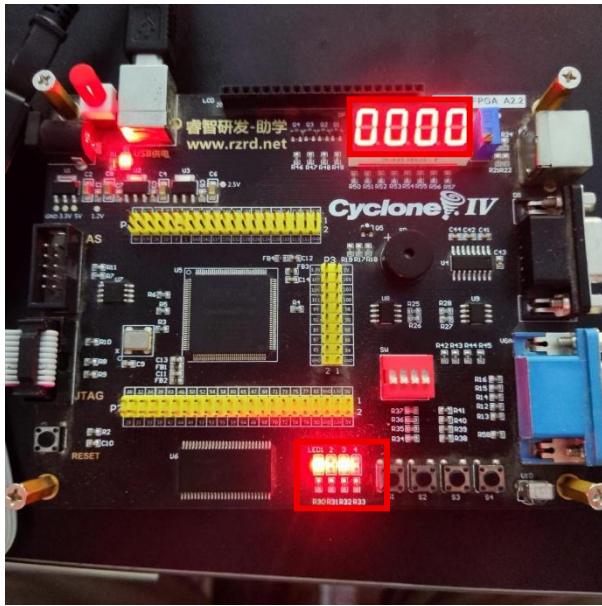
9. SEL=1000 (~A)



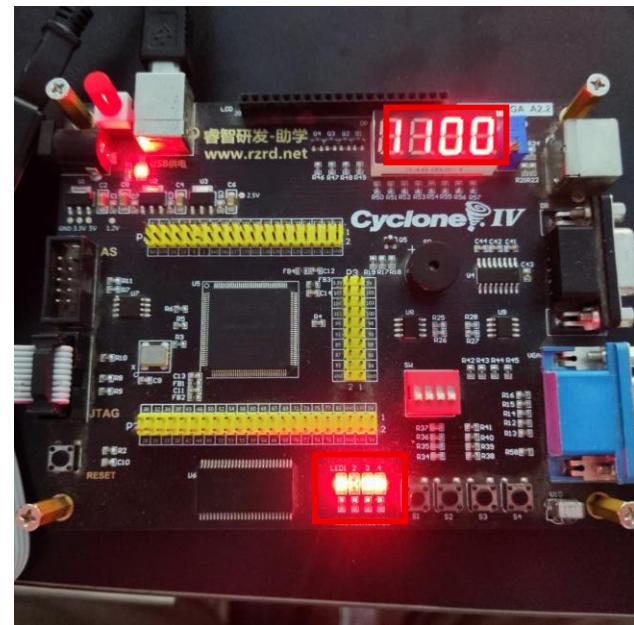
10. SEL=1001 (~B)



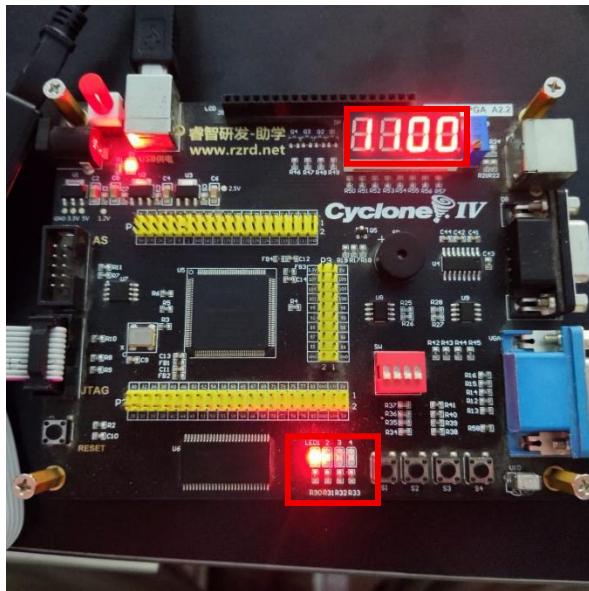
11. SEL=1010 (AND A & B)



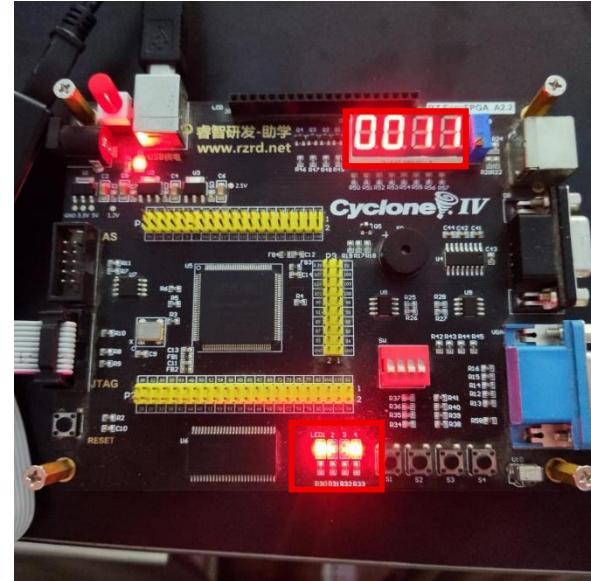
12. SEL=1011(OR A | B)



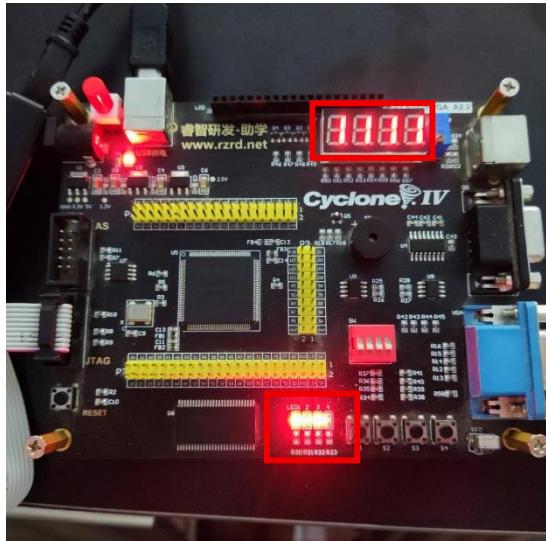
13. SEL=1100 (XOR A[^]B)



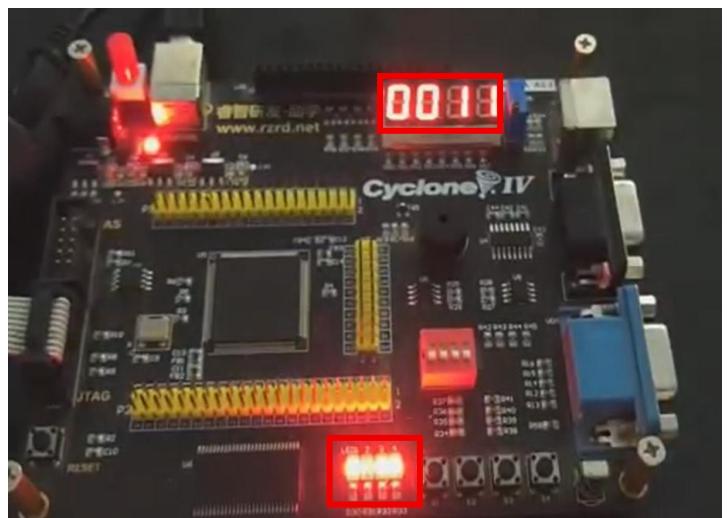
14. SEL=1101 (XNOR ~(A[^]B))



15. SEL=1110 (NAND $\sim(A \& B)$)



16. SEL=1111 (NOR $\sim(A|B)$)



Conclusion of FPGA Section

The FPGA implementation successfully demonstrated the **practical functionality** of the 4-bit ALU design. The use of real hardware allowed us to **verify the correctness of our Verilog code** beyond simulation. Although this design targets small bit-width operations, it reflects a complete digital system pipeline—from logic design to hardware realization. This semi-custom implementation confirms that the design is **functional, reliable, and adaptable** for future extensions on larger FPGAs.