Section 4 – Processor units

1. 16-bit ALU Design

Objective:

Design and verify a 16-bit Arithmetic Logic Unit (ALU) that supports multiple operations using SystemVerilog.

Requirements:

1. **ALU Operations:**
    - Addition (`ADD`)
    - Subtraction (`SUB`)
    - Bitwise AND (`AND`)
    - Bitwise OR (`OR`)
    - Bitwise XOR (`XOR`)
    - Shift operations (`SHIFT_LEFT`, `SHIFT_RIGHT`)
2. **Operation Select:**
    - Use a **SystemVerilog `enum`** for operation selection.
      Example:

      ```
      typedef enum logic [2:0] {
          ADD,
          SUB,
          AND,
          OR,
          XOR,
          SHIFT_LEFT,
          SHIFT_RIGHT
      } alu_op_t;
      ```
3. **Inputs/Outputs:**
    - `input logic [15:0] A, B` — operands
    - `input alu_op_t OP` — operation select
    - `output logic [15:0] RESULT` — ALU result
    - `output logic CARRY, ZERO` — optional flags (carry, zero)
4. **Verification:**
    - Create a **SystemVerilog testbench**.
    - Use **randomized input stimulus with constraints**:
        - Ensure `A` and `B` are 16-bit random values.
        - Constrain shift operations to `[0:15]`.
    - Check correctness of all operations.
    - Print results in a readable format.

2. Synchronous FIFO Design

Objective:

Design and verify a parameterized synchronous FIFO using SystemVerilog.

Requirements:

1. **FIFO Design:**
    - Parameterized **depth** (e.g., 8, 16, 32).
    - **Width fixed at 16 bits** (same as ALU result width).
    - Synchronous write and read (clocked).
    - Signals:
        - `input logic clk, rst, write_en, read_en`
        - `input logic [15:0] data_in`
        - `output logic [15:0] data_out`
        - `output logic full, empty`
2. **Testbench:**
    - Randomized write/read sequences.
    - Check that FIFO output matches expected queue behavior.