16-bit Arithmetic Logic Unit (ALU) Design Package

Section 1: Assignment Brief

1. Objectives

The goal of this assignment is to design and implement a complete, 16-bit **combinational Arithmetic Logic Unit (ALU)** in VHDL or Verilog, implementing all 20 specified functions.

This project focuses on:

1. Mastering combinational logic design.
2. Implementing efficient arithmetic and shift/rotate circuits.
3. Generating accurate status flags for error checking.

Core Component Structure

The ALU is a purely combinational circuit with the following ports:

Port	Direction	Width (bits)	Description
A	Input	16	First Operand
B	Input	16	Second Operand
F	Input	5	Function Select Code (F[4:0])
Cin	Input	1	Carry In (Used for ADD/SUB with carry/borrow)
Result	Output	16	Output of the selected operation
Status	Output	6	**Status Flags [C, Z, N, V, P, A]** (Carry, Zero, Negative, Overflow, Parity, Aux. Carry)

2. Function Table (F[4:0] Control Code)

Your ALU must implement the following 20 functions.

F[4:0] Code	**Operation**	**Required Flags**
00000	N/A	All
00001	**INC**	C, Z, N, V, P, A
00010	N/A	All
00011	**DEC**	C, Z, N, V, P, A
00100	**ADD**	C, Z, N, V, P, A
00101	**ADD_CARRY**	C, Z, N, V, P, A
00110	**SUB**	C, Z, N, V, P, A
00111	**SUB_BORROW**	C, Z, N, V, P, A
01000	**AND**	Z, N, P
01001	**OR**	Z, N, P
01010	**XOR**	Z, N, P
01011	**NOT**	Z, N, P
10000	**SHL**	C (bit shifted out), Z, N, P
10001	**SHR**	C (bit shifted out), Z, N, P
10010	**SAL**	C (bit shifted out), Z, N, P
10011	**SAR**	C (bit shifted out), Z, N, P
10100	**ROL**	C (bit rotated out/in), Z, N, P
10101	**ROR**	C (bit rotated out/in), Z, N, P

10110	**RCL**	C (bit rotated out/in), Z, N, P
10111	**RCR**	C (bit rotated out/in), Z, N, P

3. Status Flags Definition (6 bits: C, Z, N, V, P, A)

The 6-bit Status output must be defined as **{Carry, Zero, Negative, Overflow, Parity, Auxiliary Carry}** in that exact bit order:

- **C (Carry/Borrow):** Set if an arithmetic operation generates a carry out of the MSB or requires a borrow. For shift/rotate, this is the bit shifted/rotated out.
- **Z (Zero):** Set if the 16-bit Result is exactly zero (16'h0000).
- **N (Negative):** Set if the MSB of the Result is '1'.
- **V (Overflow):** Set if the result of a *signed* arithmetic operation is too large to fit in 16 bits.
- **P (Parity):** Set if the 16-bit Result has an **even** number of '1' bits (Even Parity).
- **A (Auxiliary Carry):** Set if an arithmetic operation generates a carry from **bit 3 to bit 4** (lower nibble to upper nibble of the LSB).

4. Implementation Requirement:

You can use whatever tool for simulation you want. If you don’t have any tool available you can use **EDA Playground** for simulation and verification. This is an online tool you can use it, it doesn’t require you to download any software on your laptop.

Website Link: https://www.edaplayground.com/

Key Task: Ensure you enable **VCD dump** in the run options and analyze the waveforms. Your submission should include the final HDL file and a screenshot of the waveform.

5. Submission Details

The assignment is to be made by **2 individuals maximum.** The deadline for submission is **29-11-25 11:59 PM.** The submitted file should as the following:

- A single PDF file named by the ID of student#1_ID of student#2.pdf (Ex. 92147_921457.pdf).
- The PDF should have 3 main points:
  - The verilog/VHDL code of the ALU
  - The verilog/VHDL code of the testbench(the code you used to test your ALU)
  - Screenshot of waveforms of your ALU verification
- You can only submit your pdf 1 time do not submit more than one time
- There are grades on following the rules, submitting before the deadline and neat reports and ofcourse the ALU design itself and the verification.
- The submission should be made in the google classroom of code: **mb6bjpzm.** Any questions or problems should be asked on the assignment post in the classroom, don't use emails or any other communication method.

- Only 1 of the students should submit the PDF file not both of them.