# Verilog

# Simple ALU with accumulator

**Objective**

To learn designing basic combinational/sequential circuits in Verilog.

**Assignment-1: 8-bit ALU Design**

Write Verilog code for an 8-bit Arithmetic and Logic Unit, and verify your design.

**Port definition**:



**Function description:**

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction | opcode | Operation | note |
| Add | 0000 | result =opa + opb + cin;  cout contains the carry | addition |
| INV | 0001 | result = ~opa  cout = 0 | Bitwise Invert |
| OR | 0010 | result = opa or opb  cout = 0 | Bitwise OR |
| AND | 0011 | result = opa and opb  cout = 0 | Bitwise AND |
| SHL | 0100 | result = {opa[6:0], 1’b0}  cout = 0 | Logical shift left(1 bit) |
| SHR | 0101 | result = {1’b0, opa[7:1] }  cout = 0 | Logical shift Right(1 bit) |
| ROL | 0110 | result = { opa[6:0],opa[7]}  cout = 0 | Rotated shift left |
| ROR | 0111 | result = { opa[0],opa[7:1]}  cout = 0 | Rotated shift right |
| CMP | 1001 | cout = 1 if opa > opb, otherwise cout = 0  result = 0 | comparison |

**Assignment-2: 8-bit ALU with an accumulator**

The ALU is the same as you’ve designed for assignment-1. The accumulator is an edge triggered register with asynchronous reset. Write Verilog code for this 8-bit ALU with an accumulator register.



Run simulation of you design: use any HDL simulator like iverilator/modelsim etc., and include following materials in your report:

--Snapshots of the waveform window when you run the testbenches.

--Snapshots for assignment 2: configure the input vector in your testbench, use the module you designed to calculate ROL((8’h05 + 8’h23) AND 8’h56) + 8’hA0