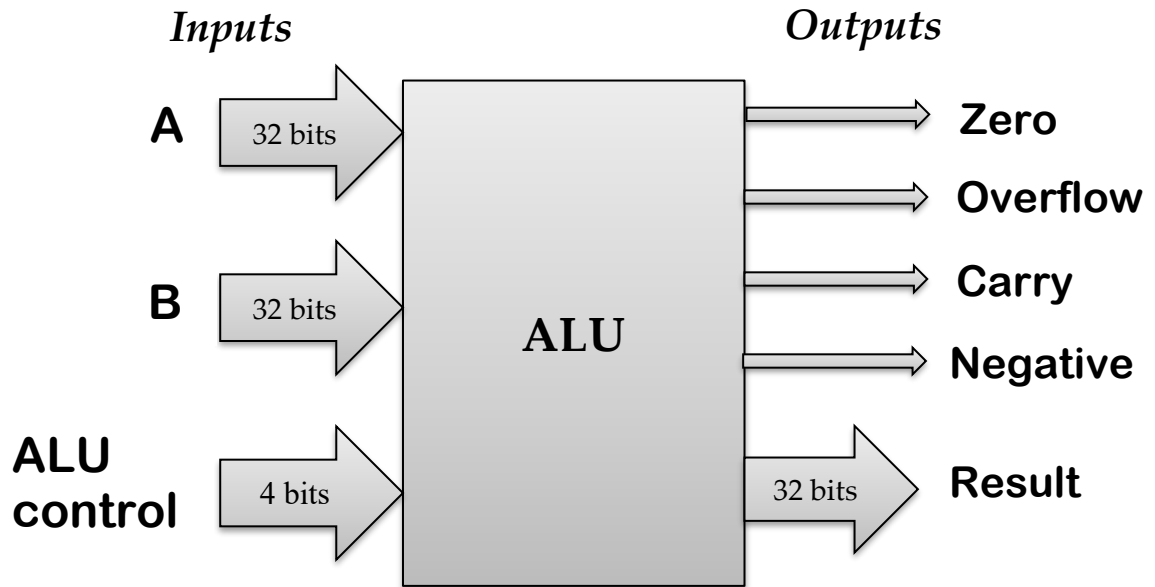

Digital Design with Verilog

Date: June 3, 2025

Designing of an ALU

1. In this tutorial, we shall be designing an ALU which takes two 32-bit operands, a 4-bit control signal, and produces a 32-bit result output.



- (a) Please do set the zero, carry, overflow, and negative flags.
- (b) Following is the relation between the two inputs and output depending on the control signals.

Table 1: Output relation based on control signals

Control signal value	Operation
4'b0000	$A + B$
4'b0001	$A - B$
4'b0010	$A \& B$
4'b0011	A / B
4'b0100	$A \wedge B$
4'b0101	{31'b0, slt}
4'b0110	{31'b0, sltu}
4'b0111	{A[31:12], 12'b0}
4'b1000	$A + \{B[31:12], 12'b0\}$
4'b1001	$\{B[31:12], 12'b0\}$
4'b1010	$A << B$
4'b1011	$A >>> B$

4'b1100	$A \gg B$
anyother	set to 32'bx

(Hint: For implementation purposes, you can consider both A and B as signed numbers then it will be easy to implement \gg (shift right arithmetic) operation.)

slt : set less than. Output is set to 1 if $A < B$.

$slt = (A[31] == B[31]) ? (A < B) : A[31]$; // because for signed numbers, of both are of same sign, we can compare A and B, but if they are of different sign we can take the MSB of A. if A is positive and B is negative \rightarrow A is not less than B, $slt = 0$, i.e., A[31].

if A is negative and B is positive \rightarrow A is definitely less than B, so $slt = 1$ i.e., A[31].

sltu: set less than unsigned: Output is set to 1 if $A < B$ and A and B are treated as unsigned numbers.

$sltu = A < B$;

Overflow: