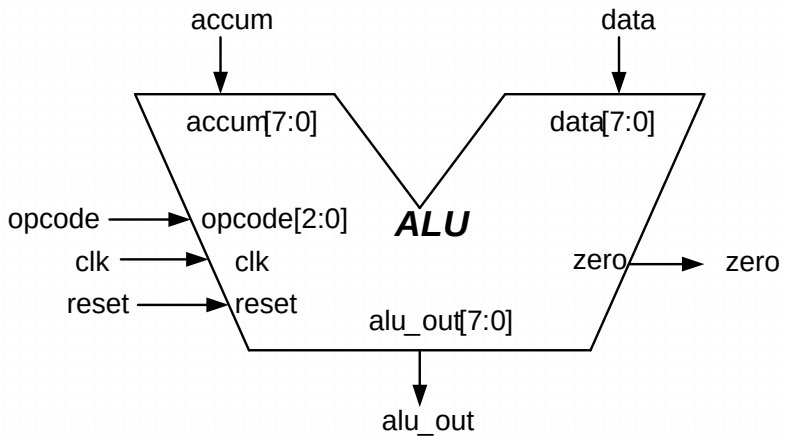
Lab: ALU

- Please modeling an Arithmetic Logic Unit (ALU)
- Specifications
  - Module name : alu
  - Input pins : accum[7:0], data[7:0], opcode, clk, reset
  - Output pins : zero, alu_out[7:0]
  - Function :



1. Model the ALU in the *alu.v* . Model port interface according to the symbol view on the over. Model the functionality according to the following specifications.
   1). All inputs and outputs(exclude "*zero*" signal) are synchronized at clock rising edge .
   2). It is a synchronous-reset architecture. *alu_out* become 0 when the reset equal 1.
   3). *accum*, *data* and *alu_out* are using 2's complement expression.
   4). The *zero* bit become 1 when the *accum* equal 0, and is 0 otherwise.
   5). The *alu_out* becomes 0 when *opcode* is X(unknow).
   6). Its value is determined when the ALU decode the 3-bits *opcode* and performs the appropriate operation as listed below.

opcode	ALU operation	
000	Pass accumulator	
001	accumulator + data	(addition)
010	accumulator - data	(subtraction)
011	accumulator AND data	(bit-wise AND)
100	accumulator XOR data	(bit-wise XOR)
101	ABS(accumulator)	(absolute value)
110	NEG(accumulator)	(negate value)
111	Pass data	

p.s. 1. opcode 為 101: 使用 accum[7]當作 signed bit 判斷正負

2. opcode 為 110: 直接對 accum 做 2 補數運算

2. Check the result.

End of Lab