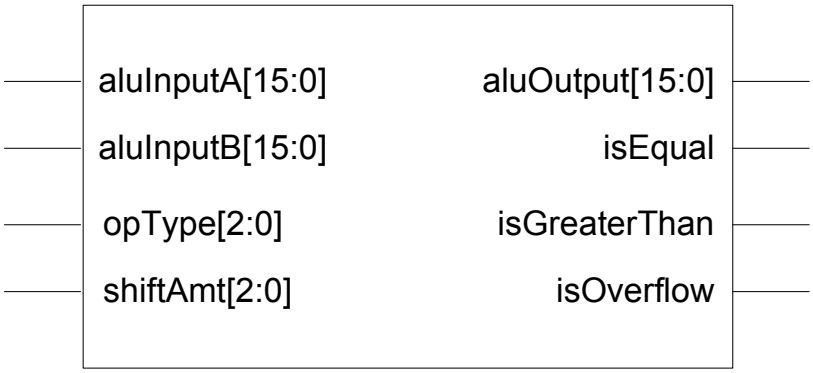
BSUCA12 ALU

Must be submitted electronically by Midnight on Friday, March 30

In this part of the project, you will build an ALU that implements all of the arithmetic and logic functions of BSUCA12 (refer to the project specs). The figure shown below illustrates the ALU, including the exact signal names that you must use in your design (to facilitate testing). For ALU operations that take a single register input (i.e., shifts), assume that the input arrives on `aluInputA`. The desired ALU operation is selected by the input `opType` as shown in the table given below. Shift amounts are unsigned 3-bit integers. The high level module must be named `alu` and file name must be `alu.v`

- If the operation is a subtraction, then
  - the output `isEqual=true` if `aluInputA==aluInputB`,
  - the output `isGreaterThan=true` if `aluInputA > aluInputB` (this is a *signed* comparison).
- If the operation is not a subtraction, then
  - `isEqual` and `isGreaterThan` are 'don't care' outputs.
- If there is an overflow. All outputs are 'don't care'.
  Note: overflow is not the same as carry_out from the adder.



operation	opType
ADD	001
SUB	010
NAND	111
XOR	100
SHL	101
SHRA	110