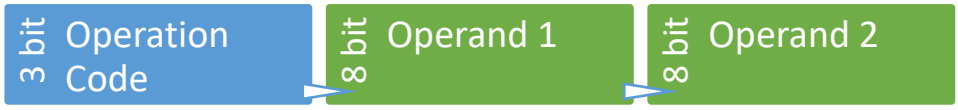
SIMPLE CPU DESIGN: VERILOG LAB 4

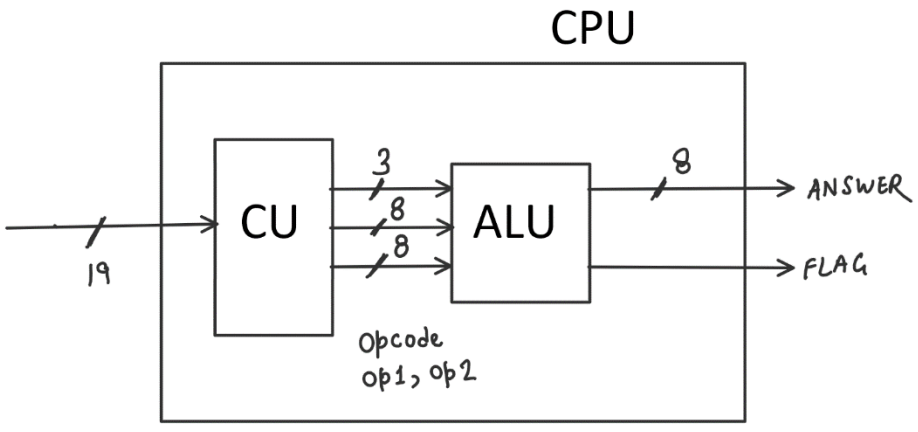
APPROACH

1) The CPU has two components Control Unit (CU) and ALU (Arithmetic and Logical Unit)
2) **Control Unit:**
   a) It generates the one-hot output (8-bit) from the 3 MSBs and passes it to ALU using 3x8 decoder.
   b) It separates the two operands (bits [15:8] and [7:0]) and passes it to ALU.
3) **Arithmetic and Logical Unit**
   a) It has 8 modules (and necessary submodules for them) for the 8 operations.
   b) The input operands are fed into all the module circuits.
   c) The correct output is selected using the one-hot input from CU (by ANDing respective bit with the output of each module).
   d) There is an additional flag bit which gets set whenever there is a carry or borrow.

DIAGRAM

INSTRUCTION





MODULES

1) **Control Unit**
   a) **3x8 Decoder**
      - It creates 8-bit one-hot output using the 3 input bits by generating all the minterms.
2) **Arithmetic and Logical Unit**
   a) **XOR bitwise**
      - Both operands were XORed using an 8-bit array of XOR gates.
   b) **ADDITION**
      - The 8-bit adder was made using 8 full adder modules (Ripple Carry Adder).
      - Each full adder module had 2 half adder modules along with an OR gate
   c) **SUBTRACTION**
      - Subtraction was performed by adding the 2's complement of second operand to the first.
      - This required an 8-bit adder module instantiation from the previous part.
   d) **AND bitwise**
      - Both operands were ANDed using an 8-bit array of AND gates.
   e) **OR bitwise**
      - Both operands were ORed using an 8-bit array of OR gates.
   f) **NOT bitwise**
      - First operand is NOTed using an 8-bit array of NOT gates.
   g) **INCREMENT**
      - **Optimisation:** Instead of instantiating a full 8-bit adder module, I created an incrementer using only 8 half adders.
   h) **DECREMENT**
      - **Optimisation:** Instead of instantiating a full 8-bit adder module, I created a decrementer using only 8 half adders.
   i) **SELECTING THE CORRECT OUTPUT**
      - Each output is ANDed with the decoder input to get the required answer.

OUTPUT

- Final Output