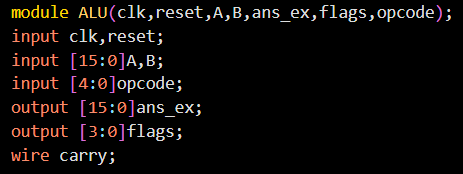
**Saumya Ramani**

**22bce295**

**Computer Architecture**

**Innovative Assignment**

**Title:- Arithmetic and Logical Unit using Verilog**



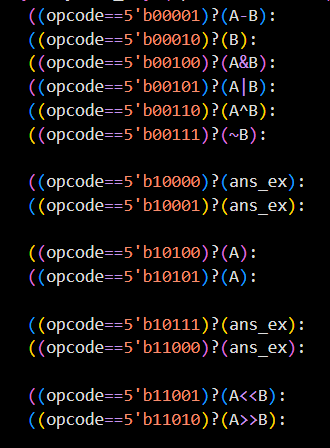
Represents the module which is created with the name of ALU.

And the parameters passed are clk, reset, A, B etc.

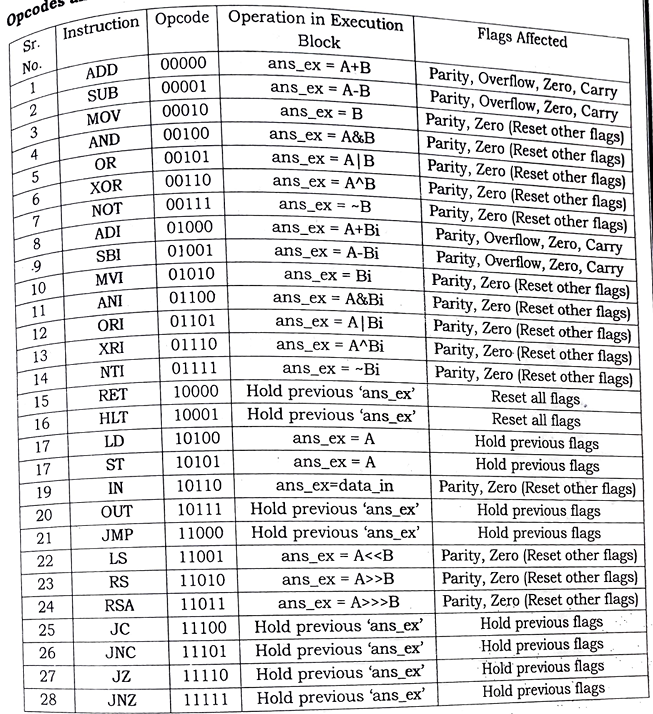
And the input , output shows the type of the attribute that whether it is going to be taken as input or it will be shown as output.

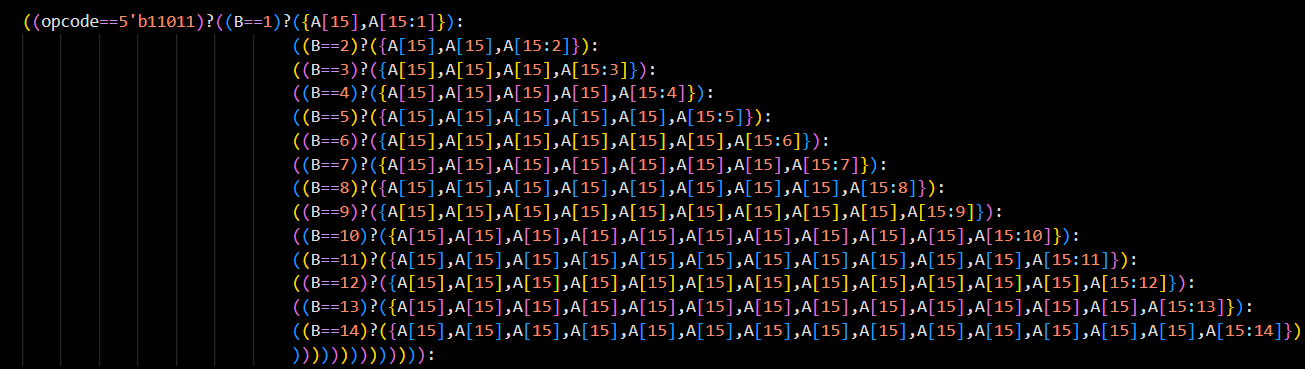


Here, it shows that the value is being assigned to the carry and the ans\_ex such that the final ans will be stored in ans\_ex and if the carry is present than it will be stored in carry variable.



Here, those are the nested ternary operators such that if one opcode is true than the corresponding operation will be executed and if it is false, then the other downward line of code will be executed correspondingly. The opcodes and their operations are mentioned below in the tabular format.

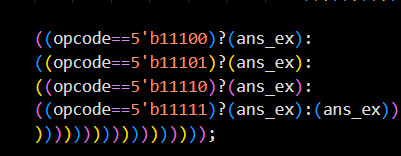




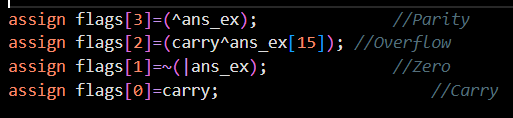
Here, the above logic shows the functionality of the arithmetic shift and the number of times the arithmetic shift is initiated.



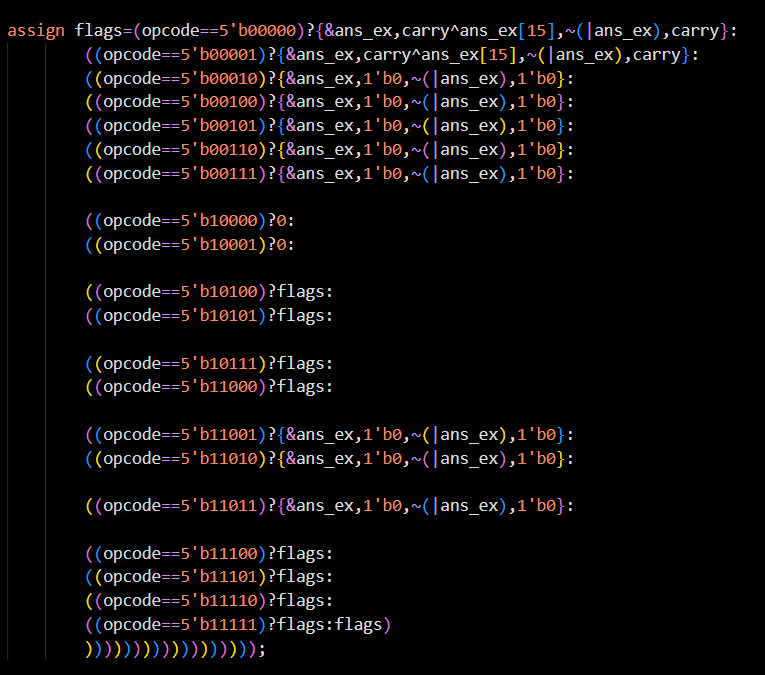
Now if the arithmetic shift is initiated 15 times, then the whole answer will be equal to the MSB(Most Significant Bit) of the answer.



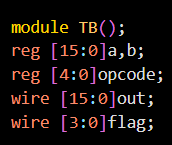
Those are the remaining opcodes and those don’t denote any kind of operations because those are the operations related to the holding and the changes will be shown into the flags.



Those denote the updation conditions of the flags as per the necessary conditions.

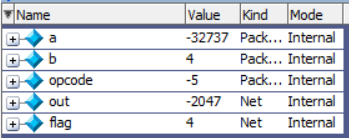


Above code shows the implementation of the flag updation without using the functions.

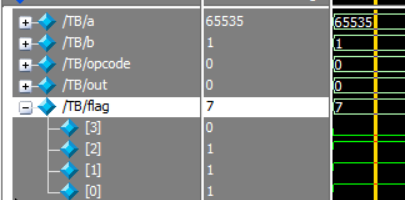


The above module is formed same to the original ALU module formed. It is mainly formed because of the execution of the module formed with the desired data and the opcode.

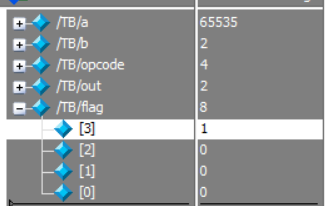
**Output**

****

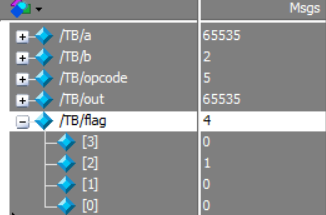
Here, the opcode is arithmetic right shift and the answer is (-32737/16) and so the answer is -2047.



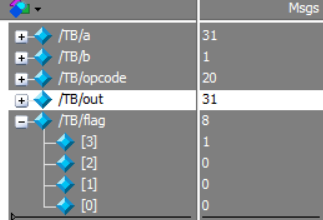
Here, the opcode Is 00000 so the addition will take place. And the corresponding status bits flags are updated.



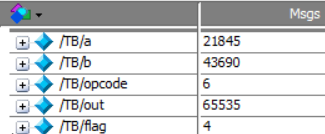
Here, the opcode is 00100, so the AND operation will take place and as the sign bit is inverted, the parity bit is changed to 1.



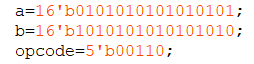
Here, the OR operation is executed and the number B is lesser than the number A so the output will be same is number A.



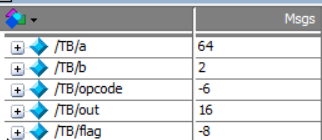
Here, the opcode is 20, so the LOAD operation will be executed and the value of A will be stored into the ans\_ex.



Both the numbers are formulated such as all have alternating bits 1 and the bits on the same index are different if we consider both the numbers.



So after XOR operations, all the 16 bits will be having 1 and hence the answer will be the (2^16 - 1).



The opcode is 6 so the right shift will take place and the 64 will be divided by 4 and the final answer will be 16.