ALU Control

ALU control for the a 64-bit Integer ALU. The control unit takes in 2 inputs:
(a) A 2-bit input that distinguishes the nature of operation to perform.
(b) Minimal bits from the funct3 and funct7 bits of the instruction to identify the operation.

TRUTH TABLE :

ALU Op		Funct - 7							Funct - 3			Op			
0	0	X	X	X	X	X	X	X	X	X	X	0	0	1	0
X	1	X	X	X	X	X	X	X	X	X	X	0	1	1	0
1	X	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	X	0	1	0	0	0	0	0	0	0	0	0	1	1	0
1	X	0	0	0	0	0	0	0	1	1	1	0	0	0	0
1	X	0	0	0	0	0	0	0	1	1	0	0	0	0	1

The highlighted bits are the minimal bits from funct - 3 and funct - 7 to identify the operation

VERILOG CODE :

```
hackspot@hackspot-inspiron-3521:~/code/Verilog/A3$ bat CU.v
       File: CU.v
   1   /*
   2           K Rahul Reddy  17CO119
   3           Sushruth V     17CO148
   4           24 October 2018
   5
   6           funct7[5] and funct3[2:0] are required to identify operation.
   7   */
   8   module CU(output [3:0] Op, input[1:0] ALU_Op, input[6:0] funct7, input[2:0] funct3);
   9       wire F0 = funct3[0];
  10       wire F1 = funct3[1];
  11       wire F2 = funct3[2];
  12       wire F3 = funct7[5];
  13       assign Op[3] = 0;
  14       assign Op[2] = ALU_Op[0] | (ALU_Op[1] & F3 & ~F2 & ~F1 & ~F0);
  15       assign Op[1] = ~F1 | ~ALU_Op[1];
  16       assign Op[0] = ALU_Op[1] & F1 & ~F0;
  17   endmodule
```

OUTPUT :

```
hackspot@hackspot-inspiron-3521:~/code/Verilog/A3$ iverilog -o tb_CU.vvp tb_CU.v
hackspot@hackspot-inspiron-3521:~/code/Verilog/A3$ vvp tb_CU.vvp
VCD info: dumpfile CU.vcd opened for output.
Case 0: Add(ALU_Op code)
ALU_Op =  00            funct7 =  0100111              funct3 =  111

Opcode = 0010

Case 1: Subtract(ALU_Op code)
ALU_Op =  01            funct7 =  0000000              funct3 =  111

Opcode = 0110

Case 2: Add(R-type)
ALU_Op =  10            funct7 =  0000000              funct3 =  000

Opcode = 0010

Case 3: Subtract(R-type)
ALU_Op =  10            funct7 =  0100000              funct3 =  000

Opcode = 0110

Case 4: And(R-type)
ALU_Op =  10            funct7 =  0000000              funct3 =  111

Opcode = 0000

Case 5: Or(R-type)
ALU_Op =  10            funct7 =  0000000              funct3 =  110

Opcode = 0001
```