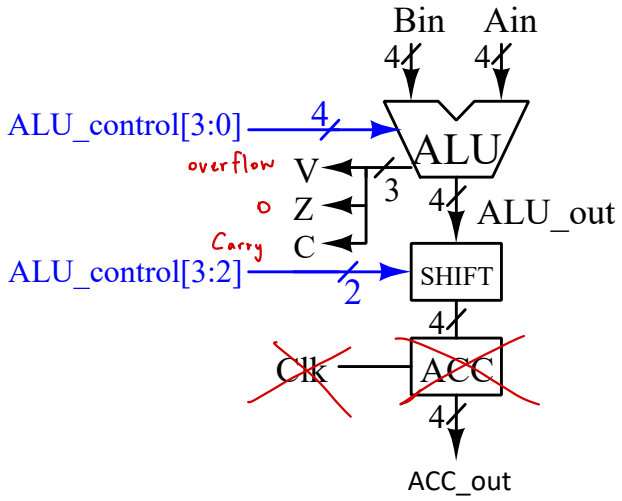
Assembly + Machine language

Opcode	Operand	Function	RTN Register Transfer Notation	Mnemonic	Machine code
0000	Address	Load	ACC ← M[Address]	LDA M[Address]	0000 aaaa
0001 0000	Address	Add w/ C	ACC ← ACC + M[Address] + C	ADC M[Address]	0001 aaaa
0010	Address	Subtract w/ C	ACC ← ACC - M[Address] - C	SBC M[Address]	0010 aaaa
0011	Address	Store	M[Address] ← ACC	STA M[Address]	0011 aaaa
0100	xxxx	Clear	ACC ← 0, C ← 0	CLA	0100 xxxx
0101 0001	Address	AND	ACC ← ACC & M[Address]	AND M[Address]	0101 aaaa
0110 0010	Address	OR	ACC ← ACC | M[Address]	OR M[Address]	0110 aaaa
0111 0011	Address	NOT	ACC ← ~ACC	NOT	0111 xxxx
10xx	xxxx	SHL	ACC ← ACC ← 2	SHL	10xx xxxx
11xx	xxxx	SHR	ACC ← ACC → 2	SHR	11xx xxxx

Procedure

Design a 4-bit ALU that can perform 10 functions, only the ALU part, not control parts and memories. Implement it on Atrix 7 FPGA board. Use DIP switch as Ain, Bin, and ALU_control, and LED to display the contents of ACC.



Example Code

```
module ALU (
    input [3:0] ALU_control, Ain, Bin,
    output reg [3:0] ACC_out,
    output V, Z, C, Clk
);
    wire reg [3:0] ALU_out;

    // write your code here

endmodule
```

TA Signature: Lab11_2A_64,68.....

6/11/63