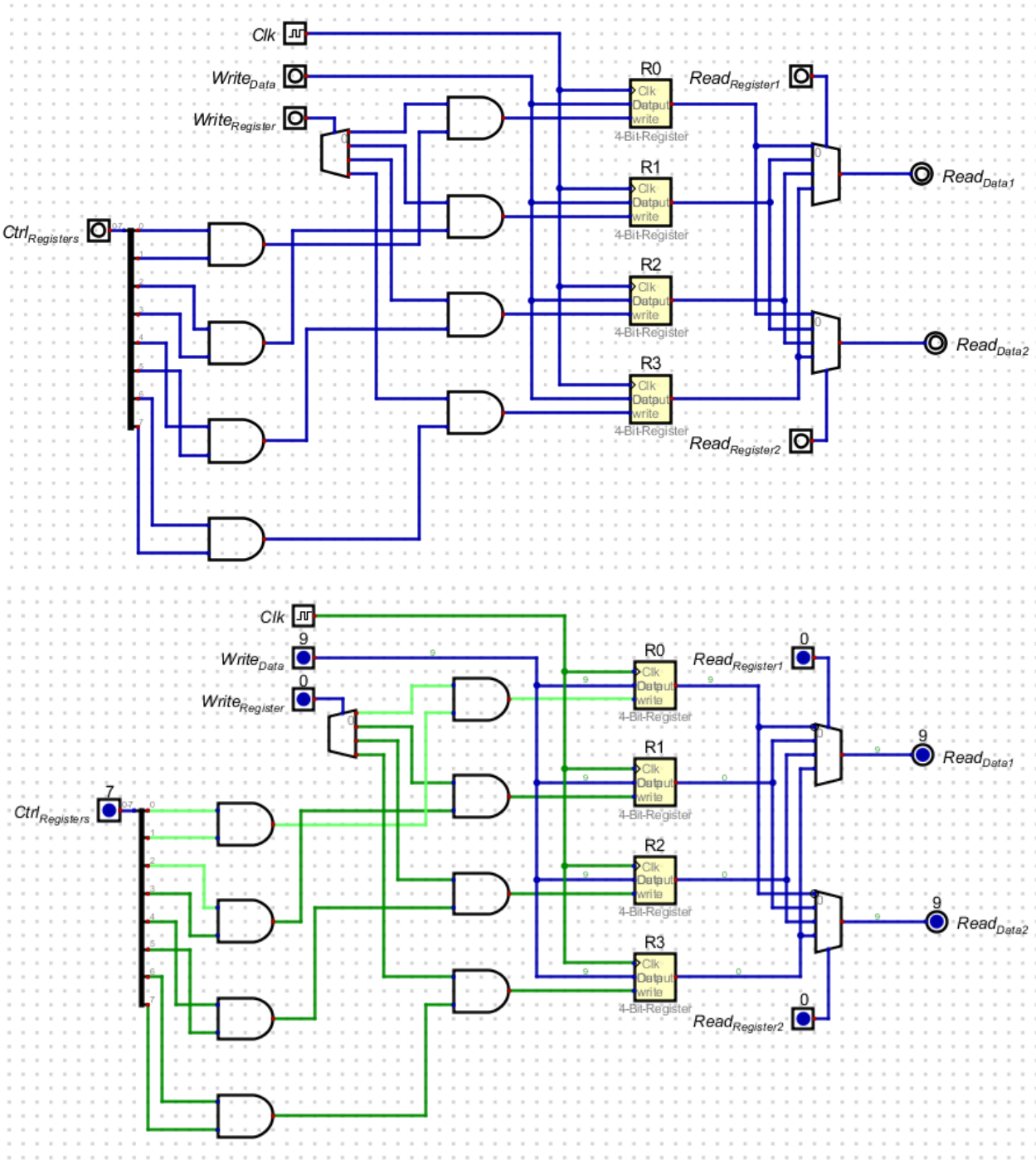
Github url:
https://github.com/Almo-o/Computer-Organization-and-Assembly/tree/main/Assignment-2

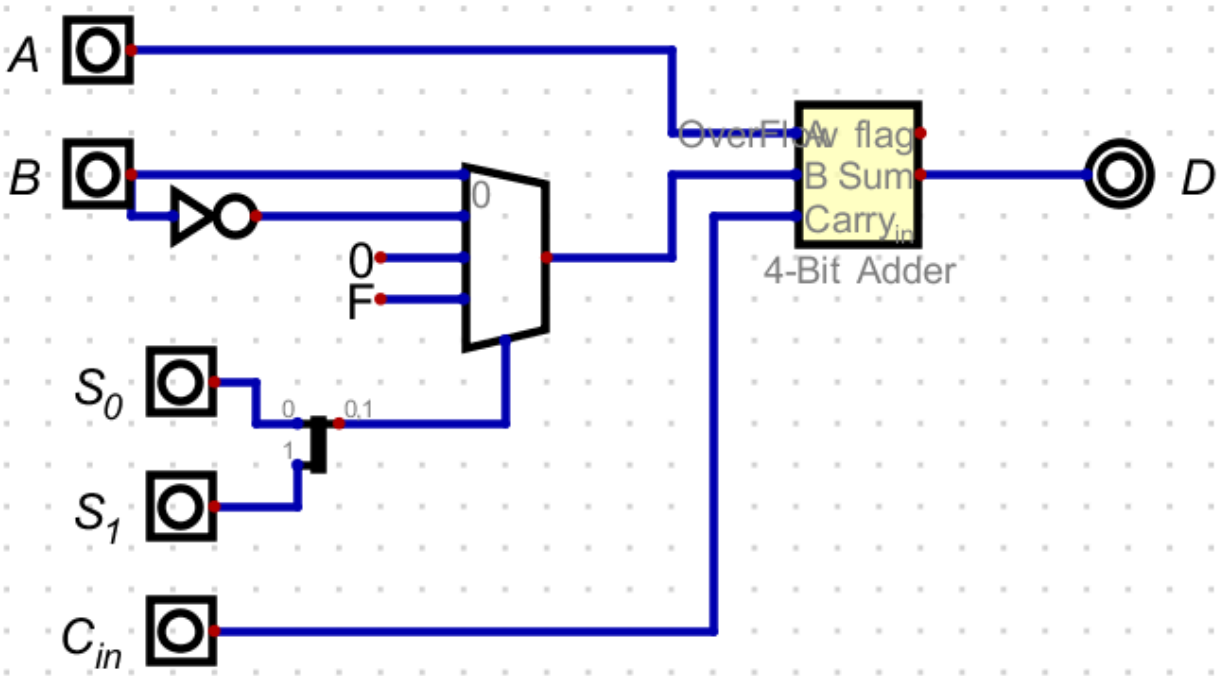
Brief Description:

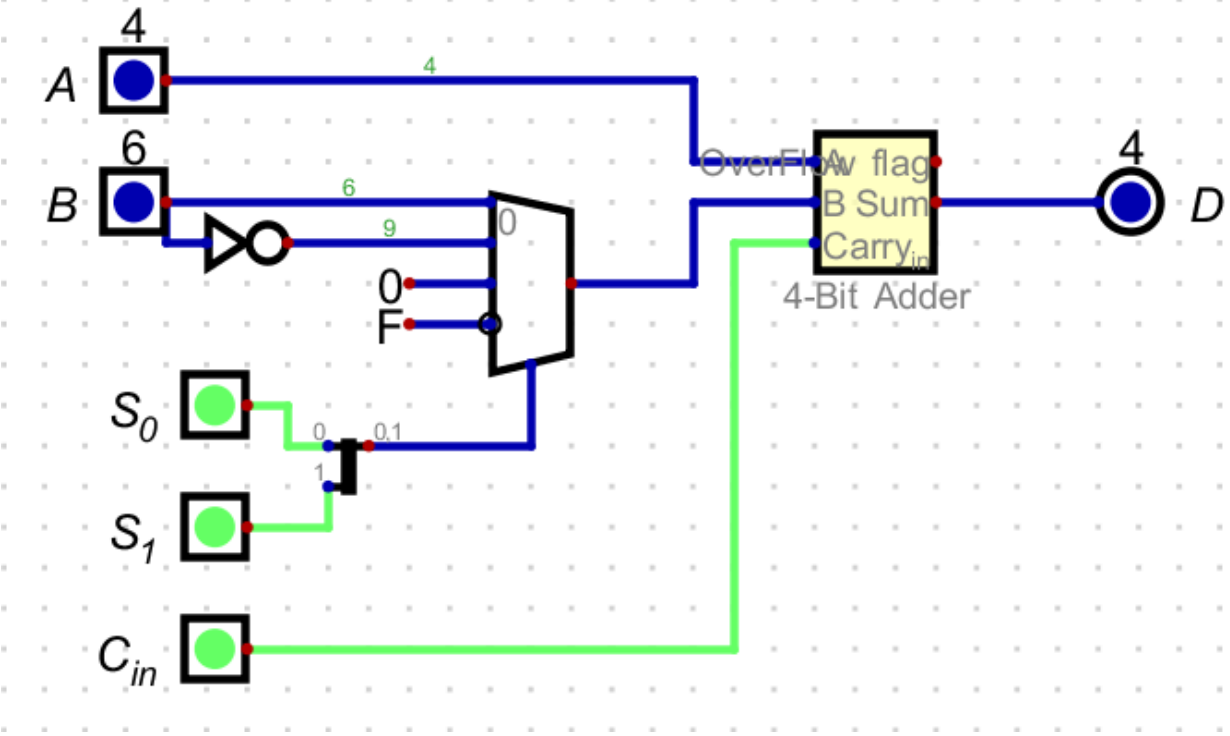
The register file is designed with two read ports and one write port to enable simultaneous data access. Two independent read address decoders fetch data from two registers via multiplexers, while a write decoder selects the target register for writing when the write-enable signal is active.

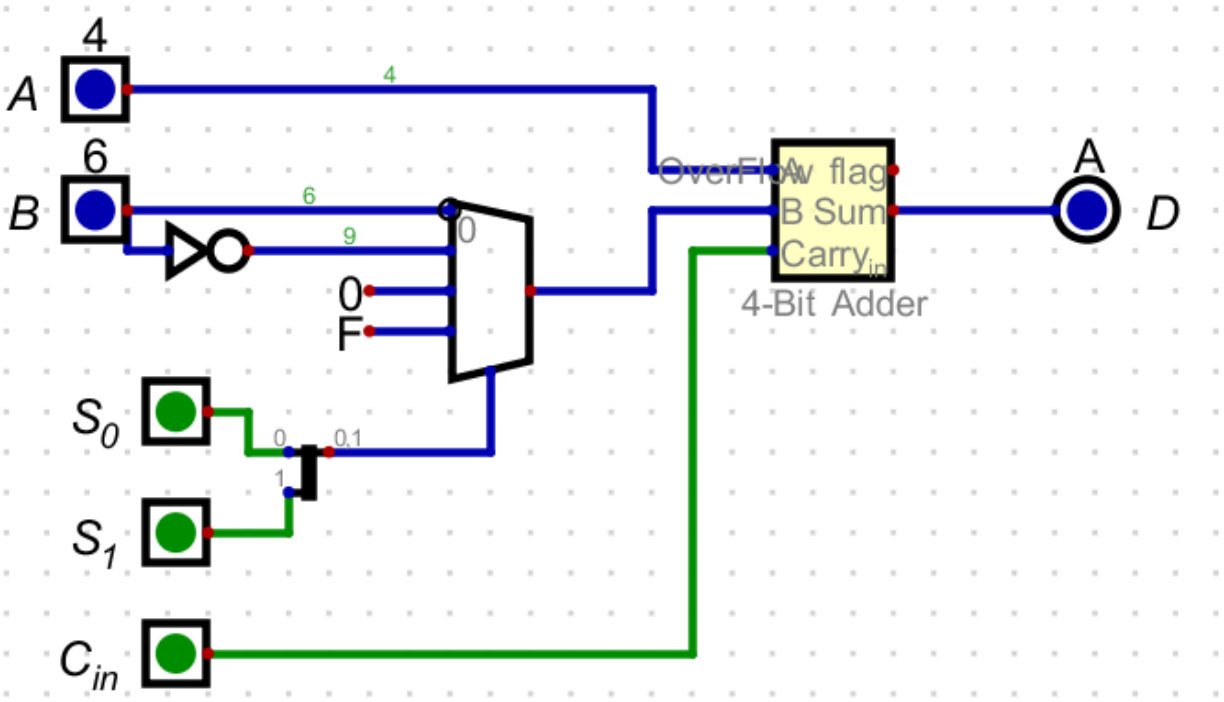
ALU: Components- a 4-bit adder, a multiplexer, and control signals (S1,S0,Cin). The MUX selects between inputs like B, -B (complement of B), 0000, or 1111, which are fed into the adder along with `A` and `Cin`. Operations such as addition, subtraction (via 2's complement), increment, decrement, and transfer are achieved by configuring the control signals to manipulate the adder's inputs dynamically. This design efficiently reuses hardware components for multiple functions.

Screenshots:


Clk
Write_Data
Write_Register
Ctrl_Registers
R0
R1
R2
R3
Clk
Datput
write
4-Bit-Register
Read_Register1
Read_Register2
Read_Data1
Read_Data2
9
0
7


A
B
0
0
F
S0
S1
Cin
OverFlow flag
B Sum
Carry in
4-Bit Adder
D


4
A
6
B
4
6
9
0
0
F
S0
S1
Cin
OverFlow flag
B Sum
Carry in
4-Bit Adder
4
D


4
A
4
6
B
6
9
0
0
F
S_0
0
0.1
1
S_1
C_{in}
OverFlow flag
B Sum
$Carry_{in}$
4-Bit Adder
A
D