

Github Url: <https://github.com/Almo-o/Computer-Organization-and-Assembly/tree/main/Assignment-3>

Design Choices:

ProgramCounter: designed using four flip flops to store the current count, and a 4-bit adder to calculate the next value, with a clock signal to constantly update.

Instruction Register: Made of four flip flops to hold the fetched input from memory. Updates its value based on a clock signal.

Instruction Memory: main components are four 4-bit instruction registers.

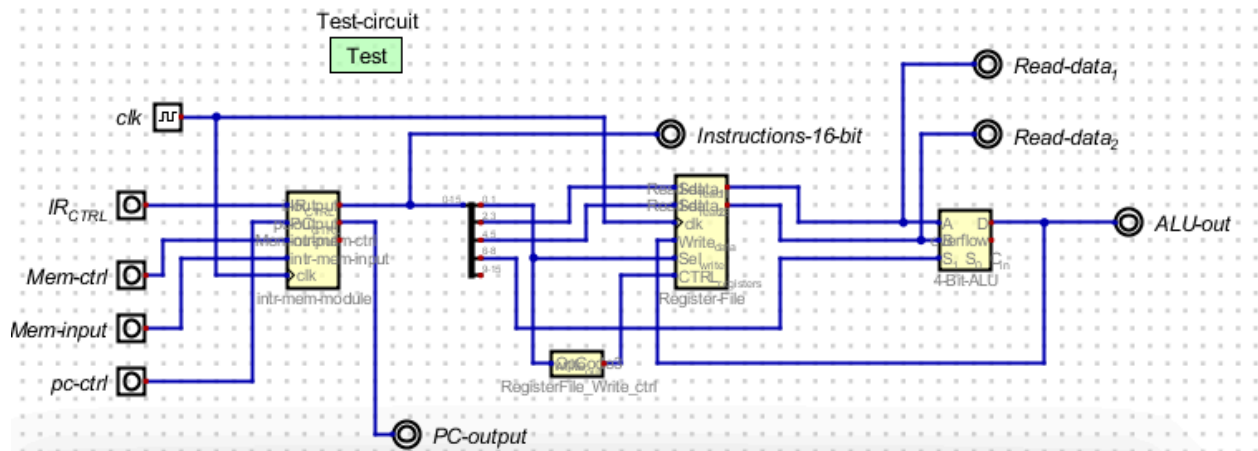
Instruction Decoder: implemented using a MUX, then the output is routed to other parts of the 4 bit processor, like Register file and Alu

RegisterFile: is designed with two read ports and one write port.. Two read address decoders fetch data from two registers via multiplexers, while a write decoder selects the target register for writing when the write signal is on.

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 wired into the adder along with A, and CIN. Operations such as addition, subtraction , increment, decrement, and transfer are then achieved by having specific s1,s0,cin configurations.

Screenshots NEXT PAGE:

4-bit Processor:



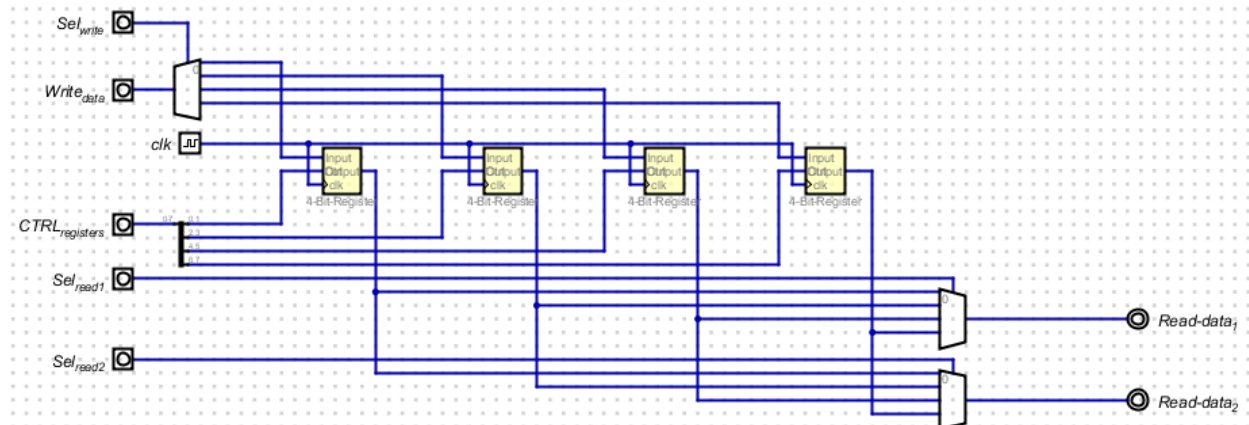
Test result

File View

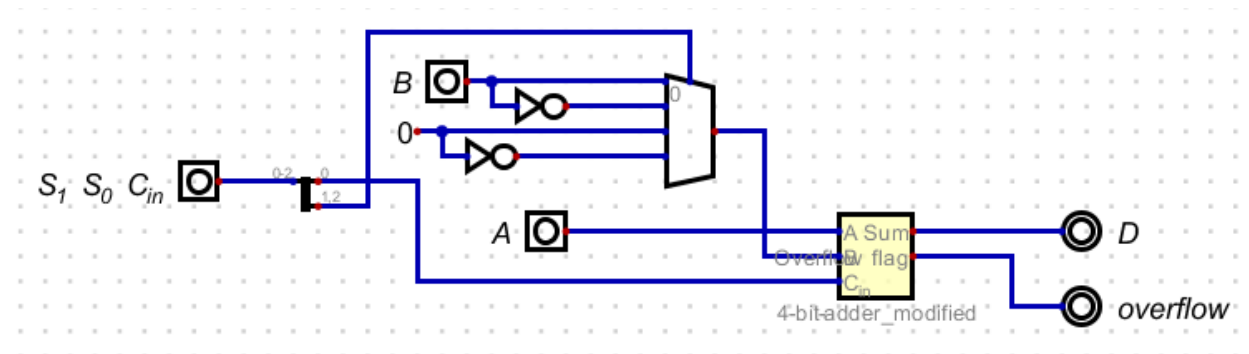
Test-circuit passed

	IR_CT...	Mem-...	Mem-i...	pc-ctrl	clk	PC-ou...	Instru...	ALU-out	Read-...	Read-...
L2	0	1	1	1	0	1	Z	0	0	0
L3	0	1	0	1	0	2	Z	0	0	0
L4	0	1	0	1	0	3	Z	0	0	0
L5	0	1	1	1	0	4	0	0	0	0
L6	0	1	2	1	0	5	Z	1	0	0
L7	0	1	3	1	0	6	Z	1	0	0
L8	0	1	4	1	0	7	Z	1	0	0
L9	1	0	5	1	0	8	0x4000	0	0	0
L10	1	0	6	1	0	9	Z	0	1	0
L11	1	0	7	1	0	A	Z	1	0	0
L12	1	0	0	1	0	B	Z	1	0	0
L13	1	0	0	1	0	C	0	0	0	0
L14	1	0	0	1	0	D	Z	E	0	0
L15	1	0	0	1	0	E	Z	E	0	0
L16	1	0	0	1	0	F	Z	E	0	0
L17	1	0	0	1	0	0	0	0	0	0

Register File:



ALU:



Instruction Memory + Four IRs:

