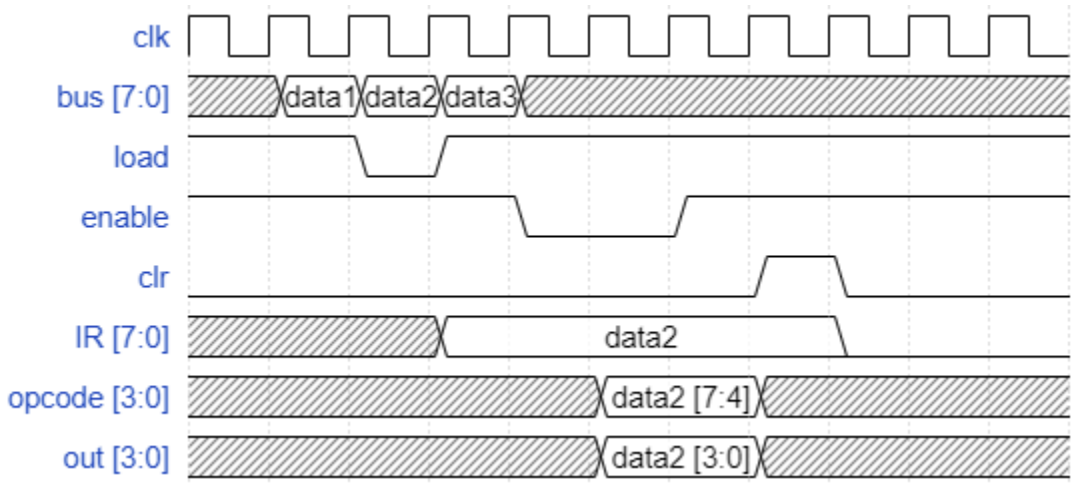
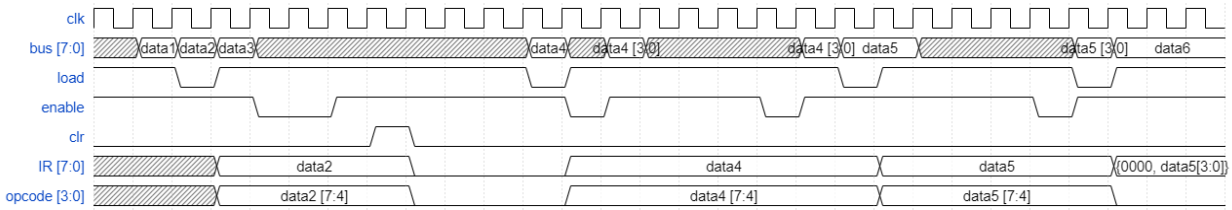
Instruction register

Label	Input/Output	Description
CLK [1 bit]	Input	Clock signal. Executes actions on rising edges.
W bus [8 bit]	Input	Takes 8 bits with the most significant 4 bits representing the opcode and the least significant 4 bits representing any other necessary value. Write them to the instruction register.
$\overline{L_I}$ [1 bit]	Input	Control signal that decides whether to read from the bus
$\overline{E_I}$ [1 bit]	Input	Control signal that decides tri-state buffer output to bus (drive register value if enabled, Z if disabled)
CLR [1 bit]	Input	Clears the instruction register's data.
Instruction register[3:0] [4 bit]	Output	Output to W bus
Instruction register[7:4] [4 bit]	Output	Output to controller/sequences



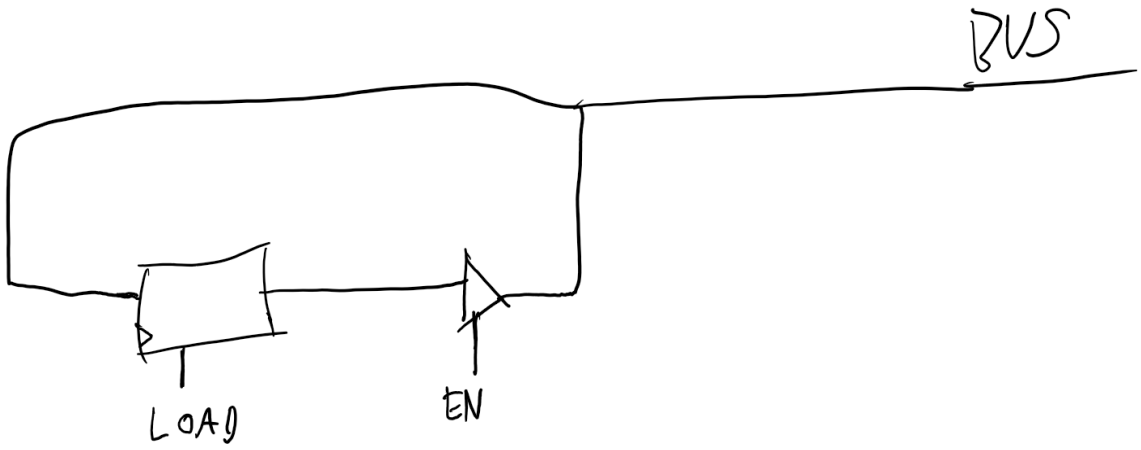
Test plan:
Ensure data is only loaded to the IR from the bus if load is on (active low). Ensure data is only driven out if the enable is on (active low). Ensure the two outputs opcode and out are the correct bits from the IR. Ensure when the CLR bit is set, IR is set to 0. Have data driven onto the W bus while load is at '1', IR should not change. Drive data onto the W bus while load is set to '0', we should observe IR updated with the data. Set enable pin to '1', should be no output. Set enable

pin to ‘0’, should be the correct split of the IR in the output. Set CLR pin to ‘1’, should clear IR. Confirm by enabling output once more and checking that both outputs are 0. Test with various data one after another.



The extended timing diagram shows edge cases where the enable and load control signals are activated close to each other.

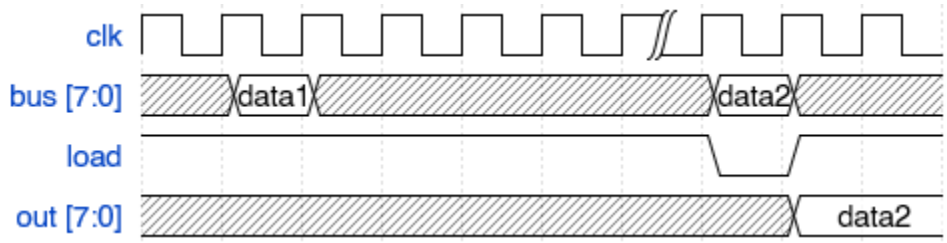
RACE CASE: if load and enable are active at the same clock cycle, then there will be a race condition between the bus’ value and register’s value. Since the bus is a wire and not a register, it would be like connecting the output of a flip flop to its input and having a value driven on that wire for a little bit. The register value might end up being what was already in the register or the previously driven wire value depending on timing.



Output register

Label	Input/Output	Description
CLK [1 bit]	Input	Clock signal. Executes actions on rising edges.
W bus [8 bit]	Input	Data from the bus lines that are to be written to the Output register.
$\overline{L_O}$ [1 bit]	Input	Control signal that decides whether to read from the bus

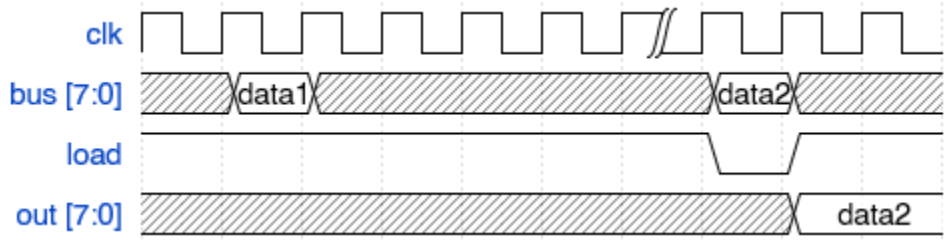
		and load onto the output register.
Output register [8 bit]	Output	Register data that will be written to the binary display.



Test plan:
Ensure that the output register is outputting the correct data. Test driving data onto the bus line and have the load at ‘1’, output should not change. Then test driving data onto the bus line and have the load at ‘0’, and check if the data is correctly outputting from the output register.

B register

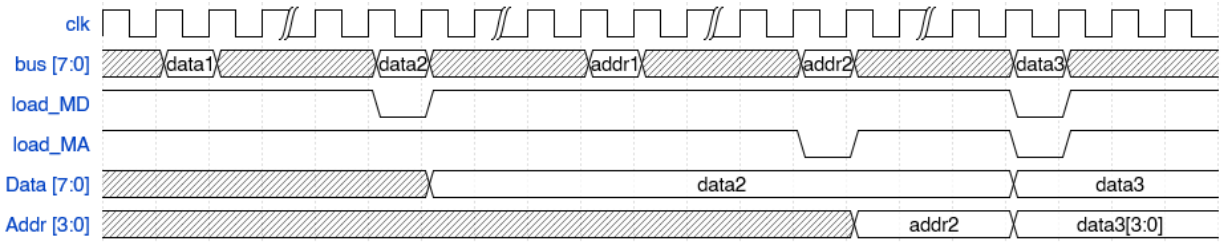
Label	Input/Output	Description
CLK [1 bit]	Input	Clock signal. Executes actions on rising edges.
W bus [8 bit]	Input	Data from the bus lines that are to be written to the B register.
$\overline{L_B}$ [1 bit]	Input	Control signal that decides whether to read from the bus and load onto the B register.
B register [8 bit]	Output	Register data that will be written to adder/subtractor.



Test plan:
Ensure that the B register is outputting the correct data. Test driving data onto the bus line and have the load at ‘1’, output should not change. Then test driving data onto the bus line and have the load at ‘0’, and check if the data is correctly outputting from B register.

Input and MAR

Label	Input/Output	Description
CLK [1 bit]	Input	Clock signal. Executes actions on rising edges.
W bus [8 bit]	Input	Data from the bus lines that are to be written either Input or MAR register. If writing to the MAR register, only 4 least significant bits are taken, the rest are discarded.
$\overline{L}_{MD}$ [1 bit]	Input	Control signal that decides if W bus data is to be written to the Input register. Should not be active at the same time as the $\overline{L}_{MA.}$control signal.
$\overline{L}_{MA}$ [1 bit]	Input	Control signal that decides if W bus data is to be written to the MAR register. Should not be active at the same time as the $\overline{L}_{MD.}$control signal.
Input register [8 bit]	Output	Register data to be written to memory.
MAR [4 bit]	Output	Register data taken by RAM that controls where the data is to be written.



Test plan:
Ensure both data out and address out take output correct values on the correct timing. Have data driven onto the W bus while both loads are at ‘1’, data and address should not change. Drive data onto the W bus while load_MD is set to ‘0’ and load_MA is set to ‘1’, we should observe data out take the value driven on the W bus, and make no change on address out. Drive data onto the W bus while load_MD is set to ‘1’ and load_MA is set to ‘0’, we should observe address out take the value of the least significant 4 bits on the W bus, and make no change to data out. Behavior if both loads are at ‘0’ is undefined, and will cause unintended consequences.

In that case, we will load the full 8 bits from the W bus into data, and the least significant 4 bits from the W bus (same data) should be written onto the address register.

- Opcodes are too long for an instruction register to be split into two.
- Memory doesn't have enough control pins
- "OUT byte D3 send a byte of data to the output register"
    - What do they mean by this? Send a byte? From where? How is it specified?