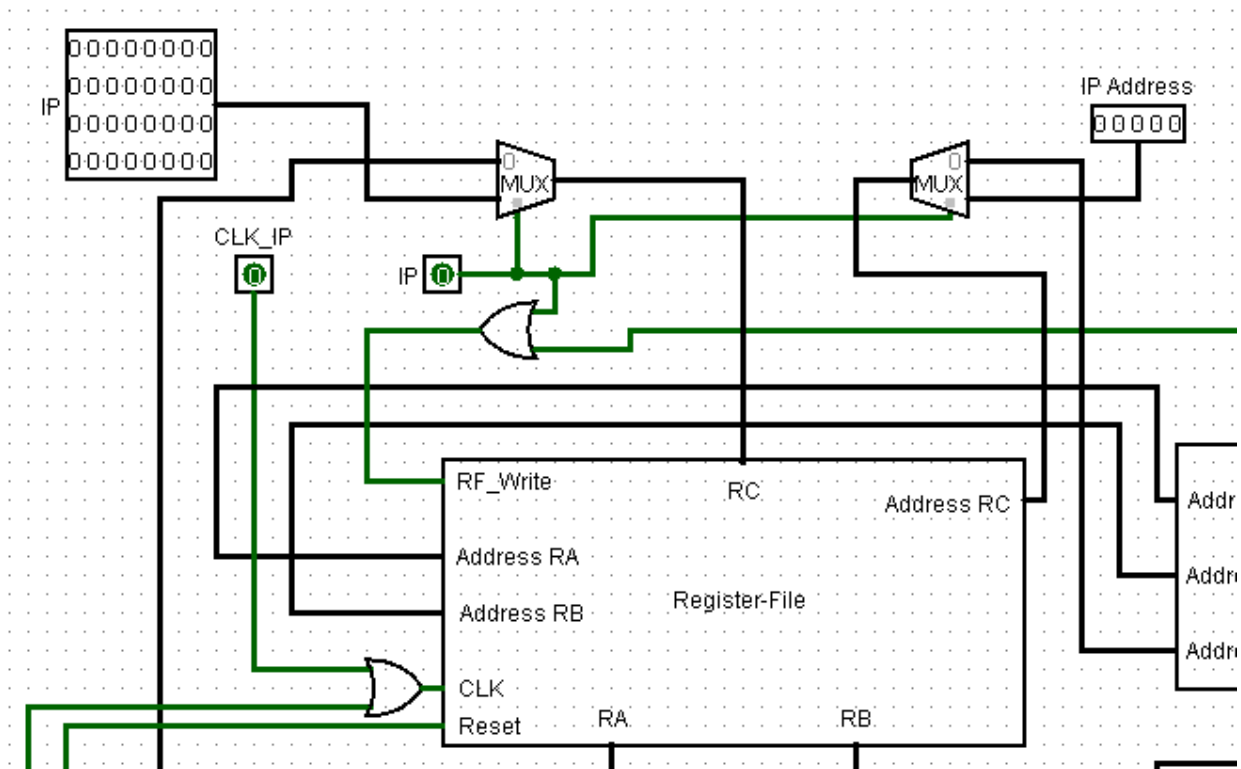


# SISD - Manual

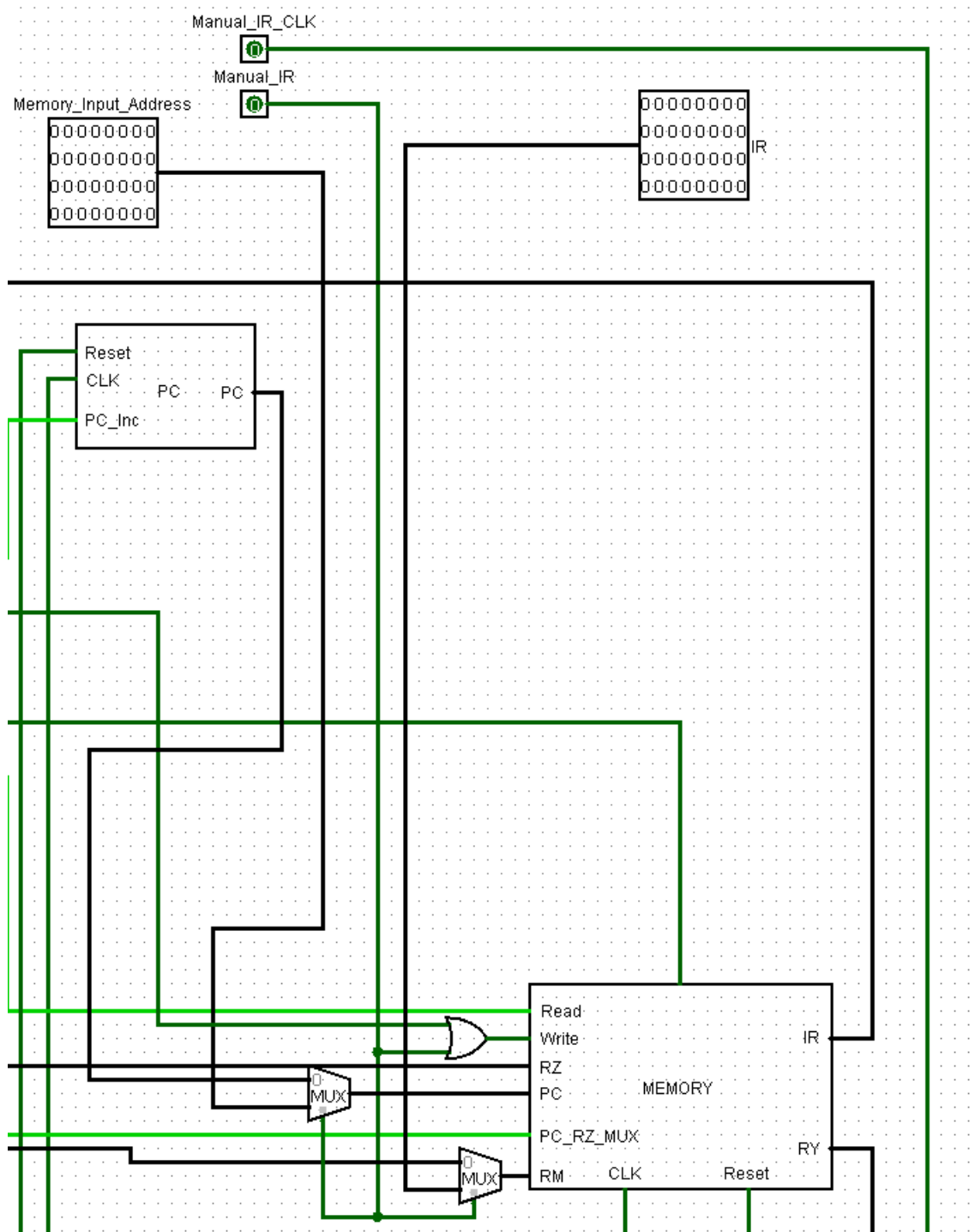
## Steps for writing Data in the Register-File?



An additional set of hardware has been added to facilitate functionality to allow data to be directly input from the main circuit into the register file.

1. When adding data, you need to make two specific inputs.
  - **Data/Value** – Must be entered as a 32-bit value in the IP (external input) input fields highlighted below.
  - **Register Address** – Must be entered as a 5-bit value in the IP Address section shown below and select the register to enter from the 32 available registers.
2. After these two inputs have been successfully entered, these values can be used to write data to the register file.
3. To do this, we need to mimic the control signals from the CU to bypass the data manipulation lock. This is achieved by keeping the **IP button** in the "ON" state. Obviously, the **IP** also acts as a MUX select control signal to select between automatically generated circuit-based 0-state writes and user-based manual 1-state inputs.
4. Now that the **IP button** is turned on and both the data and address values are in the correct locations, all we need to do is turn on the clock ripple with a single button to start updating the register file pointer values. register.

## Steps for enter Instructions in the Circuit?

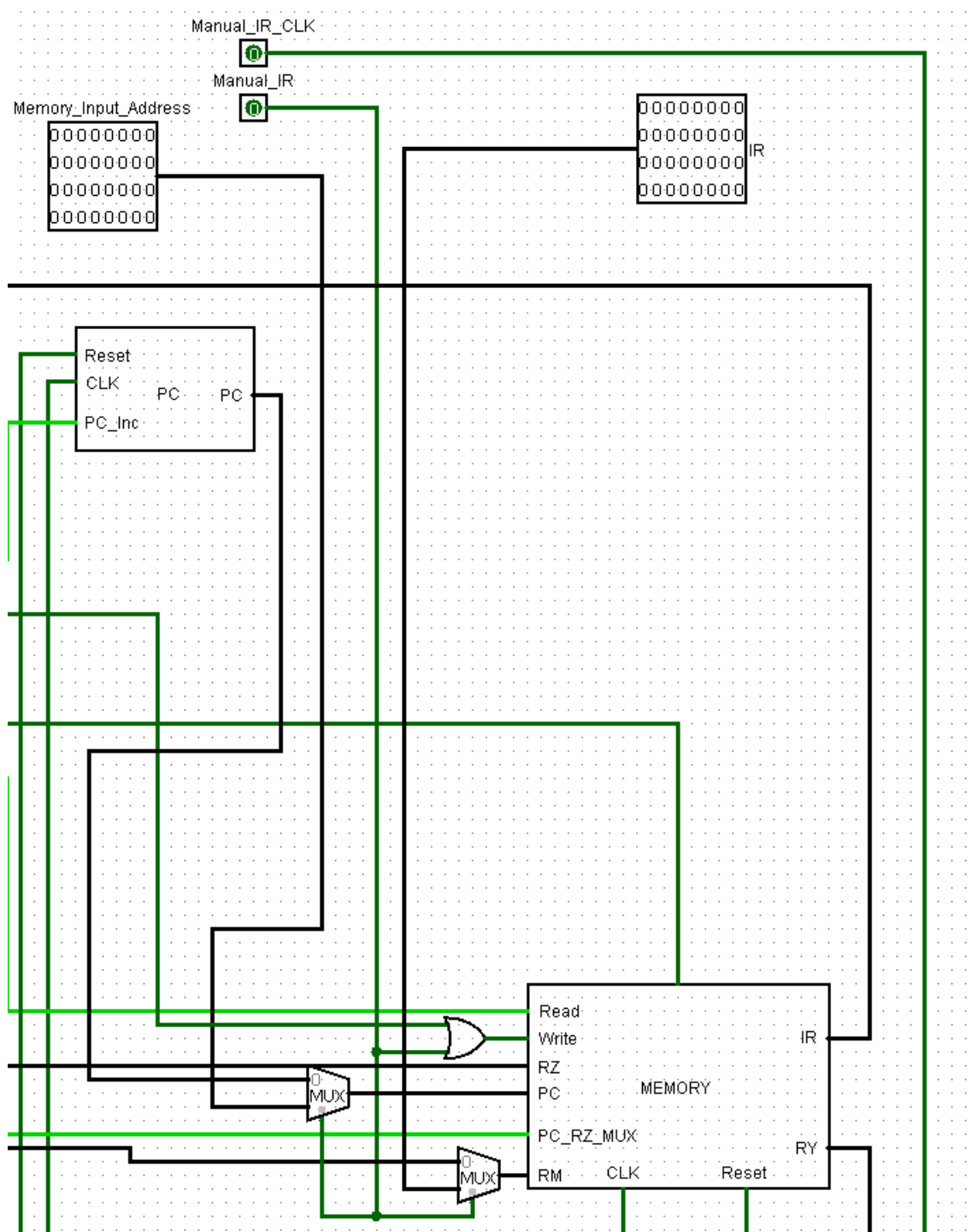


1. The Instructions that need to be executed in-circuit are stored in **Memory in 32-bit form**. The structure is described in the instruction splitting documentation. See the table below for quick reference.

Op. No.	Operation	31-26	25-21	20-16	15-11	10-0
0	MOV R1, R2	x0xxxx	R1	R2	x	x
1	MOV R1, Imm	x0xxxx	R1	x	Imm	Imm
2	LOAD R1, X(R2)	x0xxxx	R1	R2	Imm	Imm
3	STORE R1, X(R2)	x0xxxx	x	R2	R1	Imm
4	ADD R1, R2, R3	x0xxxx	R1	R2	R3	x
5	ADI R1, R2, Imm	x0xxxx	R1	R2	Imm	Imm
6	SUB R1, R2, R3	x0xxxx	R1	R2	R3	x
7	SUI R1, R2, Imm	x0xxxx	R1	R2	Imm	Imm
8	AND R1, R2, R3	x0xxxx	R1	R2	R3	x
9	ANI R1, R2, Imm	x0xxxx	R1	R2	Imm	Imm
10	OR R1, R2, R3	x0xxxx	R1	R2	R3	x
11	ORI R1, R2, Imm	x0xxxx	R1	R2	Imm	Imm
12	HLT	x0xxxx	x	x	x	x

2. Similar to register files, memory files are configured with an additional memory write component. The functionality of these components is exactly the same as the RF input.
3. When adding data, just like the Register File, you need to make two specific inputs.
  - **Data/value** – Must be entered as a 32-bit value in the **IR (external input)** input box highlighted below.
  - **Register Address** – Must be entered as a 5-bit value in the **Memory\_input\_Address** section shown below and select a register to populate from the 32 available registers.
4. After these two inputs are successfully entered, these values can be used to write data to the memory file with the **Manual\_IR\_MUX** button.
5. To do this, we need to mimic the control signals from the CU to bypass the data manipulation lock. This is accomplished by keeping his **Manual\_IR\_MUX** button in the "ON" state. Apparently, the **Manual\_IR\_MUX** also acts as a MUX select control signal, selecting inputs from automatically generated circuit-based writes in the '0' state and user-based manual inputs in the '1' state. increase.
6. Now that the **Manual\_IR\_MUX** button has been turned on and both the data and address values are in the correct locations, simply turn on the one-button clock ripple to start updating the values in memory. Start of file pointer register.

## Steps to enter Data in the Memory?



The data storage process is the same as the instruction feed mechanism.

1. Similar to register files and instruction lines, memory files are configured with additional memory write components. The functionality of these components is exactly the same as the RF input. Inserting data uses common changes at the command prompt.

2. When adding data, just like registering a file, two specific inputs must be made.
  - **Data/value** – Must be entered as a 32-bit value in the **IR** (external input) input box highlighted below.
  - **Register Address** – Must be entered as a 5-bit value in the **Memory\_Input\_Address** section shown below and select a register to populate from the 32 available registers.
3. After these two inputs have been successfully entered, these values can be used to write data to the memory file using the **Manual IR MUX** button.
4. To do this, we need to mimic the control signals from the CU to bypass the data manipulation lock. This is accomplished by keeping the **Manual IR MUX** button in the "on" state. Apparently, the **Manual IR MUX** button also acts as a MUX select control signal, selecting inputs from automatically generated circuit-based writes in the '0' state and user-based manual inputs in the '1' state. increase.
5. Now that the **Manual IR MUX** button is turned on and both the data and address values are in the correct positions, simply turn on the one-button clock ripple to start updating the values. Memory file pointer register to start from.