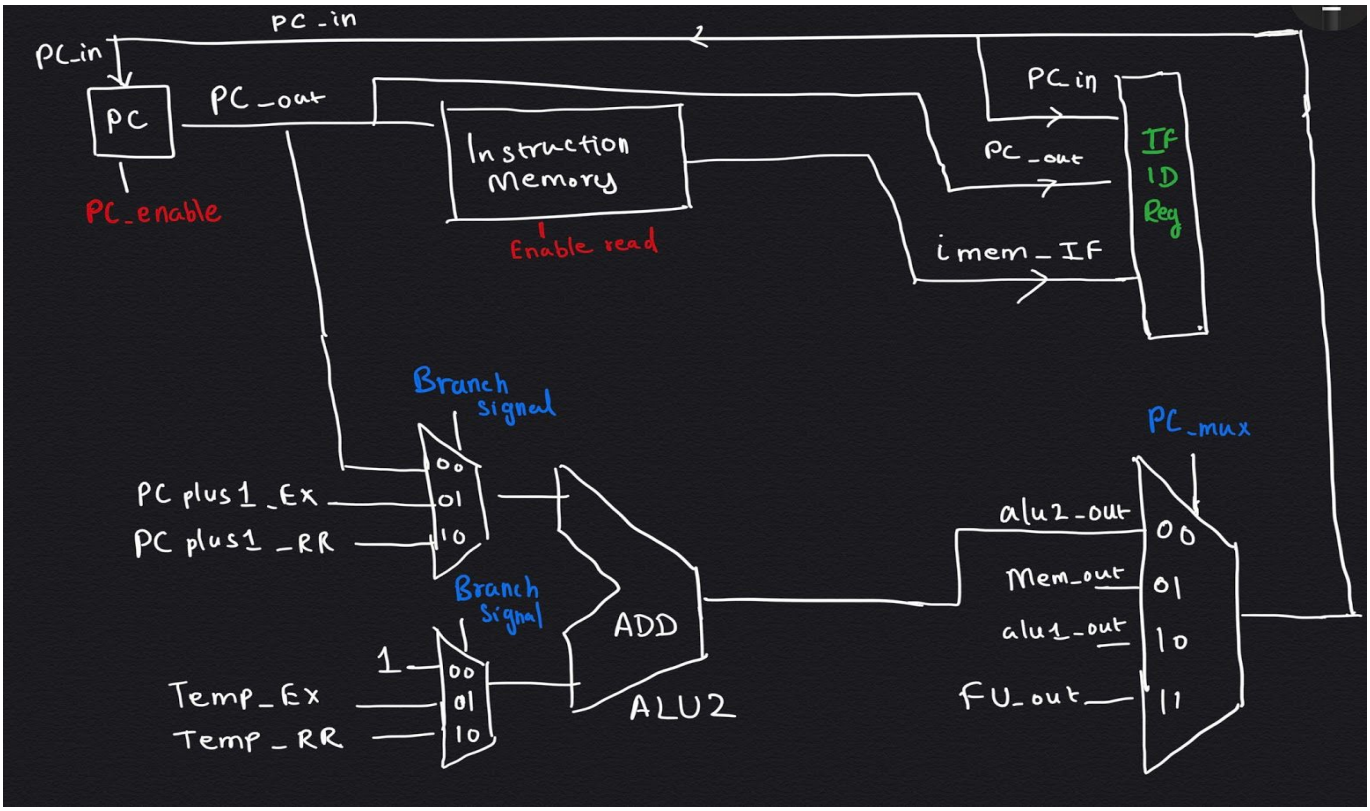
Six Stage Pipelined Design Implementation

Design Report

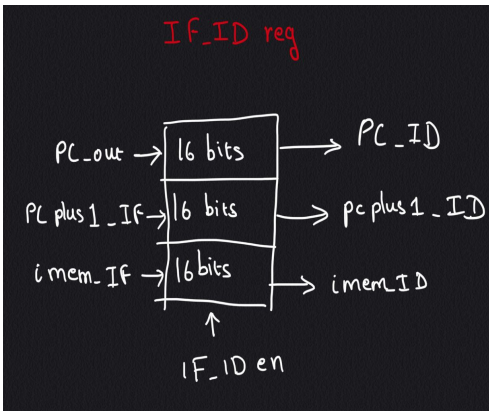
Divycharaneeswar R P 170260018

Thoutam Shiva Teja 170260035

Instruction Fetch Stage

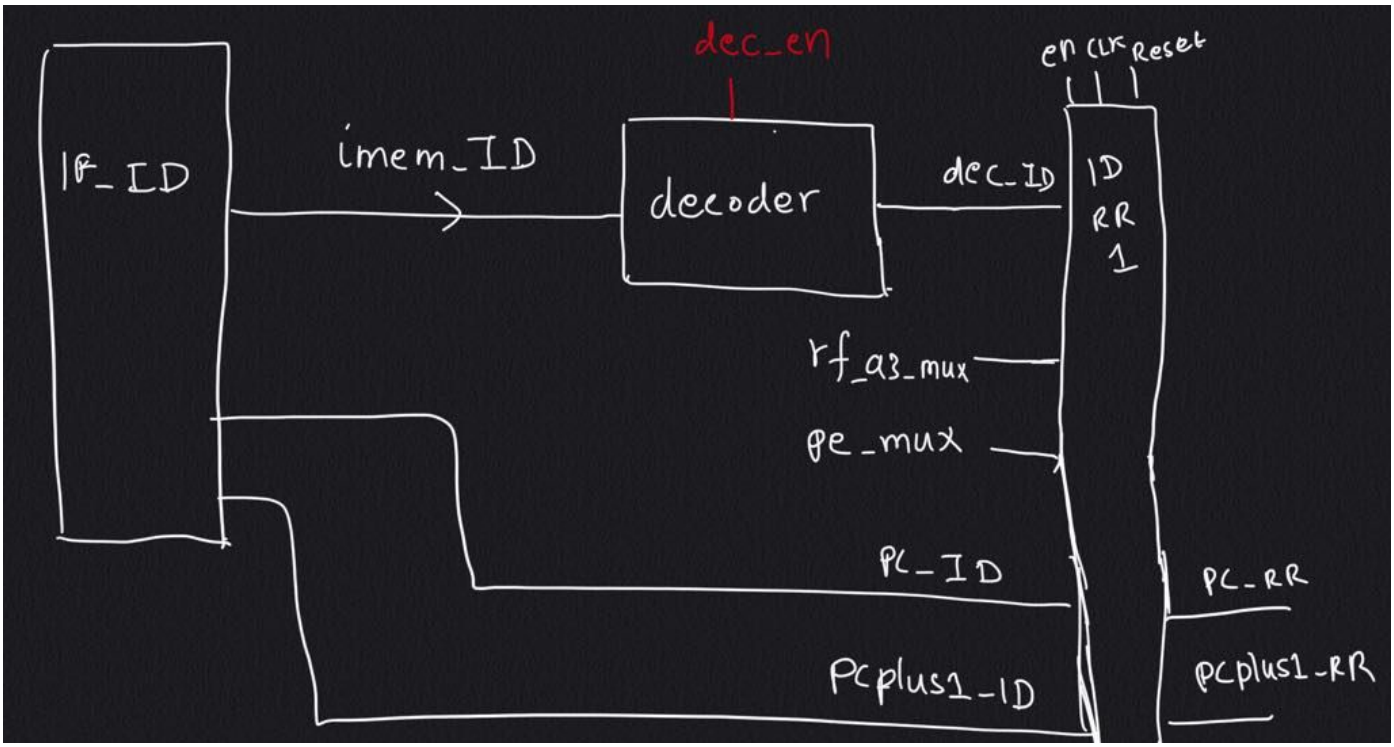


1. **PC register**: Stores the PC value of the current instruction.
2. **Instruction memory** : Instruction Memory from which the instruction is obtained.
3. **ALU2** : Used to update the PC value(PC+1 / PC+Imm)
4. **Branch Signal**
    a. 01 if BEQ , we get to know know in Execution stage
        Temp_EX -- Immediate value to be added
    b. 10 if JAL , we designed such that Imm is available from RR stage
        Temp_RR -- Immediate value to be added
    c. 00 else
5. **Enable read =1 always**
6. **PC_enable = 1 if no stalling in IF stage**
7. **PC_MUX** **--**Decides the PC value when the destination register is R7(Considered that on writing in R7 register also changes the PC)
    a. 10 if ADD/ADI/NDU and If the destination register is R7
    b. 01 if LW/LM and If the destination register is R7
    c. 11 if LHI/JLR and If the destination register is R7
    d. 00 else



- PCplus1_IF is connected with input of PC
- imem_IF is connected with output of the instruction memory
- IF_ID **= 1 if no stalling in IF stage**

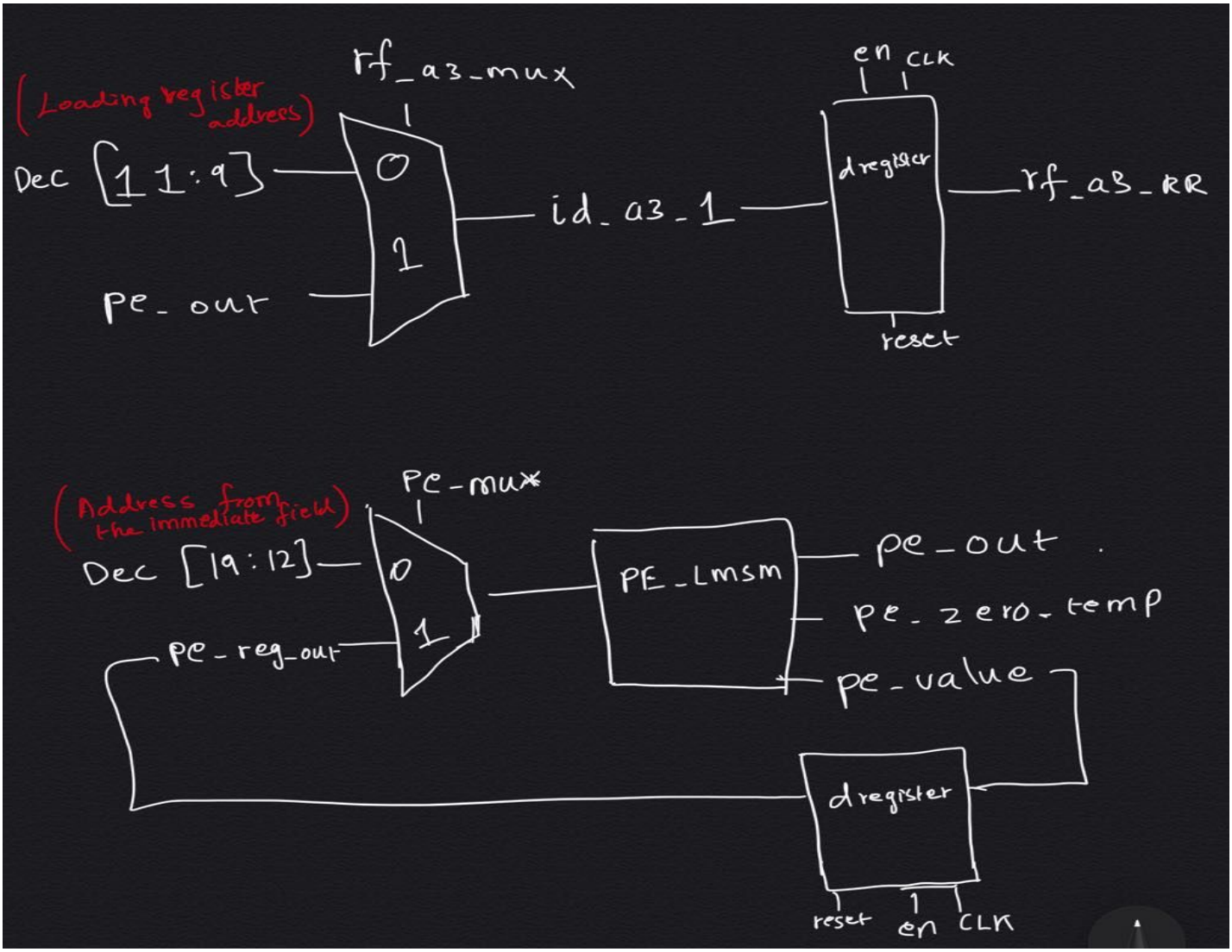
Instruction Decoder



In ID stage, the instruction is passed to the decoder from the IF_ID register.Two pipeline registers ID_RR1 and ID_RR2 are used ,where ID_RR1 takes care of the LM/SM instructions.

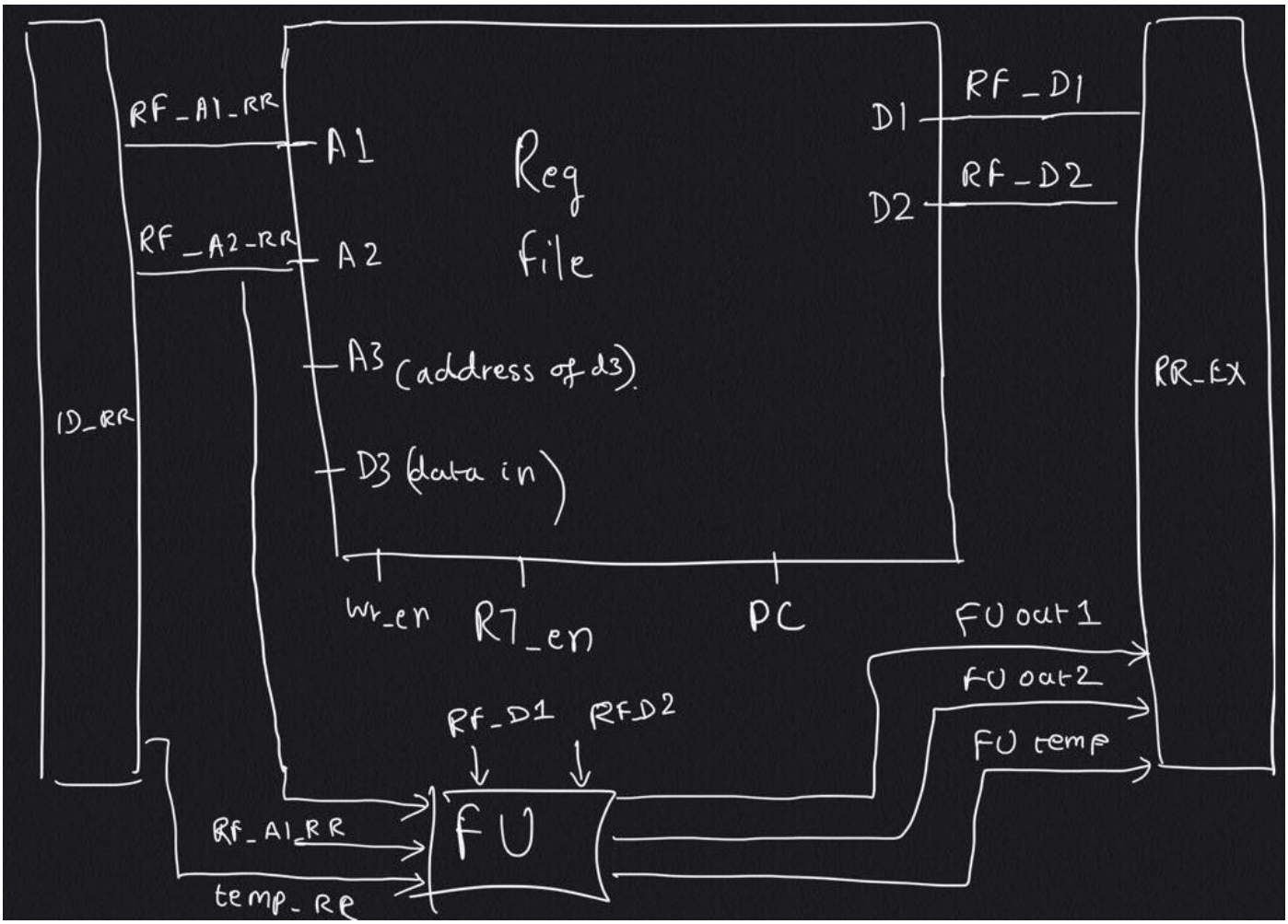
1. **Decoder :**
    a. Takes the input as the instruction and gives output in 45 bits which includes
    b. If **decoder_enable is set**, then **mem_wr_en , c flag_en , z flag_en are set.**

Inside ID_RR



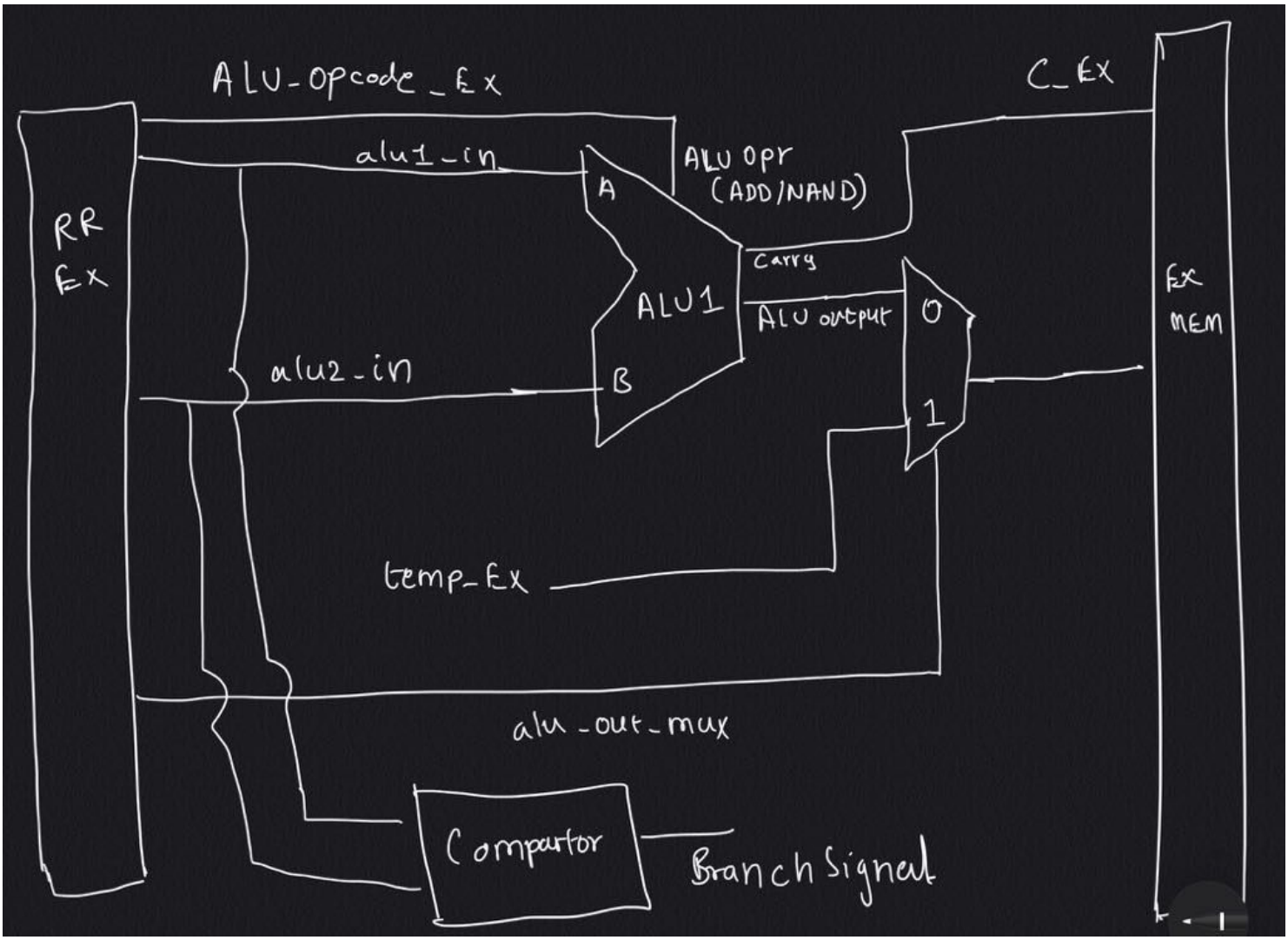
2. **Rf_a3_mux : It is 1 when the instruction is LM** and this mux gives the address of the register to loaded.
3. **PE_lmsm :**
    a. This block takes the address from the immediate field and gives output as the address of the register(pe_out) to be used in LM / SM instruction.
    b. It also updates the address from the immediate field of already used register with 0 (pe_value)and when all 8 bits are zero in the address , the pc_zero is set as 1.
4. **PE_mux :**
    a. This chooses between the address from the immediate field or the updated address from the PE_lmsm block

Register Read



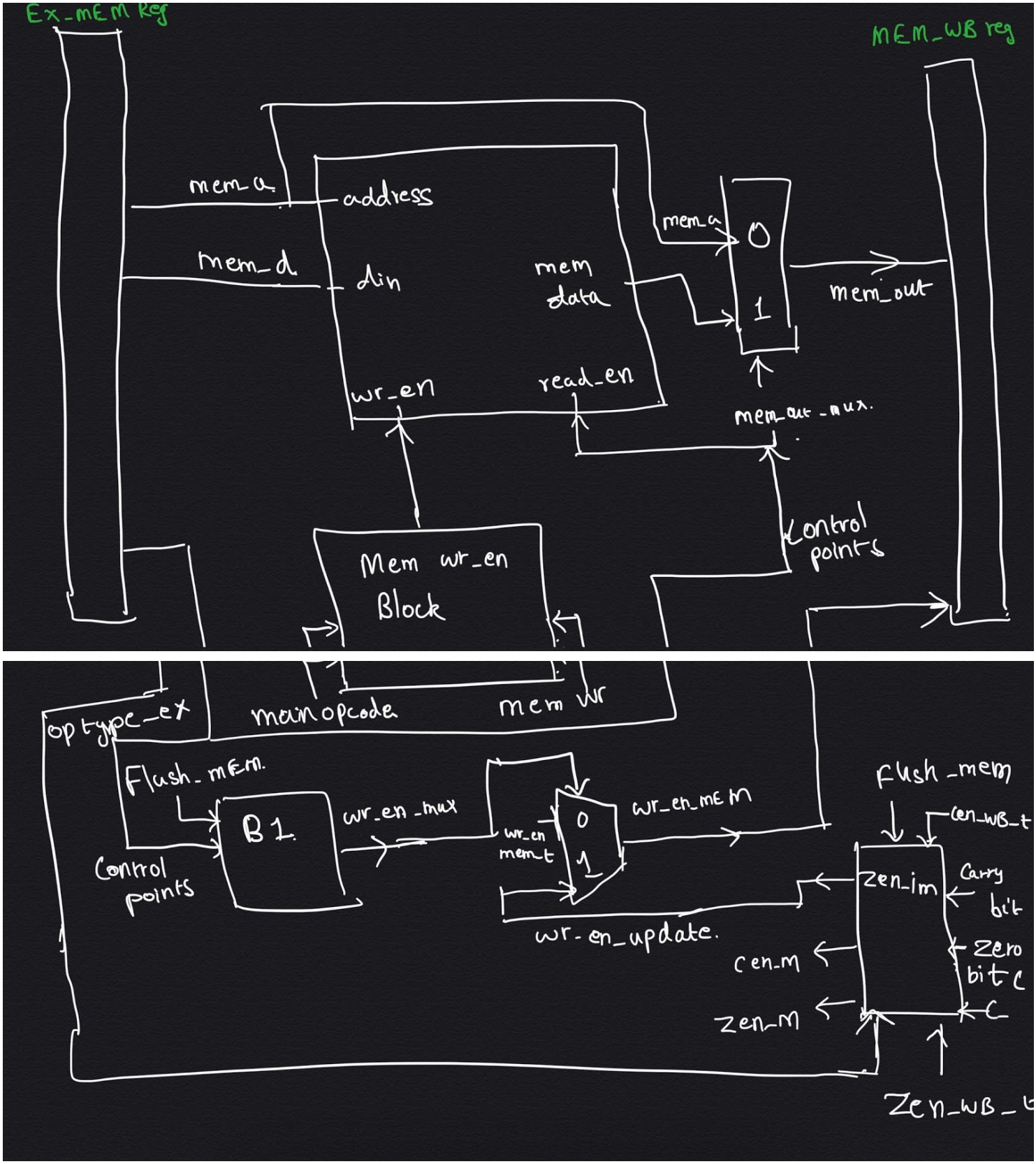
1. This stage has the forwarding unit and the register file
2. FU out1 goes for alu _in1
3. FU out2 goes for alu_in2
4. FU_temp is data in for memory

Execution



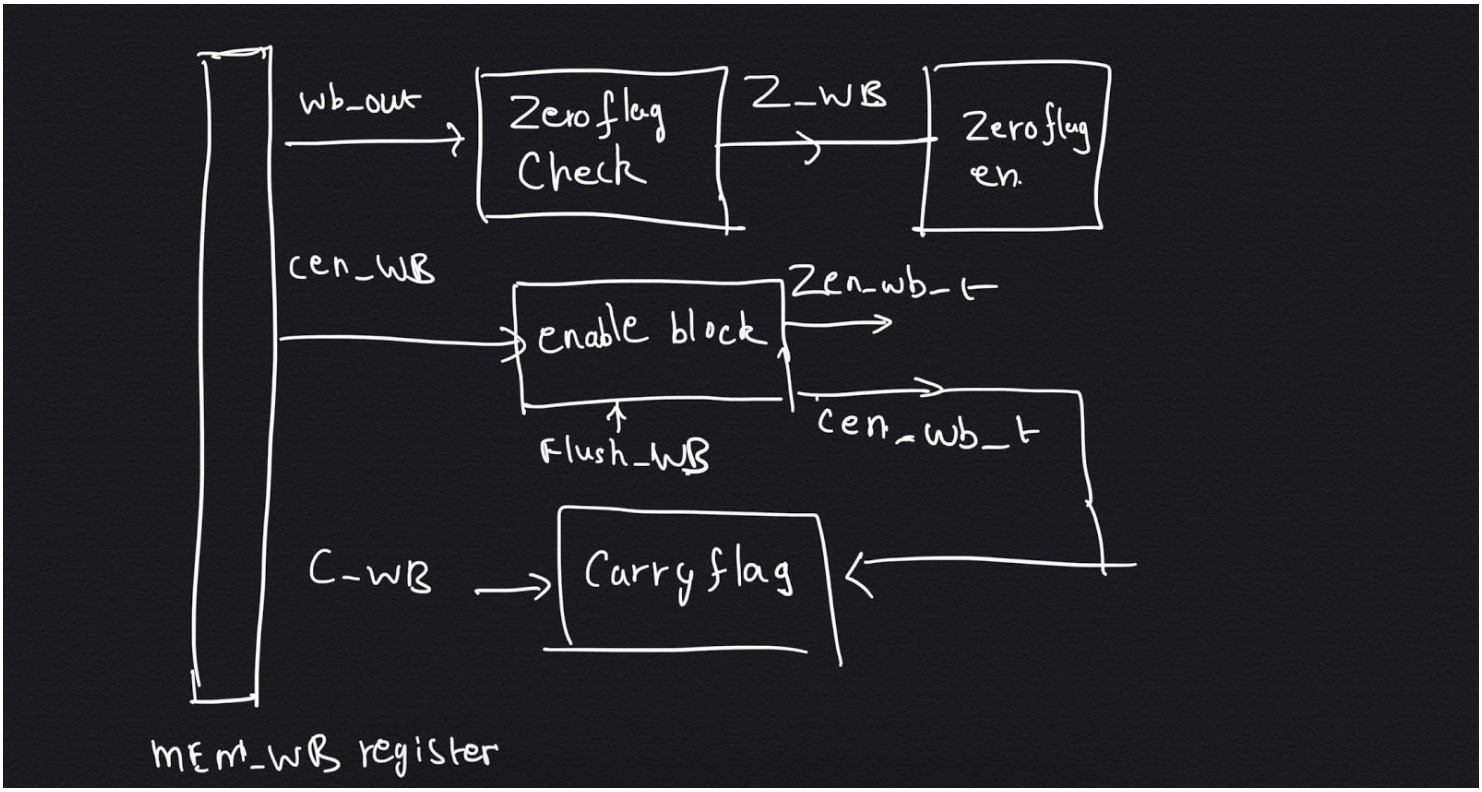
1. **ALU** : Perform ADD, NAND operation
2. **Comparator :** Checks the equality and generates branch signal for choosing the updated PC in IF stage
3. **Temp EX :** Has the sign extended value or the content of the register depending on the instruction
4. **ALU_out_mux:** decided from the 3rd bit of controlpoint

Memory Write/Read



1. **MEM** : Data memory from which the instruction read data or writes data.
2. **MEM wr en block**:
    a. During the store instruction if the memory stage is needed to flushed then wr_en is made 0
    b. In other cases the wr_en signal from the decoder is taken
3. The wr_en_update corresponds to the wr_en for ADC,ADZ,NDC,NDZ instructions

WriteBack



1. **Enable block:**
    a. If flush is active for WB stage then it doesn't write enable the zero and carry flag.
    b. If there is no flush then write enable signals for carry and zero flag received from decoder is taken .
2. **Zero flag check:**
    a. WB_out gets the memory output in load case and in other cases gets the alu1 output.
    b. This block checks if WB_out is zero
3. C_WB comes from the carry of the alu1

Control points:

1. **Wr_en mux ⇒ 0**
2. **Rf_a3_mux ⇒ 1**
3. **Rf_a2_mux ⇒ 2**
4. **alu_out _mux ⇒ 3**
5. **Mem_out_mux ⇒ 4**
6. **MEM_WB_en ⇒ 5**

Pipeline Register Contents

Pipeline Register	Components	Length(bits)
IFID	PC	16
	PCplus1	16
	instruction	16
IDRR	Alu opcode	1
	Op type(z/c)	2
	main_opcode	4
	Control points	6
	rf_a1	3
	rf_a2	3
	rf_a3	3
	pe_out	3
	sign extended values	16
	wr_en	1
	mem_wr_en	1
	carry_en	1
	zero_en	1
	pc	16
	pcplus1	16
RREX	Alu opcode	1
	Op type(z/c)	2
	main_opcode	4
	Control points	6
	Alu input 1	16
	Alu input 2	16
	Sign extended value/rf output	16
	PC	16
	mem_wr_en	1
	carry_en	1
	zero_en	1
EXMM	Op type(z/c)	2
	main_opcode	4
	Control points	6
	mem_wr_en	1

	zero_en	1
	Memory address	16
	Memory data in	16
	pc	16
MMWB	Main _opcode	4
	carry	1
	Control points	8
	Zero_en	1
	Mem output/mem address	16
	pc	16