**Six Stage Pipelined**

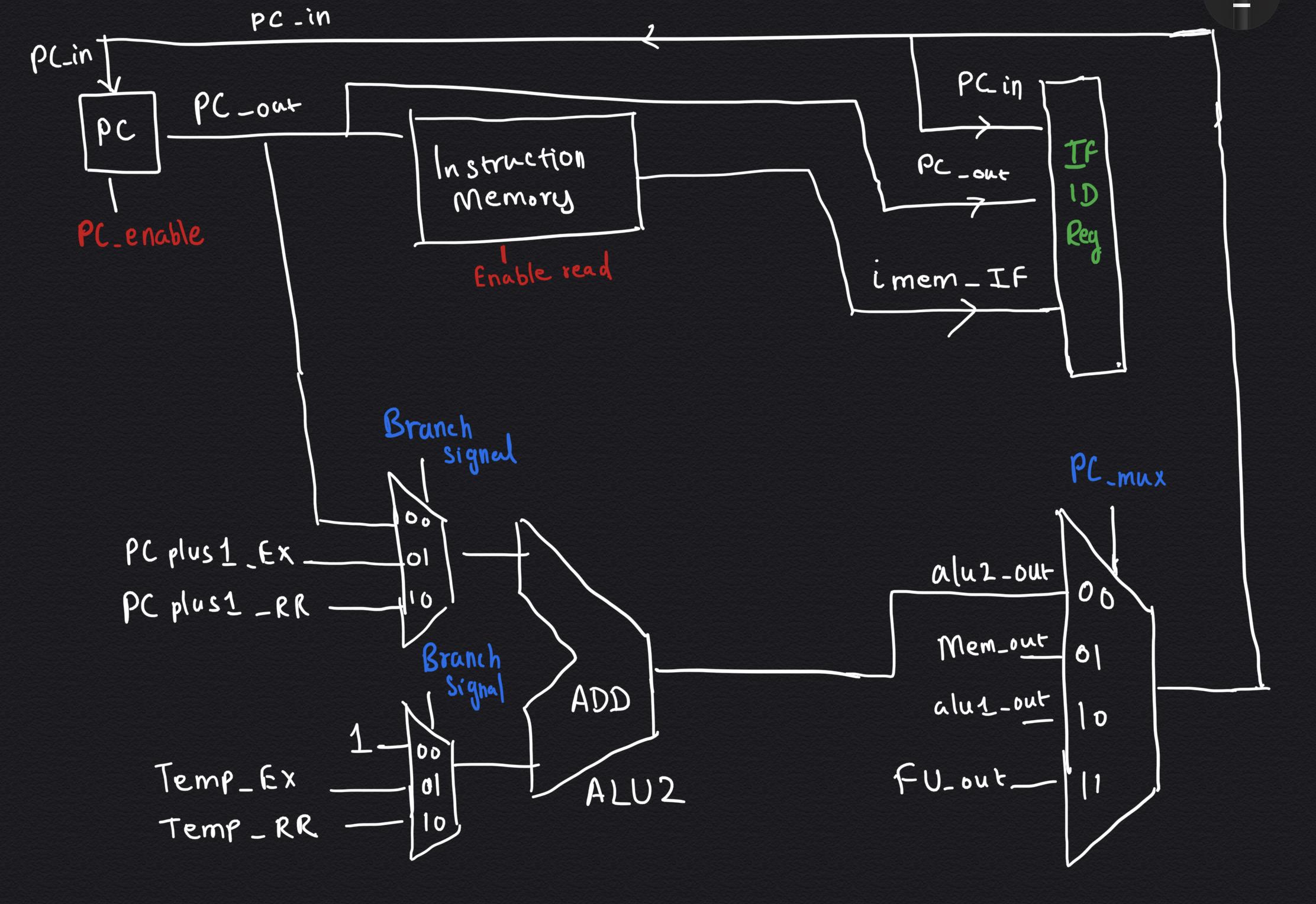
**Design Implementation**

**Design Report**

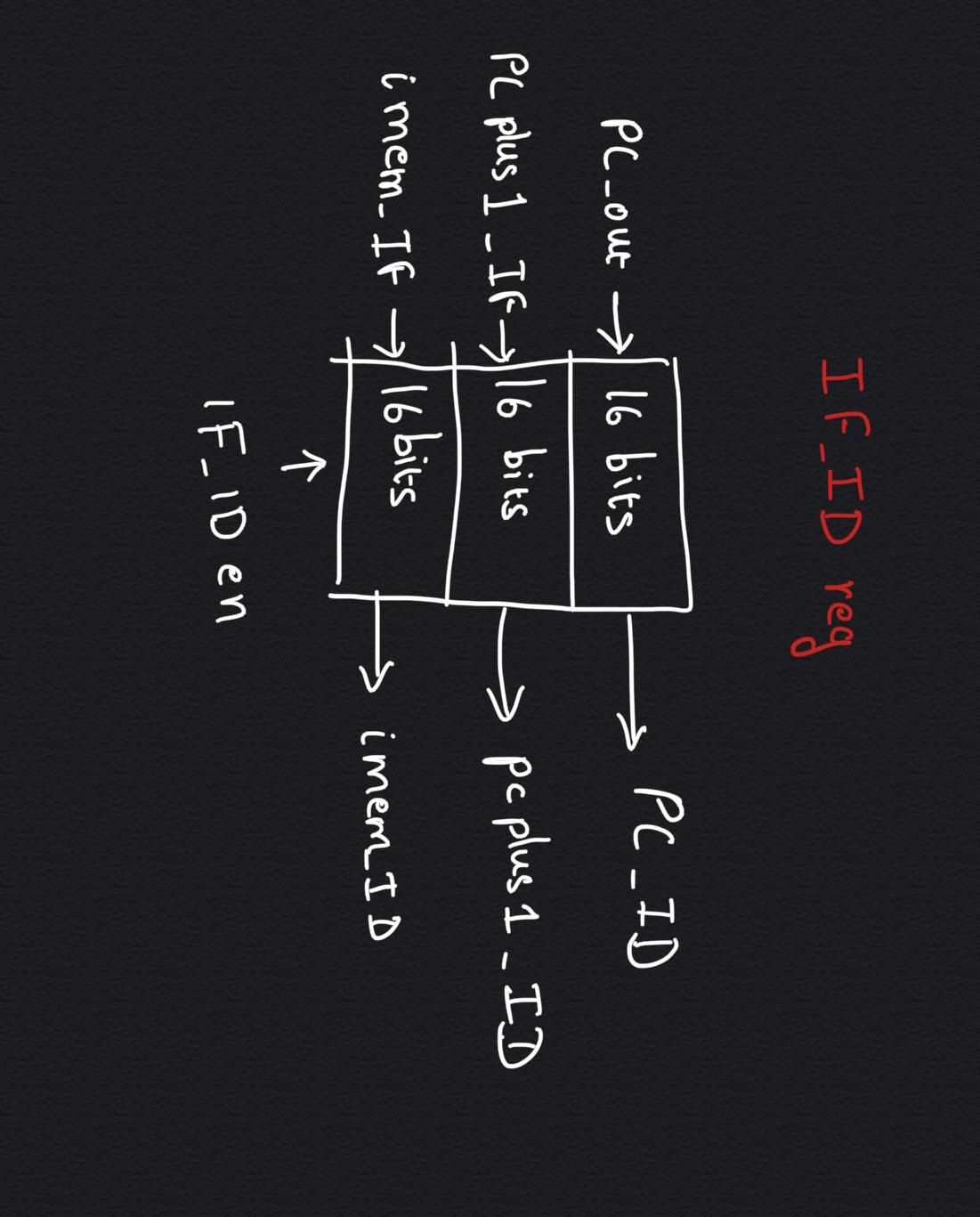
Divycharaneeswar R P Thoutam Shiva Teja

170260018 170260035

**Instruction Fetch Stage**

****

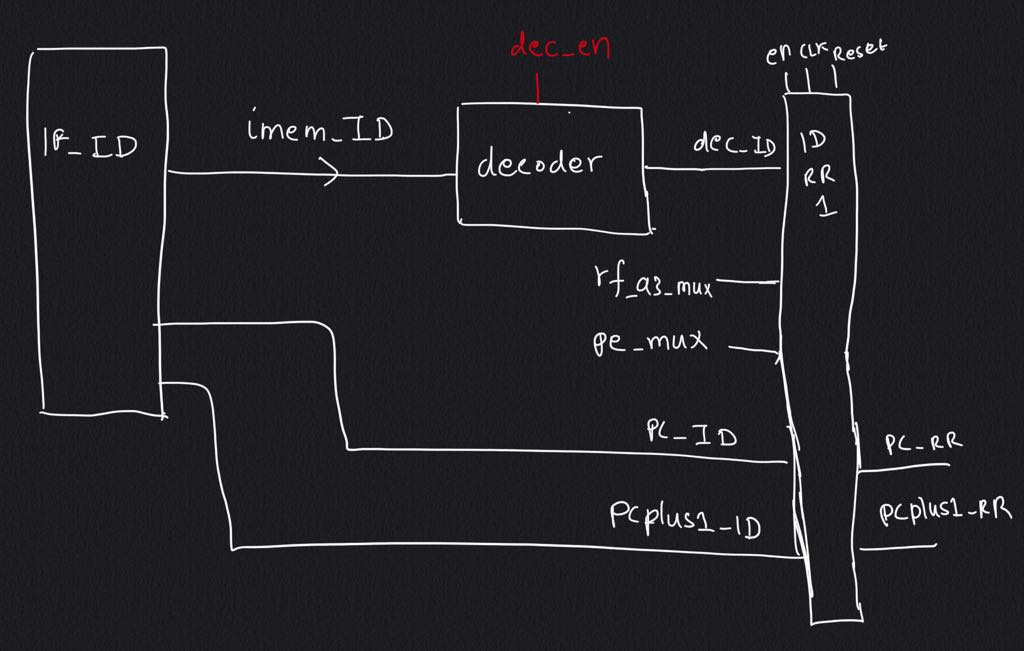
1. **PC register**: Stores the PC value of the current instruction.
2. **Instruction memory** : Instruction Memory from which the instruction is obtained.
3. **ALU2**  : Used to update the PC value(PC+1 / PC+Imm)
4. **Branch Signal**
   1. 01 if BEQ , we get to know know in Execution stage   
       Temp\_EX -- Immediate value to be added
   2. 10 if JAL , we designed such that Imm is available from RR stage   
       Temp\_RR -- Immediate value to be added
   3. 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 )
   1. 10 if ADD/ADI/NDU and If the destination register is R7
   2. 01 if LW/LM and If the destination register is R7
   3. 11 if LHI/JLR and If the destination register is R7
   4. 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**

**-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

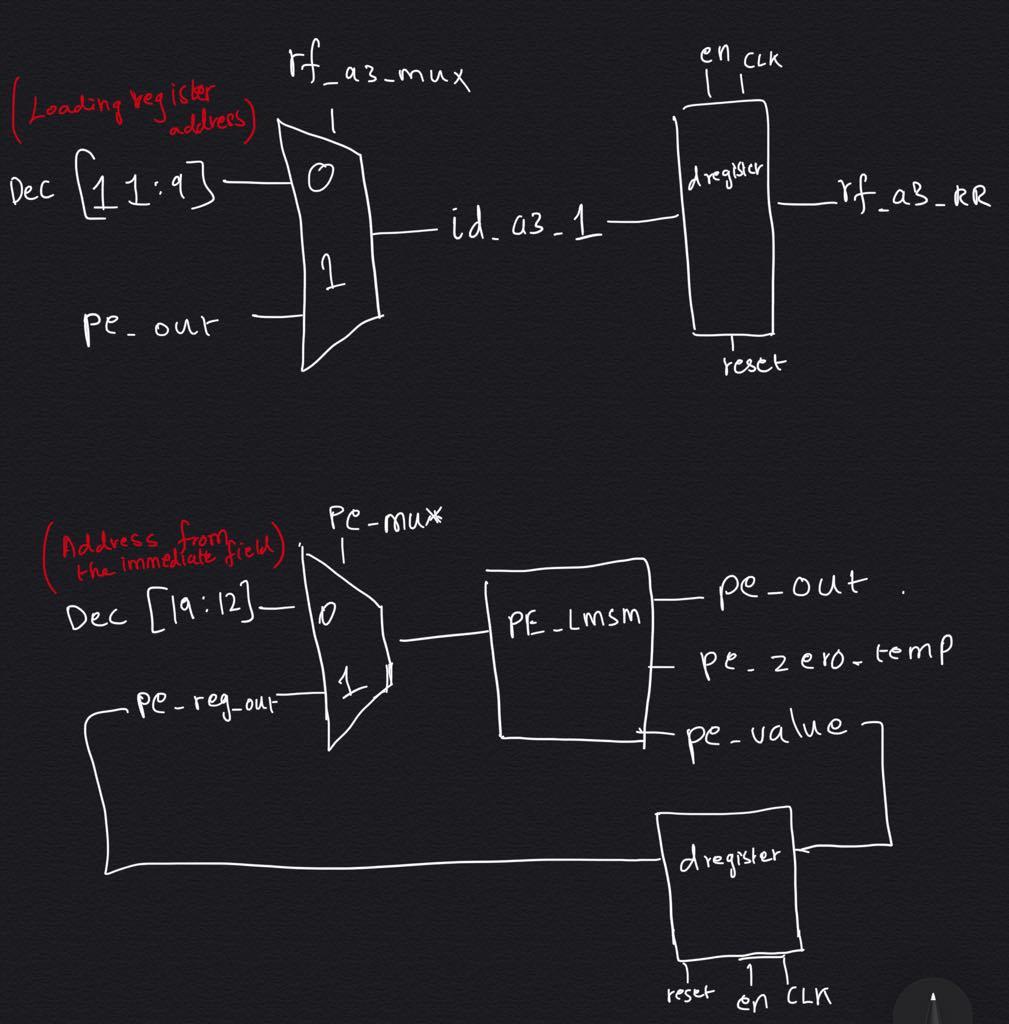
**Instruction Decoder**

****

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 :**
   1. Takes the input as the instruction and gives output in 45 bits which includes
   2. If **decoder\_enable is set**, then  **mem\_wr\_en , c flag\_en , z flag\_en are set.**

**Inside ID\_RR**

****

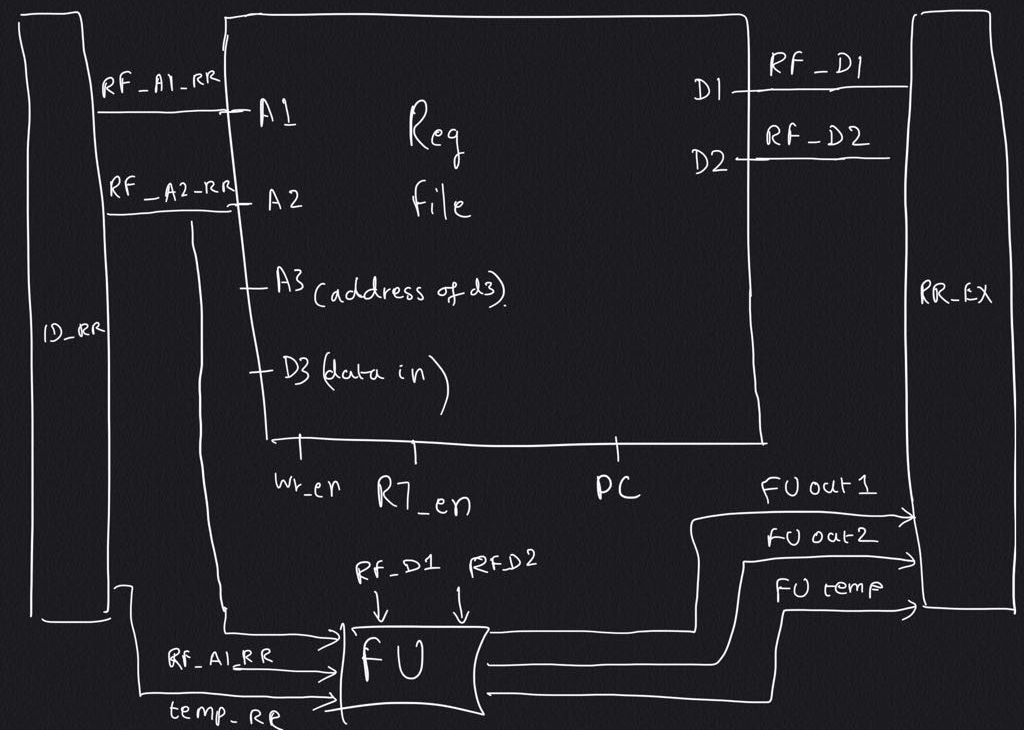
1. **Rf\_a3\_mux :** It is **1 when the instruction is LM** and this mux gives the address of the register to loaded.
2. **PE\_lmsm :** 
   1. This block takes the address from the immediate field and gives output as the address of the register(pe\_out) to be used in

LM / SM instruction.

* 1. 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.

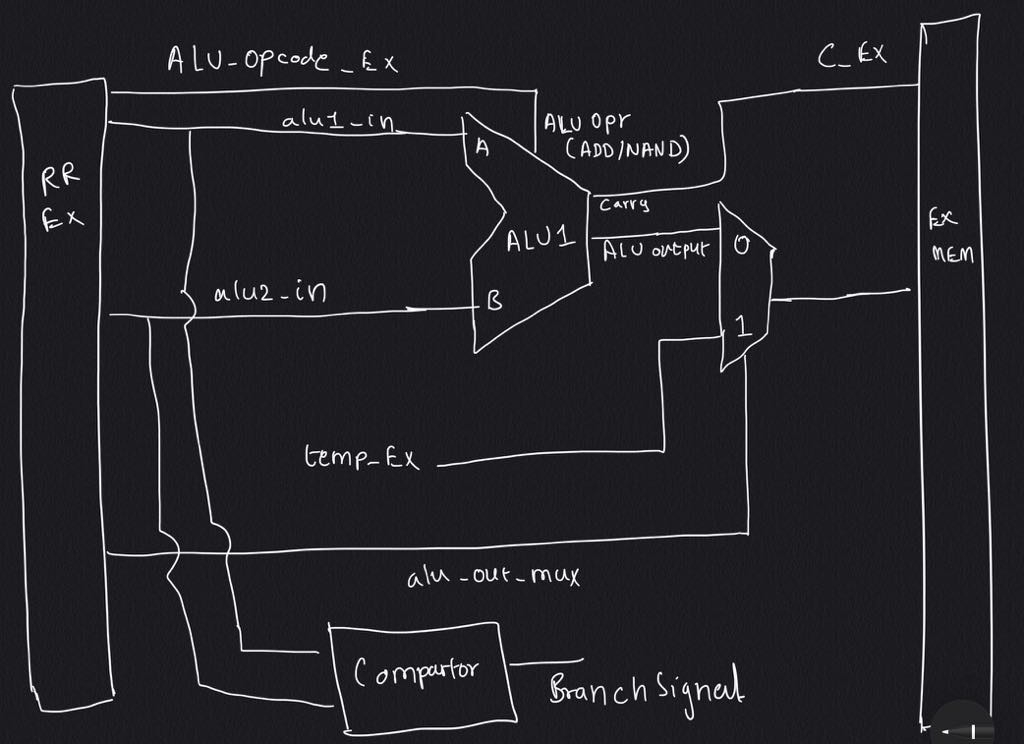
1. **PE\_mux :**
   1. This chooses between the address from the immediate field or the updated address from the PE\_lmsm block

**Register Read**

****

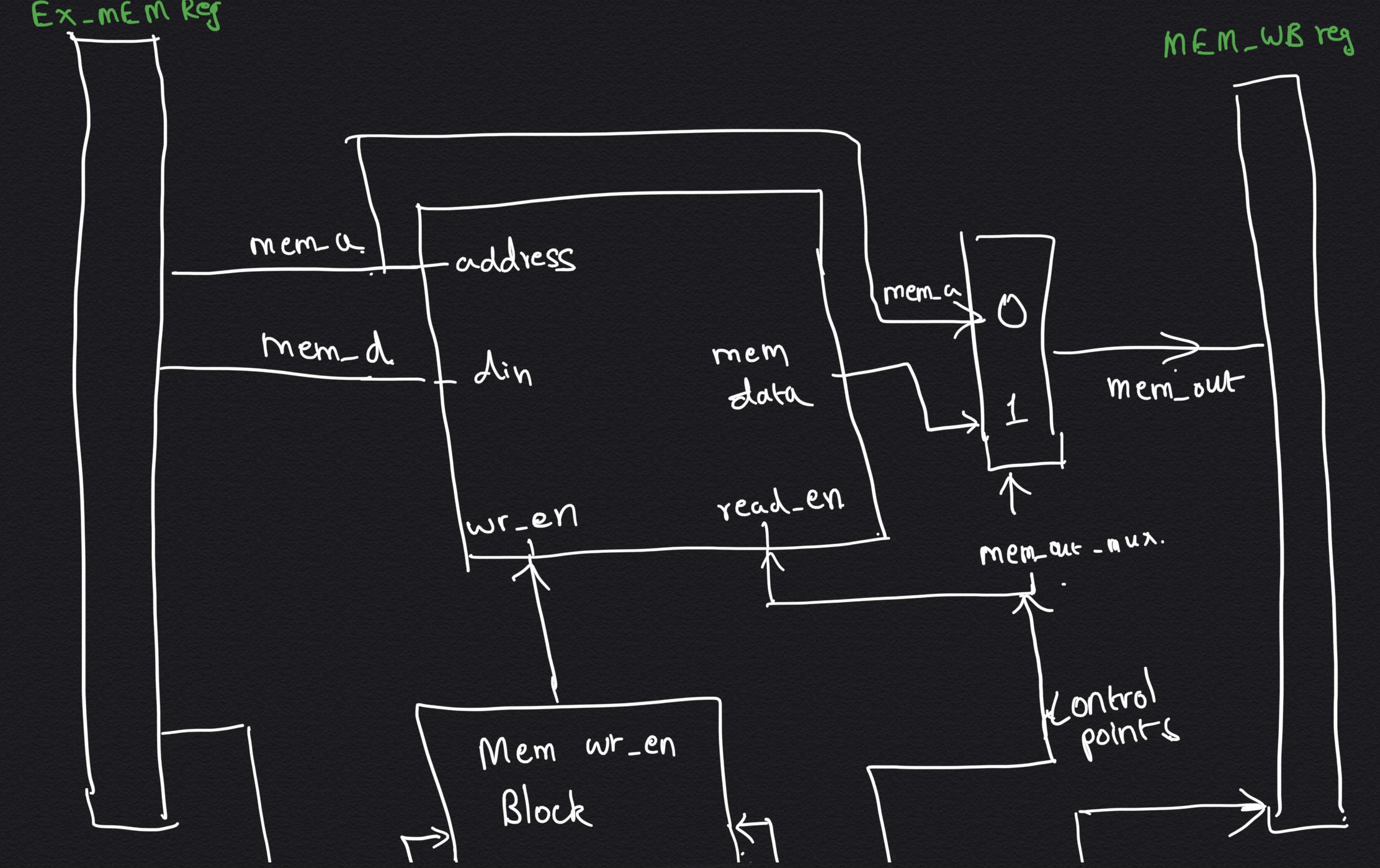
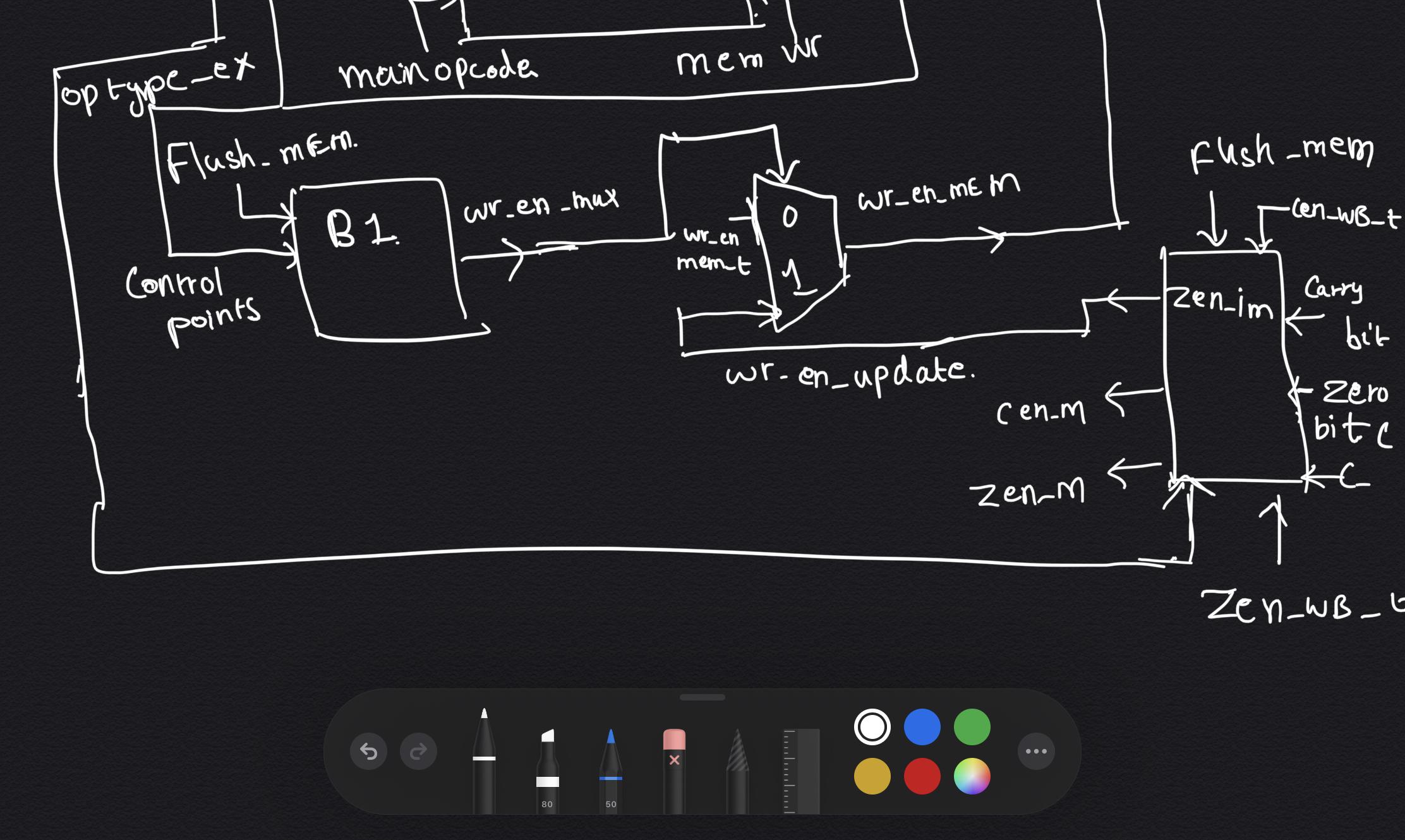
1. This stage has the forwarding unit and the register file
2. FU out1 goes for alu \_in1
3. FU out2 goes for alu\_in2
4. FU\_temp is data in for memory

**Execution**

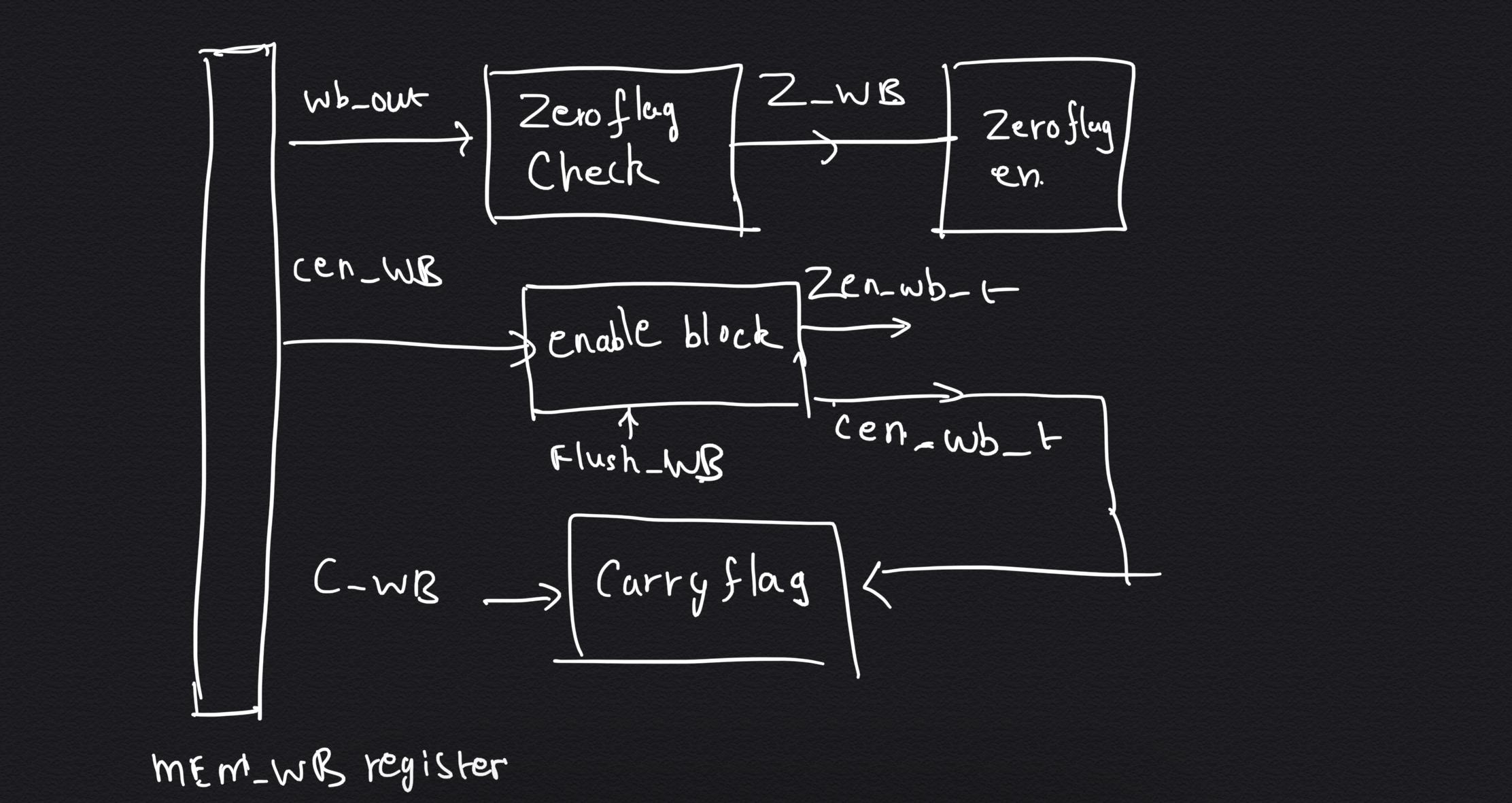


1. **ALU**  : Perform ADD, NAND operation
2. **Comparator :** Checks the equality and generates branch signal for choosing the updated PC in IF stage
3. **Temp EX :** Has the sign extended value or the content of the register depending on the instruction
4. **ALU\_out\_mux:** decided from the 3rd bit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

**Memory Write/Read**

1. **MEM** : Data memory from which the instruction read data or writes data.
2. **MEM wr en block**:
   1. During the store instruction if the memory stage is needed to flushed then wr\_en is made 0
   2. 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:**
   1. If flush is active for WB stage then it doesn’t write enable the zero and carry flag.
   2. If there is no flush then write enable signals for carry and zero flag received from decoder is taken .
2. **Zero flag check:**
   1. WB\_out gets the memory output in load case and in other cases gets the alu1 output.
   2. 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 |