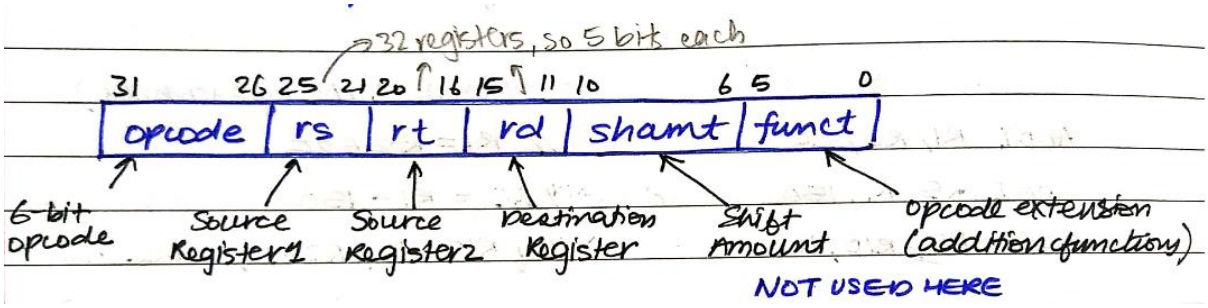
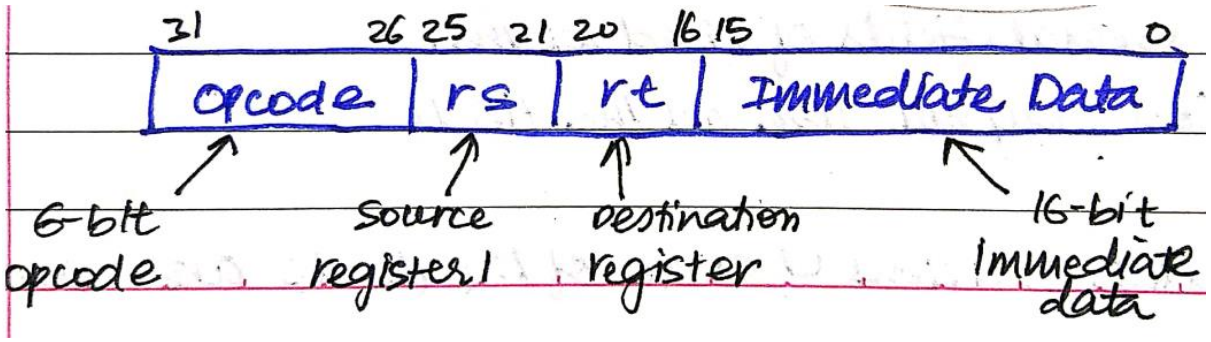
Reduced MIPS -32 pipelined architecture

(Verilog code in separate file)

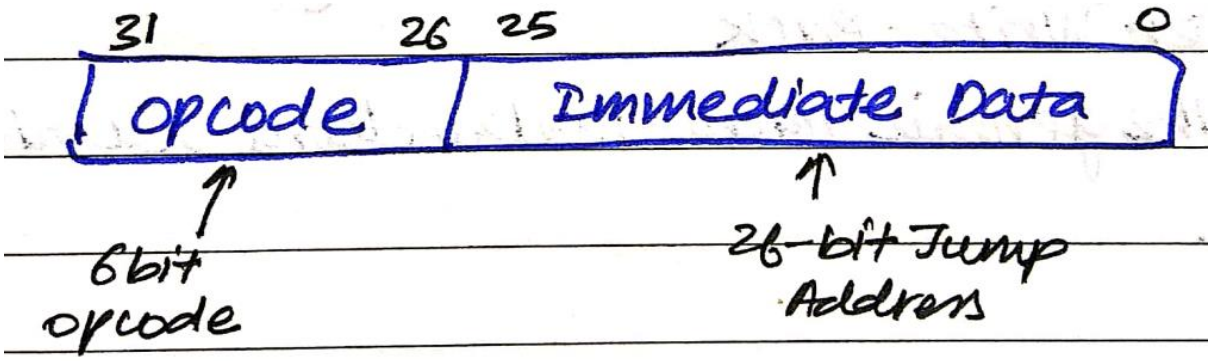
R-Type Instruction:



I – Type Instruction



J- Type Instruction



MIPS 32 Instruction cycle

We divide the instruction execution cycle into 5 steps :

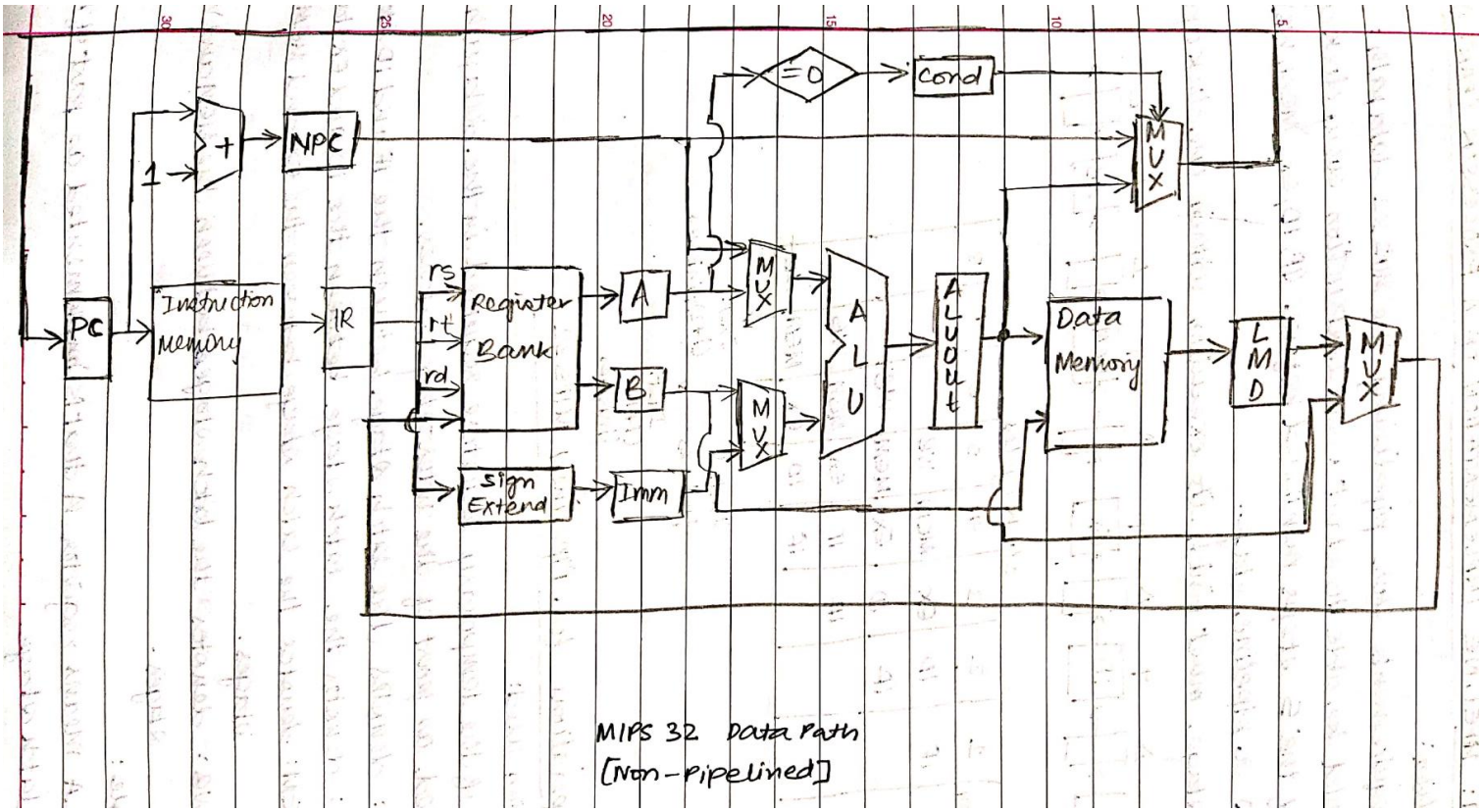
(a) IF : Instruction fetch

(b) ID : Instruction Decode / Register fetch

(c) EX : Execution / Effective Address Calculation

(d) MEM : Memory Access / Branch Completion

(e) WB : Register Write-Back

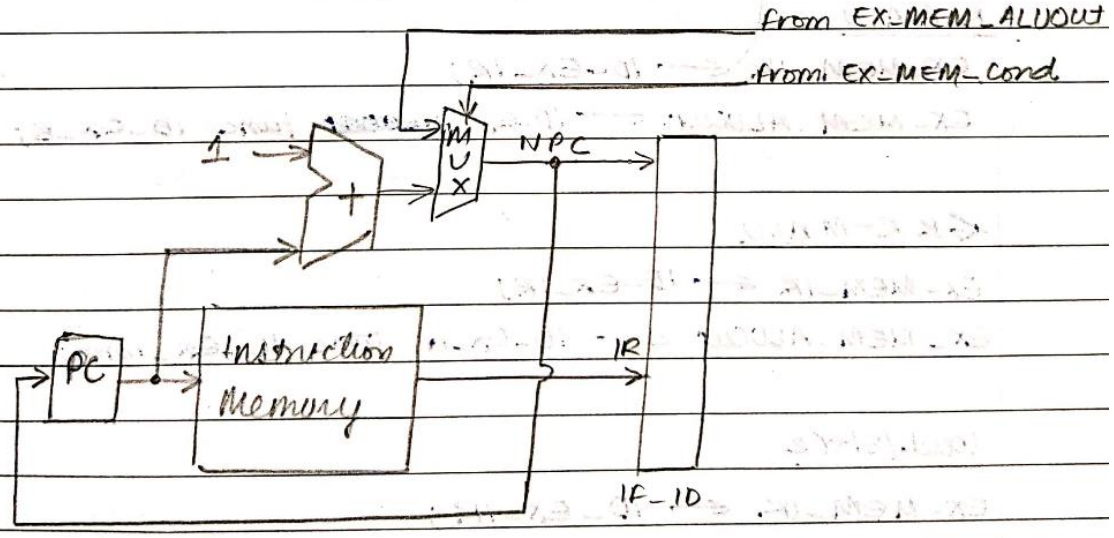


MIPS 32 Data Path
[Non-Pipelined]

(a) Micro-operations for Pipeline Stage IF

IF_ID_IR ← Mem [PC]

IF_ID_NPC, PC ← (if ((Ex_MEM_IR [opcode] == branch) &
Ex_MEM_cond.)
{EX_MEM_ALUOut}
else {PC+1});



(b) Micro-operations for Pipeline stage ID.

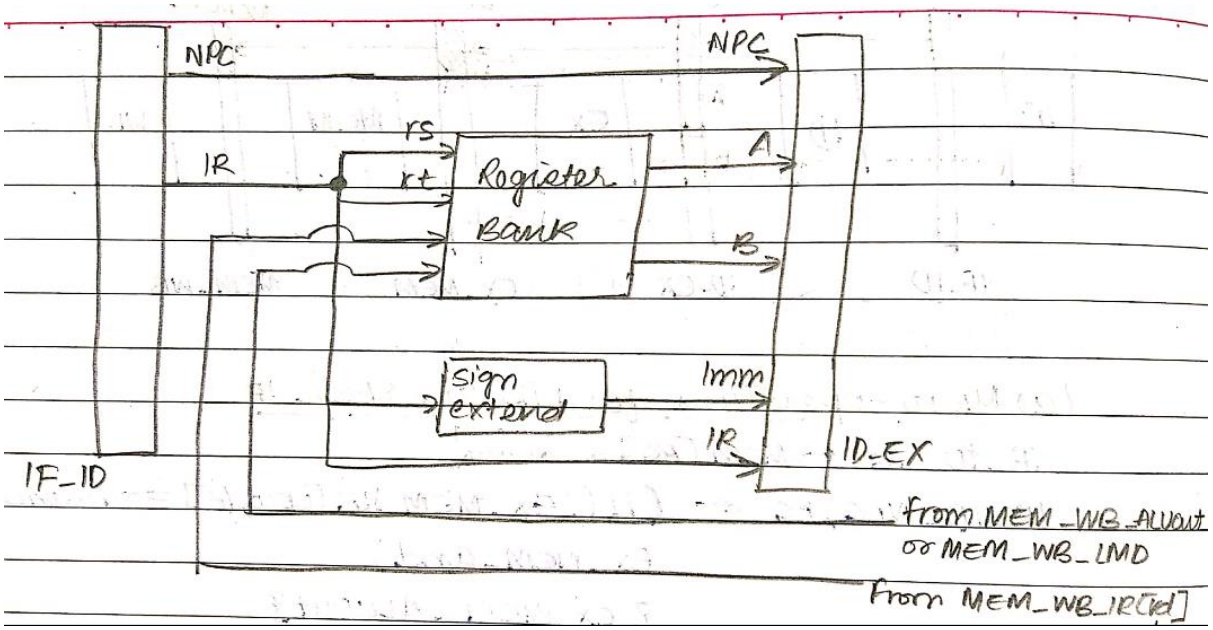
ID_EX_A ← Reg[IF_ID_IR[rs]];

ID_EX_B ← Reg[IF_ID_IR[rt]];

ID_EX_NPC ← IF_ID_NPC;

ID_EX_IR ← IF_ID_IR;

ID_EX_Imm ← Sign_extend(IF_ID_IR$_{15..0}$);



(b) Micro-operations for Pipeline stage ID.

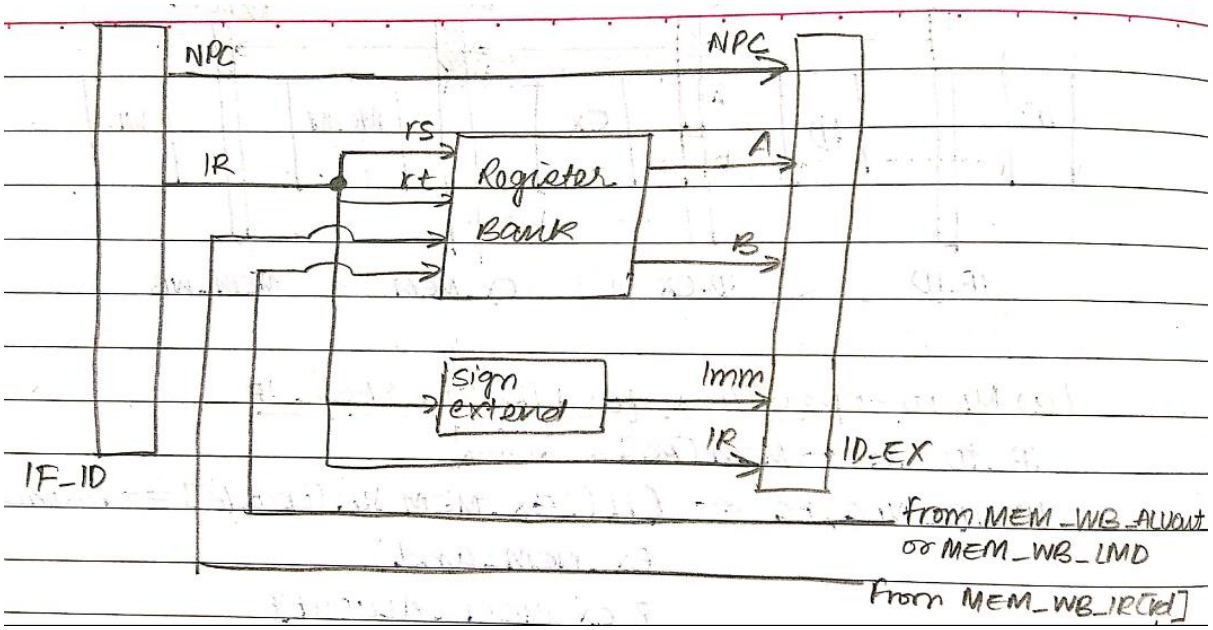
ID_EX_A ← Reg[IF_ID_IR[rs]];

ID_EX_B ← Reg[IF_ID_IR[rt]];

ID_EX_NPC ← IF_ID_NPC;

ID_EX_IR ← IF_ID_IR;

ID_EX_Imm ← Sign_extend(IF_ID_IR$_{15..0}$);



(c) Micro-Operations for Pipeline Stage EX.

R-R ALU

EX_MEM_IR ← ID_EX_IR;

EX_MEM_ALUOut ← ID_EX_A func ID_EX_B;

R-M ALU

EX_MEM_IR ← ID_EX_IR;

EX_MEM_ALUOut ← ID_EX_A func ID_EX_Imm;

Load/Store

EX_MEM_IR ← ID_EX_IR;

EX_MEM_ALUOut ← ID_EX_A + ID_EX_Imm;

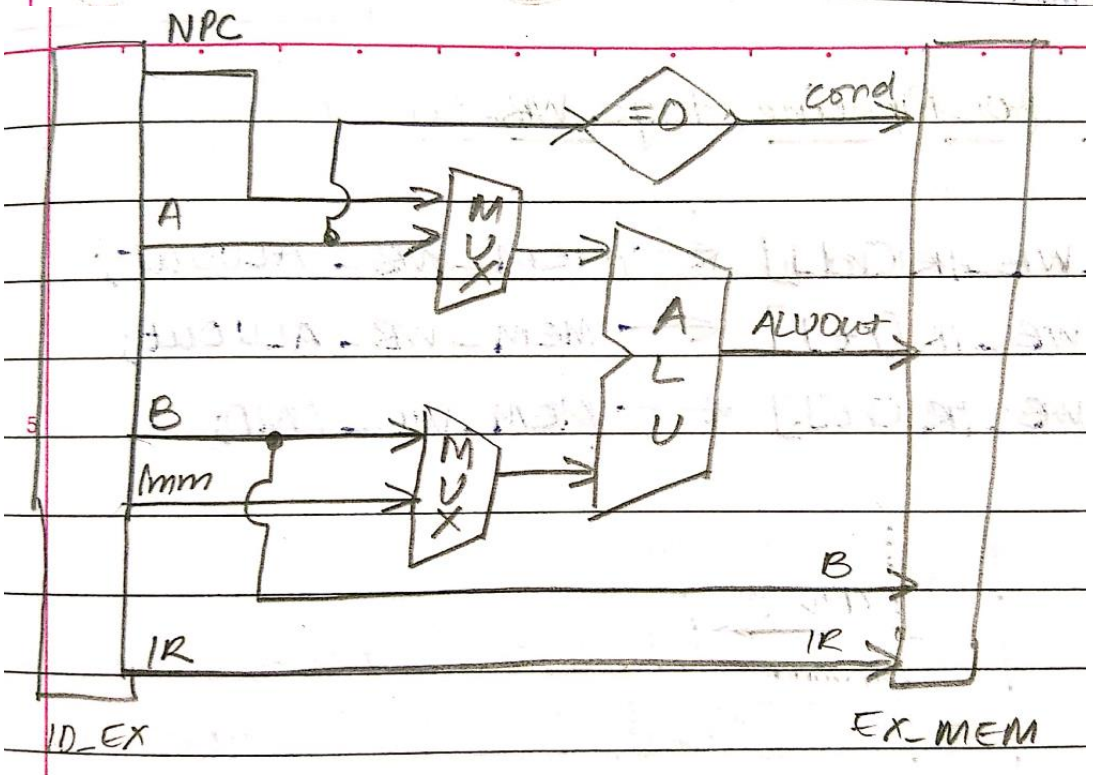
EX_MEM_B ← ID_EX_B;

Branch

EX_MEM_ALUOut ← ID_EX_NPC + ID_EX_Imm;

EX_MEM_cond ← (ID_EX_A == 0);

EX_MEM_IR ← ID_EX_IR;



(d) Micro-operations for Pipeline Stage MEM

ALU:

MEM_WB_IR ← EX_MEM_IR;

MEM_WB_ALUOut ← EX_MEM_ALUOut;

Load:

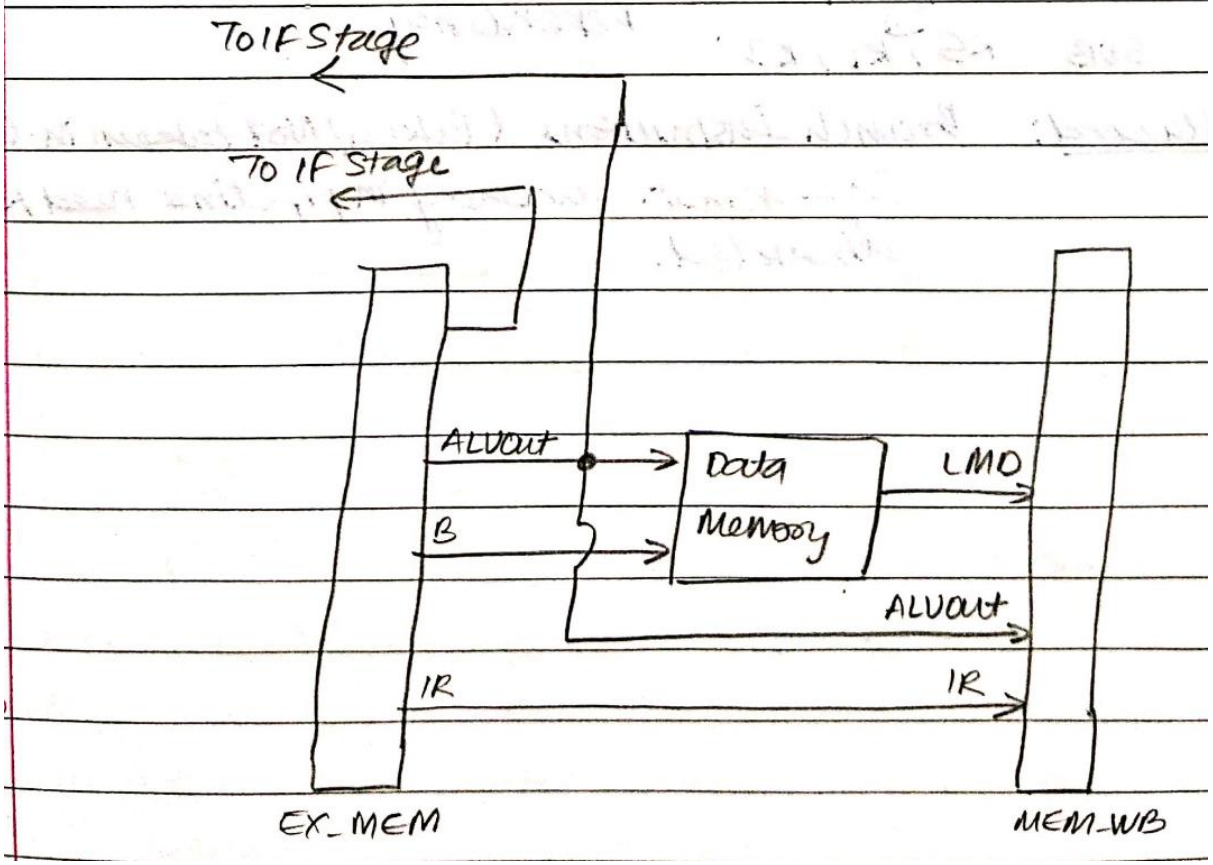
MEM_WB_IR ← EX_MEM_IR;

MEM_WB_LMD ← Mem[EX_MEM_ALUOut];

Store:

MEM_WB_IR ← EX_MEM_IR;

Mem[EX_MEM_ALUOut] ← EX_MEM_B;

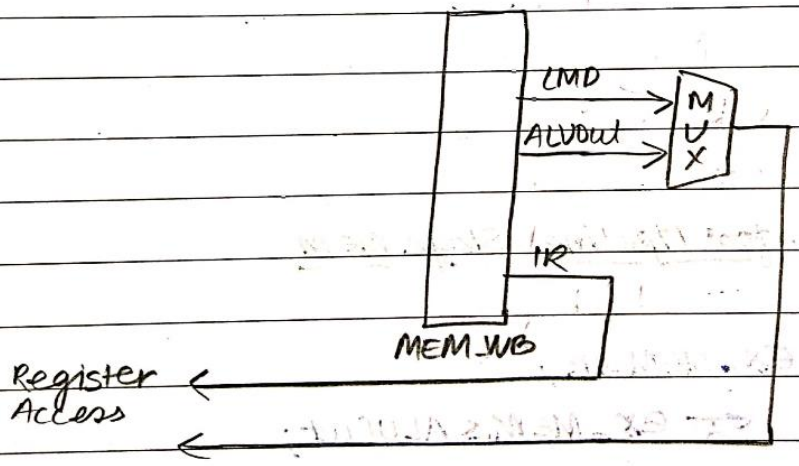


(e) Micro - Operations for Pipeline Stage WB

R-R ALU: Reg[MEM_WB_IR[rd]] ← MEM_WB_ALUOut;

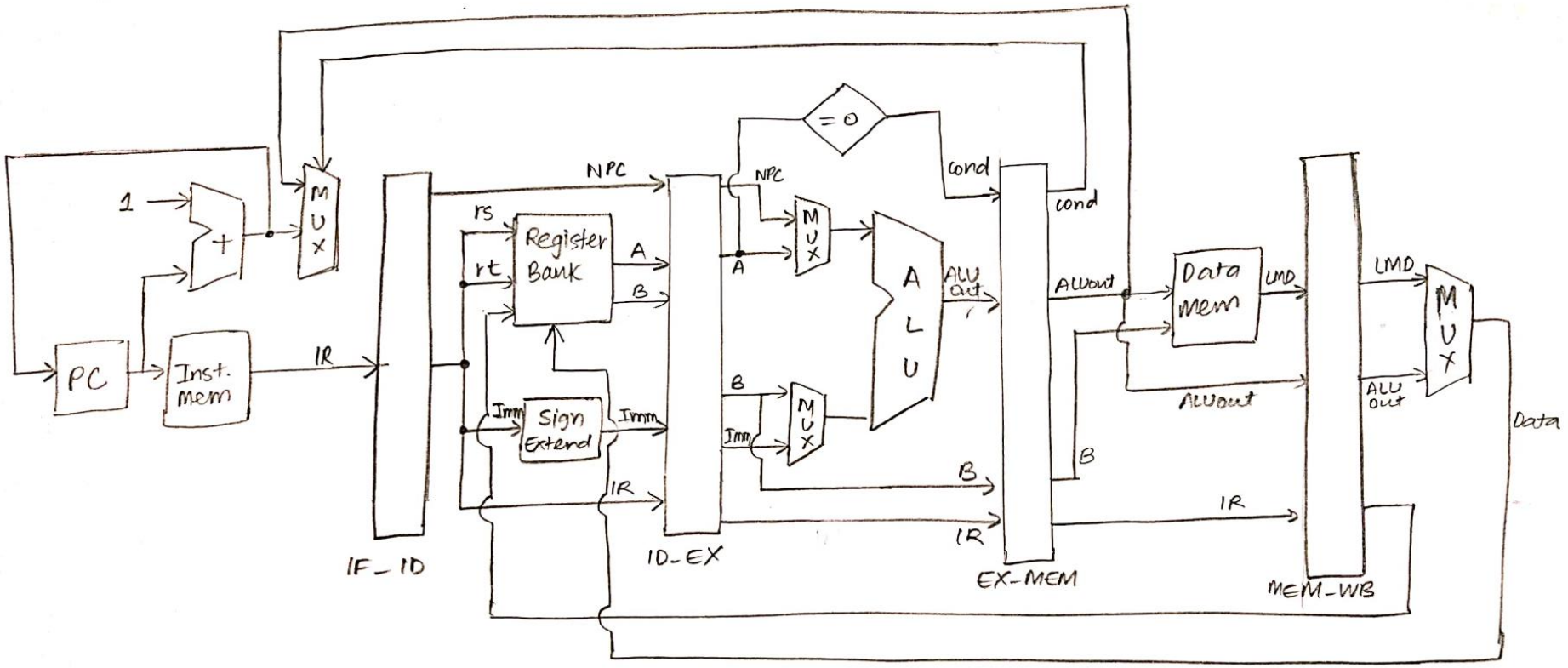
R-M ALU: Reg[MEM_WB_IR[rt]] ← MEM_WB_ALUOut;

LOAD : Reg[MEM_WB_IR[rt]] ← MEM_WB_LMD;



Two special 1-bit variables are used:

- HALTED :: Set after a HLT instruction executes and reaches the WB stage.
- TAKEN_BRANCH:: Set after the decision to take a branch is known. Required to disable the instructions that have already entered the pipeline from making any state changes.



PIPELINED - MIPS 32