# EE-309 - Microprocessors

## Course Project

# IITB-RISC Pipelined Implementation

**Prakhar Bansal : 200070056**
**Pulkit Adil : 200070062**
**Sahil Garg : 200070070**
**Samarth Agarwal : 200070072**

# IITB RISC PIPELINE

## Stages

1. Instruction Fetch
2. Instruction Decode
3. Register Read
4. Execute
5. Memory Read/ Write
6. Write Back

## Components used in different stages along with their functions

1. **Instruction Fetch**

- **Code Memory**- It stores all the instructions. The instructions are fetched from here by program counter.
- **PC**- It stores the address of instruction stored in code memory
- **PC_Inc**- It is used to increment the value of Program Counter by 1

2. **Instruction Decode**

- **SE**- Sign Extender used to convert 6 bit or 9 bit numbers to 16 bit.
- **SE_ALU** - Used to add sign extended value from sign extender to program counter
- **LS** - Used to left shift a 9 bit number to make it a 16 bit number by padding 7 zeroes to the right.

3. **Register Read**

- **Register File**- It contains 8 special function registers (R0-R7)

4. **Execute**

- **ALU** - Arithmetic Logical Unit handles all the logical instructions (ADD and NAND).
- **EQ** - Used to check equality in the BEQ instruction and give the final 16-bit output.
- **1S** - Used to left shift a 16 bit number by padding 1 zero to the right

5. **Memory Read/Write**

- **Data Memory** - Instruction read and write data from this memory block

6. **Write Back**

- No additional hardware component used. A MUX is used to find the final output.

## Flow of instructions through various stages

1. **ADD**
   IF → ID → RR → EX → WB

2. **ADC**
   IF → ID → RR → EX → WB

3. **ADZ**
   IF → ID → RR → EX → WB

4. **ADL**
   IF → ID → RR → EX → WB

5. **ADI**
   IF → ID → RR → EX → WB

6. **NDU**
   IF → ID → RR → EX → WB

7. **NDC**
   IF → ID → RR → EX → WB

8. **NDZ**
   IF → ID → RR → EX → WB

9. **LHI**
   IF → ID → WB

10. **LW**
    IF → ID → RR → EX → Mem → WB

11. **SW**
    IF → ID → RR → EX → Mem

**12. LM**

LM is a special instruction doing LW multiple times. In this, instruction fetch stage is disabled for a few number of cycles (8)

$IF \rightarrow (ID \rightarrow RR \rightarrow EX \rightarrow Mem \rightarrow WB)_8$

**13. SM**

SM is a special instruction doing SW multiple times. In this, instruction fetch stage is disabled for a few number of cycles (8)

$IF \rightarrow (ID \rightarrow RR \rightarrow EX \rightarrow Mem)_8$

**14. BEQ**

$IF \rightarrow ID \rightarrow RR \rightarrow EX$

**15. JAL**

$IF \rightarrow ID \rightarrow WB$

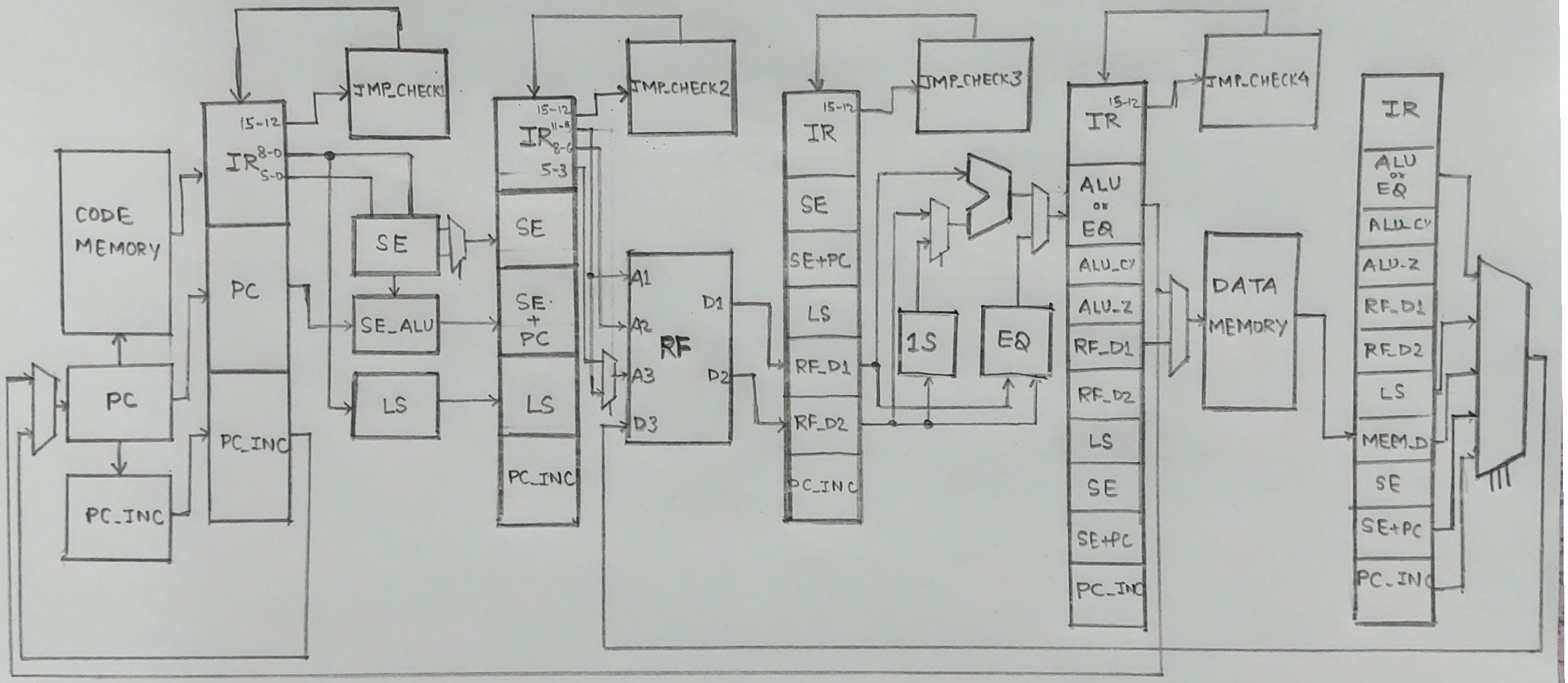
**16. JLR**

$IF \rightarrow ID \rightarrow RR \rightarrow WB$

**17. JRI**

$IF \rightarrow ID \rightarrow RR \rightarrow Ex \rightarrow Mem \rightarrow WB$

# DATAPATH


CODE MEMORY
PC
PC_INC
PC
PC_INC
IR
15-12
8-0
5-0
JMP_CHECK1
SE
SE_ALU
LS
IR
11-8
8-6
5-3
JMP_CHECK2
SE
SE + PC
LS
PC_INC
RF
A1
A2
A3
D3
D1
D2
IR
15-12
JMP_CHECK3
SE
SE+PC
LS
RF_D1
RF_D2
PC_INC
1S
EQ
IR
15-12
ALU or EQ
ALU_CY
ALU_Z
RF_D1
RF_D2
LS
SE
SE+PC
PC_INC
JMP_CHECK4
DATA MEMORY
IR
ALU or EQ
ALU_CY
ALU_Z
RF_D1
RF_D2
LS
MEM_D
SE
SE+PC
PC_INC