## Instructions Sequence:

**Instructions divided into 4 categories:**

- Instructions with no operand
- Instructions with only one operands
- Instructions with only two operands
- Instructions with three operands

1) **Instructions with no operand:**

   **Divided into 3 groups as following:**

   - CLRC , SETC → For carry bit
   - RET , RTI → For Memory (SP to PC)
   - NOP

2) **Instructions with one operand:**

   **Divided into 2 Groups as following:**

   1) **Read from register file and this divided into 4 Groups:**
      - OUT → Destination is OUT port
      - PUSH → Destination is Memory
      - JMP , JZ , JC → Destination is PC
      - CALL → Destination is PC and Memory
   2) **Write into register file and this divided into 2 Groups:**
      1) IN → Source is IN Port
      2) POP → Source is Memory

3) **Instructions with two operands:**

   **Divided into 4 groups as following:**

   1) **Memory:**
      - LOAD
        1. LDM → SRC will be immediate value
        2. LDD → SRC will be any Register

- Store → STD

2) **MOV**
3) **NOT**
4) **INC or DEC**

**4) Instructions with Three operands:**

**Divided into 4 Groups as following:**

1) **Adding :**
   - ADD → SRC2 will be from Register file
   - IADD → will read from Immediate value
2) **SUB**
3) **AND**
4) **OR**

---

## Instruction details:

Total Instruction → 32 bits

RDST & RSRC1 & RSRC2 → 3-bits , X → Don't Care , IMM (16 bits immediate value)

**NOTE)** Colored bits means → will be read from Register File

1) **NOP → 00 RDST RSRC1 RSRC2 X 00 XX IMM**
2) **SETC → 00 RDST RSRC1 RSRC2 0 01 XX IMM**
3) **CLRC → 00 RDST RSRC1 RSRC2 1 01 XX IMM**
4) **RET → 00 RDST RSRC1 RSRC2 0 10 XX IMM**
5) **RTI → 00 RDST RSRC1 RSRC2 1 10 XX IMM**

[31:30] → Identify one of four categories that mentioned above

[19:18] → Select one operation from the selected category

[20] → Used only for Identify which operation will be done on the carry (SET , CLEAR) and for RTI and RET to identify restoring of PC only or PC and Flag registers

-------------------------------------------------------------------------------------------

31:30 rd:rs1:rs2 20: 19:18 17:16

**6) OUT → 01 RDST RSRC1 RSRC2 0 00 XX IMM**
**7) PUSH → 01 RDST RSRC1 RSRC2 0 01 XX IMM**
**8) JZ → 01 RDST RSRC1 RSRC2 0 10 10 IMM**
**9) JC → 01 RDST RSRC1 RSRC2 0 10 01 IMM**

**10) JMP → 01 RDST RSRC1 RSRC2 0 10 11 IMM**

**11) CALL → 01 RDST RSRC1 RSRC2 0 11 11 IMM**

**12) IN → 01 RDST RSRC1 RSRC2 1 00 XX IMM**

**13) POP → 01 RDST RSRC1 RSRC2 1 01 XX IMM**

[31:30] → Select one of Categories mentioned above

[20] → Indicate Write operation or Read operation

[19:18] → Select one Operation from this Category

[17:16] → Not be used for all except JMP to Select which JMP

NOTE: IN/OUT has same [19:18] Select between them by bit 20

-------------------------------------------------------------------------------------------

**14) MOV → 10 RDST RSRC1 RSRC2 X 00 0X IMM**

**15) NOT → 10 RDST RSRC1 RSRC2 X 01 0X IMM ( ALU )**

**16) INC → 10 RDST RSRC1 RSRC2 0 10 0X IMM ( ALU )**

**17) DEC → 10 RDST RSRC1 RSRC2 1 10 0X IMM ( ALU )**

**18) LDM → 10 RDST RSRC1 RSRC2 0 11 1X IMM ( ALU + Imm )**

**19) LDD → 10 RDST RSRC1 RSRC2 0 11 0X IMM**

**20) STD → 10 RDST RSRC1 RSRC2 1 11 0X IMM**

[31:30]→ Select one of four categories mentioned above

[19:18]→ Select which operation from this category

Notes: INC/DEC have same [19:18] select Add or sub by bit 20 and

LDM/ LDD have same [19:18] we will select IMM or Reg by anding bit [17] with [31] if 1 → REG and else IMM value (16-bits) , Store also has same [19:18] as LDM/LDD but we distinct it from LOAD operations by bit [20]

-----------------------------------------------------------------------------------------

**21) ADD → 11 RDST RSRC1 RSRC2 X 00 0X IMM ( ALU )**

**22) IADD → 11 RDST RSRC1 RSRC2 X 00 1X IMM ( ALU + Imm )**

**23) SUB → 11 RDST RSRC1 RSRC2 X 01 0X IMM ( ALU )**

**24) AND → 11 RDST RSRC1 RSRC2 X 10 0X IMM ( ALU )**

**25) OR → 11 RDST RSRC1 RSRC2 X 11 0X IMM ( ALU )**

[31:30]→ Select one of four categories mentioned above

[19:18]→ Select one operations from this category

Notes: ADD/IADD have same [19:18] and we will difference between them by bit [17] anding with bit 31 and if 1 → IMM value will be used , else → REG will be used