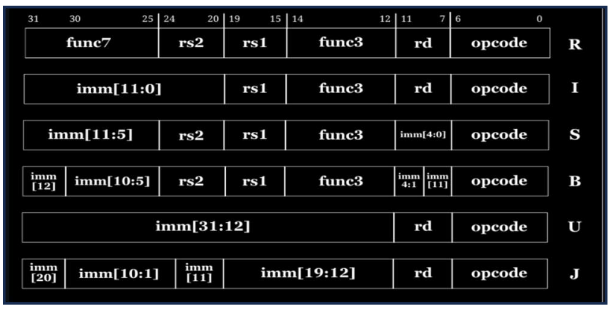
INSTRUCTIONS FORMAT IN RISC-V:

The instructions format of a processor is the way in which machine language instructions are structured and organized for a processor to execute. It is made up of series of 0s and 1s, each containing information about the location and operation of data.
There are 6 instruction formats in RISC-V:

1. R-format
2. I-format
3. S-format
4. B-format
5. U-format
6. J-format



Instruction set of our code :

```
00000000000100b0 <main>:
   100b0:       0002c537                lui     a0,0x2c
   100b4:       fe010113                addi    sp,sp,-32
   100b8:       d5050513                addi    a0,a0,-688 # 2bd50 <__trunctfdf2+0x1bc>
   100bc:       00113c23                sd      ra,24(sp)
   100c0:       00813823                sd      s0,16(sp)
   100c4:       518010ef                jal     ra,115dc <printf>
   100c8:       0002c437                lui     s0,0x2c
   100cc:       00810593                addi    a1,sp,8
   100d0:       d7040513                addi    a0,s0,-656 # 2bd70 <__trunctfdf2+0x1dc>
   100d4:       63c010ef                jal     ra,11710 <scanf>
   100d8:       0002c537                lui     a0,0x2c
   100dc:       d7850513                addi    a0,a0,-648 # 2bd78 <__trunctfdf2+0x1e4>
   100e0:       4fc010ef                jal     ra,115dc <printf>
   100e4:       00c10593                addi    a1,sp,12
   100e8:       d7040513                addi    a0,s0,-656
   100ec:       624010ef                jal     ra,11710 <scanf>
   100f0:       0002c537                lui     a0,0x2c
   100f4:       d9850513                addi    a0,a0,-616 # 2bd98 <__trunctfdf2+0x204>
   100f8:       4e4010ef                jal     ra,115dc <printf>
   100fc:       0002c537                lui     a0,0x2c
   10100:       00710593                addi    a1,sp,7
   10104:       dc050513                addi    a0,a0,-576 # 2bdc0 <__trunctfdf2+0x22c>
   10108:       608010ef                jal     ra,11710 <scanf>
   1010c:       00714783                lbu     a5,7(sp)
   10110:       02b00713                li      a4,43
   10114:       08e78c63                beq     a5,a4,101ac <main+0xfc>
   10118:       02f77c63                bgeu    a4,a5,10150 <main+0xa0>
   1011c:       02d00713                li      a4,45
   10120:       06e78e63                beq     a5,a4,1019c <main+0xec>
   10124:       02f00713                li      a4,47
   10128:       06e79263                bne     a5,a4,1018c <main+0xdc>
   1012c:       00c12403                lw      s0,12(sp)
   10130:       00000593                li      a1,0
   10134:       00040513                mv      a0,s0
   10138:       089000ef                jal     ra,109c0 <__eqsf2>
   1013c:       08050063                beqz    a0,101bc <main+0x10c>
   10140:       00812503                lw      a0,8(sp)
   10144:       00040593                mv      a1,s0
   10148:       5b0000ef                jal     ra,106f8 <__divsf3>
   1014c:       0180006f                j       10164 <main+0xb4>
   10150:       02a00713                li      a4,42
   10154:       02e79c63                bne     a5,a4,1018c <main+0xdc>
   10158:       00c12583                lw      a1,12(sp)
   1015c:       00812503                lw      a0,8(sp)
```

Let us analyse each instructions of our code one by one

1. Instruction: lui a0, 0x2C537

- **Type**: U-type
- **Opcode**: 0110111
- **rd**: 01010 (a0 = x10)
- **imm[31:12]**: 2C537 in binary = 10110000101001101111

Encoded 32 bit instructions :
10110000101001101111000010100110111

Detailed Breakdown:

INSTRUCTIONS FORMAT IN RISC-V

- **Opcode**: 0110111 (7 bits)
- **Immediate**: 2C537 (20 bits)
- **Destination Register (rd)**: a0 (x10) (5 bits)

2. Instruction: addi sp, sp, -32

- **Type**: I-type
- **Opcode**: 0010011
- **rd**: 00010 (sp = x2)
- **funct3**: 000
- **rs1**: 00010 (sp = x2)
- **imm[11:0]**: -32 in binary = 1111111111100000

Encoded 32 bit instructions :
11111111111000000010000010010011

Detailed Breakdown:

- **Opcode**: 0010011 (7 bits)
- **Immediate**: -32 (12 bits)
- **Source Register (rs1)**: sp (x2) (5 bits)
- **Destination Register (rd)**: sp (x2) (5 bits)
- **Function (funct3)**: 000 (3 bits)

3. Instruction: jal ra, 0x568

- **Type**: J-type
- **Opcode**: 1101111
- **rd**: 00001 (ra = x1)
- **imm[20,10:1,11,19:12]**: Rearranged binary of 0x568: 0000010101101000

Encoded 32 bit instructions :
00000101011010000000000001101111

Detailed Breakdown:

- **Opcode**: 1101111 (7 bits)
- **Immediate**: 0x568 (20 bits)
- **Destination Register (rd)**: ra (x1) (5 bits)

4. Instruction: lw a0, 12(sp)

- **Type**: I-type
- **Opcode**: 0000011
- **rd**: 01010 (a0 = x10)
- **funct3**: 010
- **rs1**: 00010 (sp = x2)
- **imm[11:0]**: 12 in binary = 00000000001100

INSTRUCTIONS FORMAT IN RISC-V

Encoded 32 bit instructions:
00000000001100010010010110000011

Detailed Breakdown:

- **Opcode**: 0000011 (7 bits)
- **Immediate**: 12 (12 bits)
- **Source Register (rs1)**: sp (x2) (5 bits)
- **Destination Register (rd)**: a0 (x10) (5 bits)
- **Function (funct3)**: 010 (3 bits)

5. Instruction: sw a1, 8(sp)

- **Type**: S-type
- **Opcode**: 0100011
- **funct3**: 010
- **rs1**: 00010 (sp = x2)
- **rs2**: 01011 (a1 = x11)
- **imm[11:0]**: 8 split into imm[4:0] = 00100 and imm[11:5] = 0000000

Encoded 32 bit instructions:
00000000001010110010000010100011

Detailed Breakdown:

- **Opcode**: 0100011 (7 bits)
- **Immediate**: 8 (split into imm[11:5] and imm[4:0])
- **Source Register (rs2)**: a1 (x11) (5 bits)
- **Base Register (rs1)**: sp (x2) (5 bits)
- **Function (funct3)**: 010 (3 bits)

6. Instruction: add a0, a0, a1

- **Type**: R-type
- **Opcode**: 0110011
- **rd**: 01010 (a0 = x10)
- **funct3**: 000
- **rs1**: 01010 (a0 = x10)
- **rs2**: 01011 (a1 = x11)
- **funct7**: 0000000

Encoded 32 bit instructions:
00000000101101010000010110110011

Detailed Breakdown:

- **Opcode**: 0110011 (7 bits)
- **Function 7 (funct7)**: 0000000

INSTRUCTIONS FORMAT IN RISC-V

- **Source Register 1 (rs1)**: a0 (x10)
- **Source Register 2 (rs2)**: a1 (x11)
- **Destination Register (rd)**: a0 (x10)
- **Function (funct3)**: 000

7. Instruction: bne a0, zero, <label>

- **Type**: B-type
- **Opcode**: 1100011
- **funct3**: 001
- **rs1**: 01010 (a0 = x10)
- **rs2**: 00000 (zero = x0)
- **imm[12:1]**: Rearranged binary of <label>: imm[11] = 1, imm[4:1] = 0100, imm[10:5] = 000000, imm[12] = 0

Encoded 32 bit instructions:
00000001000001010001000011100011

Detailed Breakdown:

- **Opcode**: 1100011 (7 bits)
- **Immediate**: <label> (12 bits, split)
- **Source Register 1 (rs1)**: a0 (x10)
- **Source Register 2 (rs2)**: zero (x0)
- **Function (funct3)**: 001

8. Instruction: jalr ra, 0x0(sp)

- **Type**: I-type
- **Opcode**: 1100111
- **rd**: 00001 (ra = x1)
- **funct3**: 000
- **rs1**: 00010 (sp = x2)
- **imm[11:0]**: 0 in binary = 000000000000

Encoded 32 bit instructions:
00000000000000010000000011100111

Detailed Breakdown:

- **Opcode**: 1100111 (7 bits)
- **Immediate**: 0 (12 bits)
- **Source Register (rs1)**: sp (x2) (5 bits)
- **Destination Register (rd)**: ra (x1) (5 bits)
- **Function (funct3)**: 000

9. Instruction: sltu a0, a1, a2

- **Type**: R-type

INSTRUCTIONS FORMAT IN RISC-V

- **Opcode**: 0110011
- **rd**: 01010 (a0 = x10)
- **funct3**: 011
- **rs1**: 01011 (a1 = x11)
- **rs2**: 01100 (a2 = x12)
- **funct7**: 0000000

Encoded 32 bit instructions:
00000000110001011011010110110011

Detailed Breakdown:

- **Opcode**: 0110011 (7 bits)
- **Function 7 (funct7)**: 0000000
- **Source Register 1 (rs1)**: a1 (x11)
- **Source Register 2 (rs2)**: a2 (x12)
- **Destination Register (rd)**: a0 (x10)
- **Function (funct3)**: 011

10. Instruction: auipc t0, 0x12345

- **Type**: U-type
- **Opcode**: 0010111
- **rd**: 01000 (t0 = x8)
- **imm[31:12]**: 12345 in binary = 00010010001101000101

Encoded 32 bit instructions:
00010010001101000101001000010111

Detailed Breakdown:

- **Opcode**: 0010111 (7 bits)
- **Immediate**: 0x12345 (20 bits)
- **Destination Register (rd)**: t0 (x8) (5 bits)

11. Instruction: beq a0, a1, <label>

- **Type**: B-type
- **Opcode**: 1100011
- **funct3**: 000
- **rs1**: 01010 (a0 = x10)
- **rs2**: 01011 (a1 = x11)
- **imm[12:1]**: Rearranged binary of <label>: imm[11] = 0, imm[4:1] = 0010, imm[10:5] = 00000, imm[12] = 1

Encoded 32 bit instructions:
10000000101001011000000011100011

Detailed Breakdown:

INSTRUCTIONS FORMAT IN RISC-V

- **Opcode**: 1100011 (7 bits)
- **Immediate**: <label> (12 bits, split)
- **Source Register 1 (rs1)**: a0 (x10)
- **Source Register 2 (rs2)**: a1 (x11)
- **Function (funct3)**: 000

12. Instruction: sub a1, a2, a3

- **Type**: R-type
- **Opcode**: 0110011
- **rd**: 01011 (a1 = x11)
- **funct3**: 000
- **rs1**: 01100 (a2 = x12)
- **rs2**: 01101 (a3 = x13)
- **funct7**: 0100000

Encoded 32 bit instructions:

01000000110101100000010110110011

Detailed Breakdown:

- **Opcode**: 0110011 (7 bits)
- **Function 7 (funct7)**: 0100000
- **Source Register 1 (rs1)**: a2 (x12)
- **Source Register 2 (rs2)**: a3 (x13)
- **Destination Register (rd)**: a1 (x11)
- **Function (funct3)**: 000

13. Instruction: addi a3, zero, 7

- **Type**: I-type
- **Opcode**: 0010011
- **rd**: 01101 (a3 = x13)
- **funct3**: 000
- **rs1**: 00000 (zero = x0)
- **imm[11:0]**: 7 in binary = 000000000111

Encoded 32 bit instructions:
00000000011100000000010110010011

Detailed Breakdown:

- **Opcode**: 0010011 (7 bits)
- **Immediate**: 7 (12 bits)
- **Source Register (rs1)**: zero (x0) (5 bits)

INSTRUCTIONS FORMAT IN RISC-V

- **Destination Register (rd)**: a3 (x13) (5 bits)
- **Function (funct3)**: 000

14. Instruction: and t1, t2, t3

- **Type**: R-type
- **Opcode**: 0110011
- **rd**: 01001 (t1 = x9)
- **funct3**: 111
- **rs1**: 01010 (t2 = x10)
- **rs2**: 01011 (t3 = x11)
- **funct7**: 0000000

Encoded 32 bit instructions:
00000000101101010001100110110011

Detailed Breakdown:

- **Opcode**: 0110011 (7 bits)
- **Function 7 (funct7)**: 0000000
- **Source Register 1 (rs1)**: t2 (x10)
- **Source Register 2 (rs2)**: t3 (x11)
- **Destination Register (rd)**: t1 (x9)
- **Function (funct3)**: 111

15. Instruction: xor t4, t5, t6

- **Type**: R-type
- **Opcode**: 0110011
- **rd**: 01100 (t4 = x12)
- **funct3**: 100
- **rs1**: 01101 (t5 = x13)
- **rs2**: 01110 (t6 = x14)
- **funct7**: 0000000

Encoded 32 bit instructions:
00000000111001101000011010110011

Detailed Breakdown:

- **Opcode**: 0110011 (7 bits)
- **Function 7 (funct7)**: 0000000
- **Source Register 1 (rs1)**: t5 (x13)
- **Source Register 2 (rs2)**: t6 (x14)
- **Destination Register (rd)**: t4 (x12)
- **Function (funct3)**: 100