**Task 3**

**Types of Instruction sets in objdump:**

1. lui a0,0x21

* Opcode: 0110111 (7 bits)
* Immediate: 0x21 (20 bits)
* Destination Register(rd): a0 (15, 5 bits)

Breakdown:

* Opcode: 0110111
* Immediate: 0000 0000 0000 0010 0001 (20 bits)
* rd (a0=15): 01010 (5 bits)

1. addi sp,sp,-16

* Opcode: 0010011
* Immediate: -16 (12 bits,2’s complement)
* Source Register (rs1): sp (x2, 5 bits)
* Destination register (rd): sp (x2, 5bits)
* Function (funct3): 000 (3 bits)

Breakdown:

* Opcode: 0010011
* Immediate: 1111 1111 0000 (12 bits)
* rs1: 00010 (5 bits)
* rd: 00010 (5 bits)
* funct3: 000 (3 bits)

1. li a2,120

* Machine Code: 07800613

Breakdown:

* Immediate: 0000 0111 1000 (12 bits)
* Source Register (rs1): 00000 (5 bits)
* Function (funct3): 000 (3 bits)
* Destination Register (rd): 01100 (5 bits)
* Opcode: 0010011 (7 bits)

1. sd ra,8(sp)

* Machine Code: 00113423

Breakdown:

* Immediate (imm[11:5]): 0000000 (7 bits)
* Source Register (rs2): 00001 (5 bits)
* Source Register (rs1): 00010 (5 bits)
* Function (funct3): 011 (3 bits)
* Immediate (imm[4:0]): 01000 (5 bits)
* Opcode: 0100011 (7 bits)

1. jal ra,10408

* Machine Code: 340000EF

Breakdown:

* Immediate: 0100 0000 0000 0000
* Destination Register (rd): 00001
* Opcode: 1101111

1. ld ra,8(sp)

* Opcode: 0000011
* Immediate: 000000000100 (8 in decimal)
* Source Register (rs1): 00010 (for sp)
* Destination Register (rd): 00001 (for ra)
* Function (funct3): 011

1. auipc a5, 0xfffff0 (0x100dc)

* Type: U-Type (Add Upper Immediate to PC).

Fields:

* opcode: 0010111 (AUIPC).
* rd: a5.
* imm[31:12]: 0xfffff0.
* Function: Sets a5 = PC + (imm << 12)

1. beqz a5, 100f4 (0x100e4)

* Type: B-Type (Branch).

Fields:

* opcode: 1100011.
* funct3: 000 (BEQ).
* rs1: a5.
* rs2: zero.
* imm: Offset to 100f4.
* Function: Branches to 100f4 if a5 == 0.

1. sub a2, a2, a0

* Type: R-Type (Register Arithmetic).

Fields:

* opcode: 0110011.
* funct3: 000 (SUB).
* rd: a2.
* rs1: a2.
* rs2: a0.
* Function: Subtracts a0 from a2 and stores the result in a2

1. lw a0, 0(sp)

* Type: I-Type (Load).

Fields:

* opcode: 0000011.
* funct3: 010 (LW).
* rd: a0.
* rs1: sp.
* imm[11:0]: 0.
* Function: Loads the value from the memory address sp + 0 into a0.

1. lbu a5, 1944(gp)

* Type: I-Type (Load Unsigned Byte).

Fields:

* opcode: 0000011 (LB).
* funct3: 100 (LBU - Load Unsigned Byte).
* rd: a5.
* rs1: gp.
* imm[11:0]: 1944.
* Function: Loads an unsigned byte from memory at gp + 1944 into register a5.

1. bnez a5, 10188 <\_\_do\_global\_dtors\_aux>

* Type: B-Type (Branch).

Fields:

* opcode: 1100011.
* funct3: 001 (BNE - Branch if Not Equal).
* rs1: a5.
* rs2: zero.
* imm[12|10:5|4:1|11]: Relative offset to 10188.
* Function: Branches to 10188 if a5 != 0.

1. bnez a5, 10188 <\_\_do\_global\_dtors\_aux>

* Type: B-Type (Branch).

Fields:

* opcode: 1100011.
* funct3: 001 (BNE - Branch if Not Equal).
* rs1: a5.
* rs2: zero.
* imm[12|10:5|4:1|11]: Relative offset to 10188.
* Function: Branches to 10188 if a5 != 0.

1. Ret

* Pseudo-instruction: jalr zero, 0(ra).
* Function: Returns from the function by jumping to the address in ra.

1. sb a5, 1944(gp)

* Type: S-Type (Store Byte).

Fields:

* opcode: 0100011.
* funct3: 000 (SB - Store Byte).
* rs2: a5.
* rs1: gp.
* imm[11:5|4:0]: 1944.
* Function: Stores the byte value of a5 (1) into memory at gp + 1944.