Question 3 (10 points):

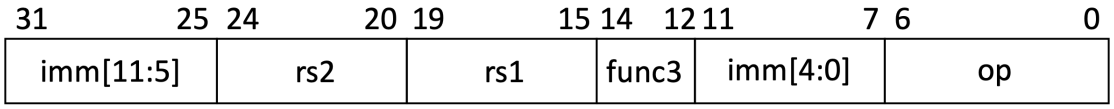

Figure 1: S-Type format. Used for store instructions in RISC-V.

This question explores the binary format of RISC-V instructions. This question will focus on the S-Type format used for the `sw` and for the `sb` instructions. Figure 1 shows the S-Type format. The opcode for both an `sw` and an `sb` instruction is `0100011`. The func3 code for `sb` is 000, while the func3 code for `sw` is `010`. In assembly the `sw` and `sb` instruction is expressed as follows — `(31:0)` indicates that the 32 bits of the register are used and `(7:0)` indicates that the eight least significant bits are used:

```
sw  rs2, imm(rs1)   # Mem[Reg[rs1]+imm](31:0) <- Reg[rs2](31:0)
sb  rs2, imm(rs1)   # Mem[Reg[rs1]+imm](7:0) <- Reg[rs2](7:0)
```

a. (**5 points**) The binary encoding of an instruction fetched from memory is `0xFF142623`. What is the assembly code for this RISC-V instruction?

```
0xFF142623
1111 1111 0001 0100 0010 0110 0010 0011
1111111 10001 01000 010 01100 0100011
immediate = 111111101100 = -20
rs2 = 10001 = x17 = a7
rs1 =  01000 = x8 = s0 = fp
```

Thus the assembly for the instruction is:

```
sw a7, -20(fp)
```

b. (**5 points**) What is the binary representation, expressed in hexadecimal, for the following assembly instruction?

```
sb t5, 2047(s10)
```

$$\begin{aligned} \text{immediate} &= 2047 = 2048 - 1 = 2^{11} - 1 \\ &= 0...0000100000000000 - 1 \\ &= 0...0000011111111111 \end{aligned}$$

opcode = 0100011
imm[4:0] = 11111
func3 = 000
rs1 = s10 = x26 = 11010
rs2 = t5 = x30 = 11110
imm[11:5] = 0111111

binary representation: 0111111 11110 11010 000 11111 0100011
0111 1111 1110 1101 0000 1111 1010 0011

Hexadecimal representation = 0x7FED0FA3