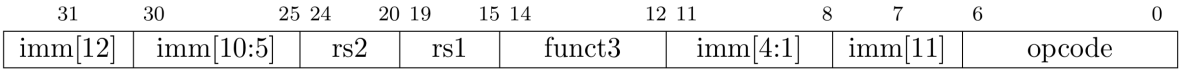
►Solution◄

Question 1: (0 points)
Bank of Questions

A RISC-V with a Large Register File (V05, V0A)

RISCV-LRF is a RISC-V instruction-set architecture with a large register file. RISCV-LRF has 128 registers and the instruction formats have to accommodate this change. Consider a branch instruction. Assume that the instruction still has to be 32-bits long and that any changes made to the instruction format only increase/decrease the size of the immediate field. Recall that in the original RISC-V architecture, the branch instruction uses the following format:



Question 2: (10 points)
How many bits are available for the immediate bit field for a branch instruction in RISCV-LRF?

Solution: With $128 = 2^7$ registers, we need 2 additional bits to specify each register. Therefore there are 12-4=8 bits available for the immediate field.

Question 3: (10 points)
How does the range of a branch in RISCV-LRF compares with the range of a branch in the original RISC-V architecture?

Solution: With 4 fewer bits for the immediate field, the range is reduced 16 times, or it is reduced by 93.75%, or it is reduced to 6.25% of the original range.

Question 4: (20 points)
Assume that in a program that uses RISCV-LRF the current PC address is `0x 0000 0010`. How many branches (no jump instructions) are necessary to get to address `0x FFFF FF00`? Provide the binary representation for the immediate field for the branch(es) instruction(s) required.

Solution: The shortest path is through the use of backward branches. The largest negative number that can be stored in a 8-bit immediate field is 0x80. After left shifting by 1 and sign extending to 32 bits it becomes 0xFFFFFF00, which brings us close to the final address. Thus we start with the largest possible backward jump:

```
Old PC = 0000 0000 0000 0000 0000 0000 0001 0000
     +   1111 1111 1111 1111 1111 1111 0000 0000 = SigExt(immed)<<1
-----------------------------------------------------------------
         1111 1111 1111 1111 1111 1111 0001 0000 = 0xFFFF FF10
```

Therefore the PC after the first branch is 0xFFFF FF10

The immediate field of the first branch is: 0x80

We can see now that `PC after first branch = 0xFFFF FF10`

Thus to get to the final destination at 0xFFFF FF00, all we have to do is to subtract 16. With the left shift done by the branch, this is equivalent to an immediate field equal -8.

Thus the immediate field of the second branch is -8 = 0xF8

Verifying:

```
PC after 1st branch =
        1111 1111 1111 1111 1111 1111 0001 0000 = 0xFFFFFF10
    +   1111 1111 1111 1111 1111 1111 1111 0000 = SigExt(immed)<<1
-----------------------------------------------------------------
Final PC = 1111 1111 1111 1111 1111 1111 0000 0000 = 0xFFFF FF00
```

Therefore, in RISCV-LRF, two branches are required to go from address `0x 0000 0010` to address `0x FFFF FF00`.