sure, I am designing this thing, and know that encoding registers is ok in 4 bits only, so I expect first complete byte as opcode with some significant bits grouping instructions based on complexity of decoding to achieve fastest implementation, and also expecting second byte to contain registers, so rd (destination) and rs1 (first source), but I also want to encode 16bit compressed formats and have all mappable to RV32emc ISA. Question is, HOW to group instructions based on complexity of decoding process to achieve smallest code

To group instructions based on the complexity of the decoding process, you can follow these steps:

1. Analyze the instruction set: Start by analyzing the RV32E ISA and the RV32C (compressed) instruction set. Identify the different types of instructions, such as arithmetic, logical, memory access, and control flow instructions. Also, pay attention to how immediates are encoded and how different registers are used.
2. Identify common patterns: Find common patterns in instruction encoding, such as similar opcodes, register usage, or immediate field encoding. Group instructions that share similar encoding patterns together.
3. Define opcode mapping: Design the first byte (opcode) to group instructions based on their complexity of decoding. Use the most significant bits to group instructions with similar encoding structures. For example, you could use the two most significant bits to represent four primary instruction groups:
    - 00: Simple arithmetic and logical instructions
    - 01: Load and store instructions
    - 10: Control flow instructions
    - 11: Complex arithmetic and miscellaneous instructions
4. Encode registers: Encode the destination register (rd) and the first source register (rs1) in the second byte, as you mentioned. You can allocate 4 bits for each register, which will fit into a single byte.
5. Design compressed instruction mapping: To support 16-bit compressed instructions, analyze the RV32C instruction set to find common patterns and design an efficient mapping. You can use a similar approach as with the 32-bit instructions, but you'll need to find ways to further compress the encoding while maintaining compatibility with the RV32E ISA.

6. Define a transformation process: Define a clear transformation process between the standard RV32E, RV32C