is possible to transfor binary encoding of RV32E ISA (16 registers) into something more easily decodable by software simulatioin, according to instructions registers and immediate fields usage, expecting only 24 or 20bit address space?

Yes, it is possible to transform the binary encoding of RV32E ISA (16 registers) into a more easily decodable format for software simulation. You can create an intermediate representation of the binary-encoded instructions that simplifies the decoding process and helps to better visualize the instruction set.

Here's a high-level process to achieve this:

1. Decode the binary instruction: First, decode the binary-encoded instructions into a more readable format, such as an assembly language or an intermediate representation. You can use lookup tables or switch-case statements for this purpose.
2. Simplify register and immediate fields: Next, simplify the register and immediate fields of the instruction set according to the RV32E ISA (16 registers). For example, if the instruction set uses 5 bits to represent a register, but RV32E only has 16 registers, then only 4 bits are needed to represent the registers.
3. Adjust the address space: Since you are expecting a 24 or 20-bit address space instead of the standard 32-bit address space, you'll need to adjust the representation accordingly. This may involve truncating the addresses or mapping them into the smaller address space.
4. Implement the simulator: Finally, create a software simulator that can execute the transformed instructions. This simulator will have to be capable of handling the adjusted register and immediate fields and the reduced address space.

It's worth noting that if you're using an existing software simulator for RV32E ISA, you might have to modify the simulator to handle the transformed instructions and the smaller address space. However, if you're developing a new software simulator, you can design it with these constraints in mind from the beginning.
