well, thanks, this sounds good... so, now, what about the address space? I am considering 24 or 20 bits as maximum address space to limit particulary some immediate fields in case of less compact encoding of new ISA. Will be better to use 24 or only 20 bits? I am okay with 16MB space and want to limit some complex cases of LUI and AUIPC

If you are okay with a 16MB address space, then a 20-bit address space would be sufficient for your needs. A 20-bit address space can address up to 2^20 bytes (1,048,576 bytes or 1MB) of memory, and with RV32E, it is assumed that addresses are word-aligned, so you would have a 4MB address space for words (1MB * 4 bytes/word). However, if you need more memory, you can go with a 24-bit address space, which would allow addressing up to 2^24 bytes (16,777,216 bytes or 16MB) of memory, or 64MB address space for words (16MB * 4 bytes/word).

When you reduce the address space, some instructions will need to be adjusted accordingly:

1. LUI (Load Upper Immediate): The LUI instruction loads a 20-bit immediate value into the upper 20 bits of the destination register (rd), leaving the lower 12 bits as zeros. In a 20-bit address space, you would only need a 16-bit immediate field, as the upper 4 bits would always be zero.
2. AUIPC (Add Upper Immediate to PC): The AUIPC instruction adds a 20-bit immediate value, left-shifted by 12 bits, to the program counter (PC) and stores the result in the destination register (rd). Similar to LUI, you would only need a 16-bit immediate field in a 20-bit address space.

To accommodate the smaller address space, you can modify the encoding of these instructions by reducing the immediate field size. For example, you could use 16 bits for the immediate field in LUI and AUIPC instructions when using a 20-bit address space. This would save encoding space and simplify the decoding process.

In summary, if you are comfortable with a 16MB address space, a 20-bit address space would be a good choice. It allows for simpler encoding and decoding while still providing enough memory for your use case.