can you please sumarize my wished design I comunicated with you?

Certainly. Based on our conversation, it seems that you are interested in designing a new higher-level assembler for a virtual RV32E processor that is optimized for compactness and portability across a range of 8-bit and 16-bit MCUs. Specifically, you are looking to:

- Group the RV32E ISA instructions based on their register and immediate field usage, in order to minimize the complexity of the decoder.
- Use a 24-bit address space to allow for up to 16MB of memory.
- Map the registers of the target MCUs to a virtual zero page, which is a 64-byte block of memory in the RV32E processor's memory map.
- Implement a set of pseudo-instructions that provide equivalent functionality to the most commonly used instructions on the target MCUs.
- Optimize the generated code for size and performance by using efficient instructions and avoiding unnecessary register spills and reloads.
- Document the syntax and usage of the new assembler thoroughly, and provide clear examples of how to use it to write code for the RV32E interpreter.

Overall, your goal is to have a common and portable syntax for writing assembly code that can be used across a range of MCUs, including AVRxt, PIC16, PIC18, PIC24, 8080, and Z80.