well, knowing this, some of the MCUs have single accumulator and 6502 has also ZERO PAGE, which I consider as set of 256 registers in fact, virtually... for the new assembler, I dont want to access the registers of such MCUs directly, but I want to allow window in memory emulating the ZERO PAGE, in fact, as RV32E has 16 4byte registers, this is 64byte space, so I want for all such MCUs to have such 64byte space as virtual zero page, similar as 6502 has. All this for most uniform higher-level assembler usage. Can you recommend something?

Yes, one option for emulating a virtual zero page for all the MCUs you mentioned would be to reserve a 64-byte block of memory in your RV32E processor's memory map and map it to a specific address range (for example, 0x0000-0x003F).

You could then define a set of pseudo-instructions in your higher-level assembler that map to the relevant registers in the virtual zero page. For example, you could define a pseudo-instruction like "LDAZP" (Load Accumulator from Zero Page) that maps to a load instruction with a source operand in the virtual zero page. Similarly, you could define other pseudo-instructions like "STA", "LDX", "STX", etc. that map to relevant instructions with operands in the virtual zero page.

By using this approach, you could provide a uniform interface for accessing the virtual zero page across all the MCUs you mentioned, and simplify the task of writing code that works across multiple architectures.
