Question Set 0x05

HQS Challenge

January 20, 2016

For these use the Intel syntax to represent the instruction structure

1. Explain the `mul` instruction.
2. Explain the `add` instruction.
3. Explain the `idiv` instruction.
4. Explain the `call` instruction.
5. Explain the `xor` instruction.
6. Explain the `scas` instruction.

Solutions

1. `mul arg` - Unsigned multiplication of the argument and the value in eax storing the result in eax. In the case of a 64-bit result, (`mul eax`), the value is the concatenation of the registers edx and eax (`edx:eax`)
2. `add dest, src` - Takes the source value and adds it to the destination value, storing the result in the dest register
3. `idiv arg` - Performs signed division
4. `call dest` - Pushes return address to stack and then jumps to the address in dest
5. `xor dest, src` - Performs an exclusive or operation on the destination and source values and stores the result in the destination register
6. `scas[b,w,d]` - Compares the specified length value (byte, word, dword) in eax with the value in edi and sets the flags accordingly.