Assignment 3: 8086

Rathour Param Jitendrakumar
190070049
EE-309, MicroProcessors
March 31, 2021

My implementation assumes direction flag is set

1. Stack Pointer contains 0xDF20 initially. Value in Stack Pointer reduces by 2 after each PUSH instruction (as a word is pushed into Stack)
    a. s2 is pushed. Now, SP = 0xDF1E
    b. s1 is pushed. Now, SP = 0xDF1C
    c. n is pushed. Now, SP = 0xDF1A
    d. Stack pointer updated due to function call. Now, SP = 0xDF18
    e. BX is pushed. Now, SP = 0xDF16
    f. CX is pushed. Now, SP = 0xDF14
    g. DX is pushed. Now, SP = 0xDF12
    h. SS is pushed. Now, SP = 0xDF10
    i. BP is pushed. Now, SP = 0xDF0E
    j. SI is pushed. Now, SP = 0xDF0C
    k. DI is pushed. Now, SP = 0xDF0A
    l. DS is pushed. Now, SP = 0xDF08
    m. ES is pushed. Now, SP = 0xDF06

    Now copying begins.
2. See .asm file for code

    As my function involves only CX, SI, DI, only those values need to be preserved, but I have pushed all registers one by one anyway.

    Nine registers and function call are pushed in the stack. So, to access n, copy SP into BP, then BP + 20 gives n as there 9+1=10 more additions to stack after n(Access these by BP, BP+ 2,...,BP + 18), and each addition uses the memory of a word.

    Now, s1 was added before n so, BP + 22 gives s1 and similarly, BP + 24 gives s2.
3. RET n where n is twice the number of arguments allows us to remove the arguments from the stack. So in my case, RET 6 removes input parameters.

    When the number of arguments is variable, one way is to use separate register/memory to store the number of arguments. But RET supports expressions that evaluate to constant in compile time. This is not possible with registers (don't contain same value every time).