U18CO018

Shubham Shekhaliya

MIT

Assignment – 7

1-> Write a program to convert a given number of binary data bytes into their BCD equivalents, and store them as unpacked BCDs in the output buffer. The number of data bytes is specified in register D in the main program. The converted numbers should be stored in groups of three consecutive memory locations. If the number is not large enough to occupy all three locations, Zeros should be loaded in those locations.

**Code:-**

**lxi sp,8000H**

**mvi d,0CCH ;binary data**

**mov a,d**

**call utility**

**hlt**

**utility: lxi h,2000H**

**mvi b,64H ;first for 100's**

**call help**

**mvi b,0AH ;for 10's**

**call help**

**mov m,a ;remaing 1's**

**ret**

**; calculate the multiplier**

**help: mvi m,00H**

**next: inr m**

**sub b**

**jnc next**

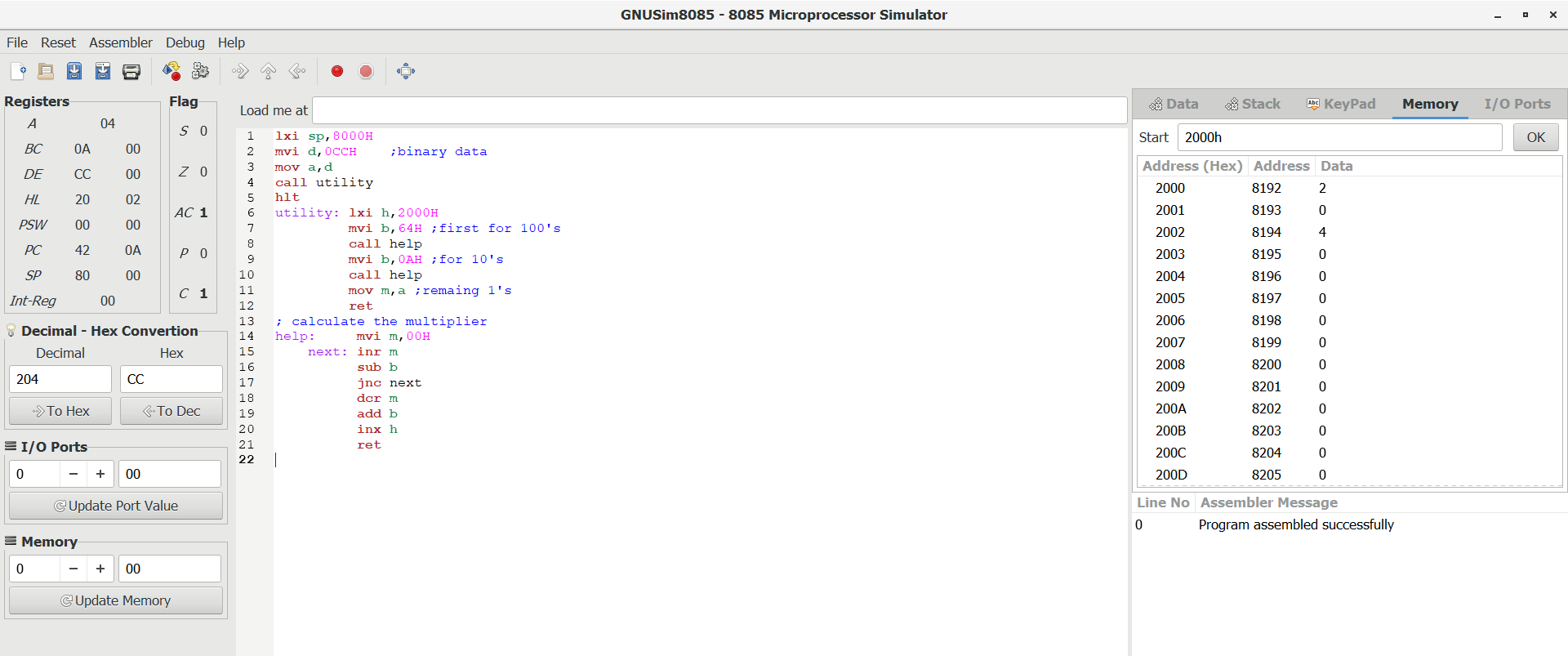
**dcr m**

**add b**

**inx h**

**ret**

**Output:-**

****

2-> A set of ten BCD readings is stored in the Input Buffer. Convert the numbers into binary and add the numbers. Store the sum in the Output Buffer, the sum can be larger than FFH.

**Code:-**

**lxi sp,8000H ;stack pointe**

**lxi h,2000H ;input buffer**

**lxi b,200AH ; output buffer**

**mvi e,00H**

**mvi d,0AH**

**next: mov a,m**

**call help**

**stax b**

**add e ; add value**

**mov e,a**

**jnc go**

**lda 2014H**

**inr a ; if carry than increment carry**

**sta 2014H**

**go: inx h**

**inx b**

**dcr d**

**jnz next**

**sta 2015H ;store the result**

**hlt**

**help: push b ;convert into the BCD to binary argument and return in A**

**push d**

**mov b,a**

**ani 00FH ; lower order in c**

**mov c,a**

**mov a,b**

**ani 0F0H ;higer order**

**jz utility**

**rrc**

**rrc**

**rrc ; rotate 4 times to get the value**

**rrc**

**mov d,a**

**xra a**

**mvi e,0AH ; 10\*higher order**

**sum: add e**

**dcr d**

**jnz sum**

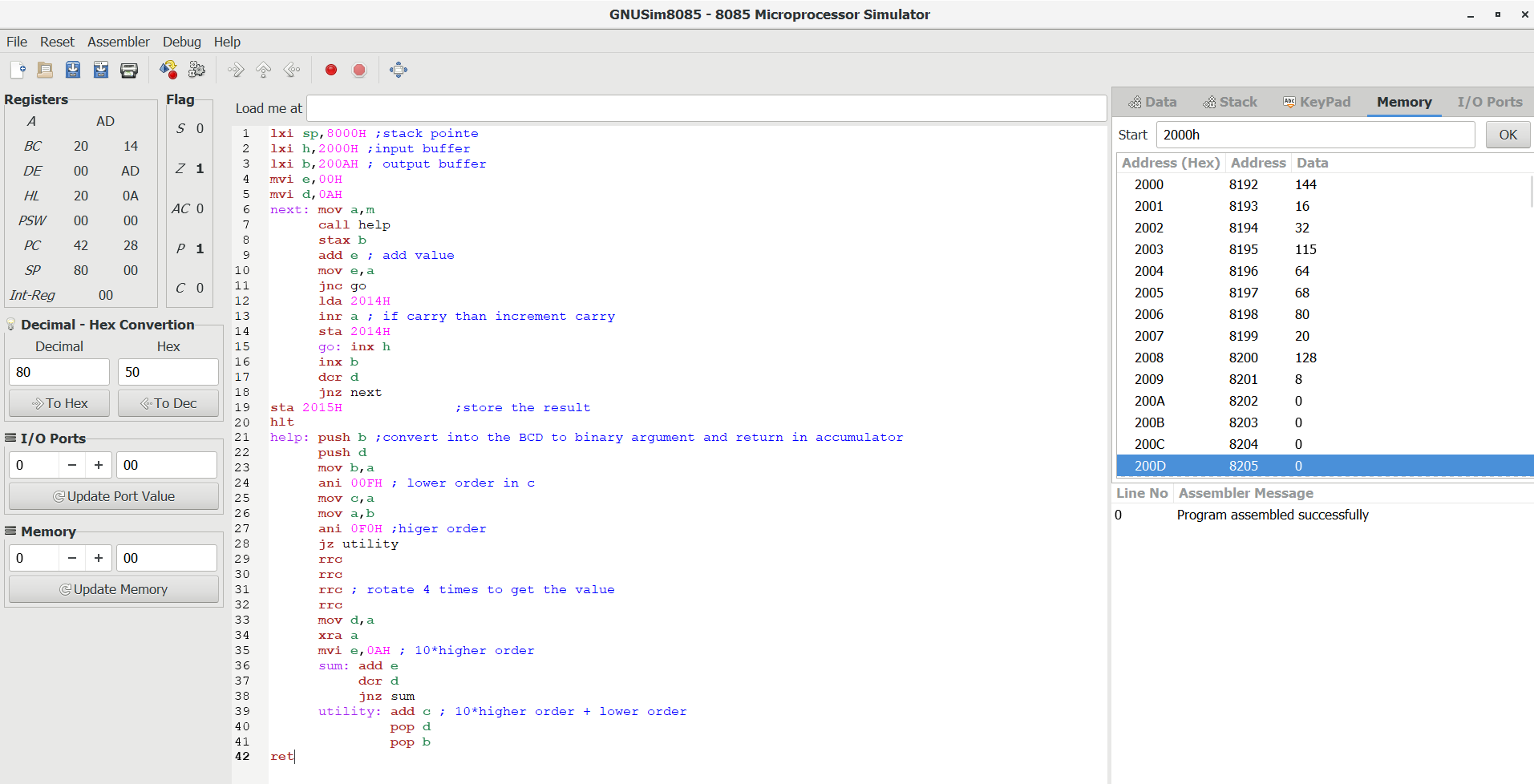
**utility: add c ; 10\*higher order + lower order**

**pop d**

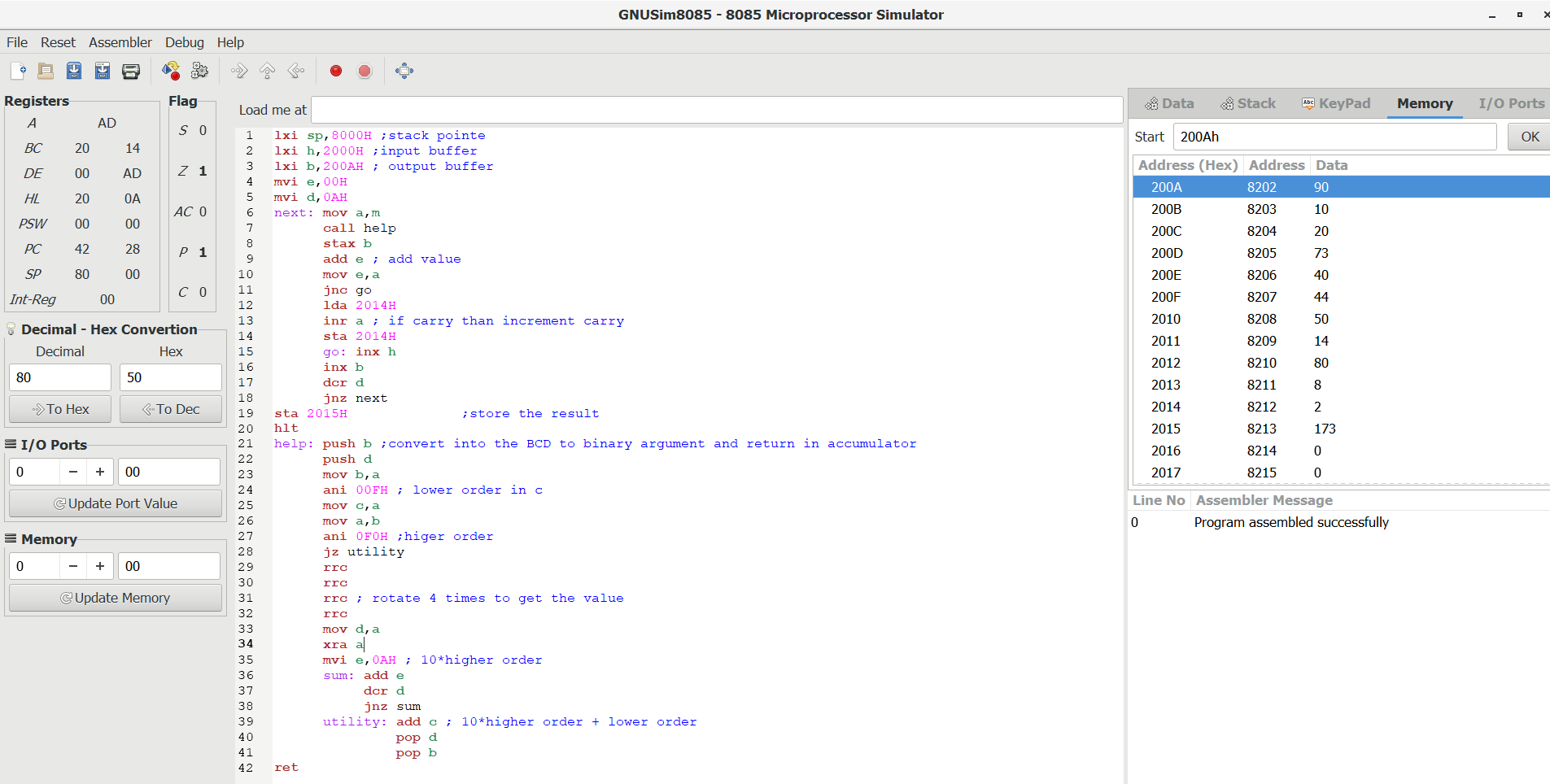
**pop b**

**ret**

**Before Executing :-**

****

**After Executing :-**

****

3-> A set of ASCII Hex digits is stored in the Input Buffer memory. Write a program to convert these numbers into binary. Add these numbers in binary, and store the result in the Output-Buffer memory.

**Code:-**

**lxi h,2000H ;input buffer**

**lxi b,200AH ;output buffer**

**mvi d,0AH ;Number of element**

**xra a ;clear cy and accumulator**

**mvi e,00H**

**next: mov a,m**

**call help ; convert Hex Ascii into the binart**

**stax b ; store into the output**

**add e**

**mov e,a ;do addition**

**inx b**

**inx h**

**dcr d**

**jnz next ;run loop till 0**

**stax b ; store sum of the number**

**hlt**

**help: sui 30H**

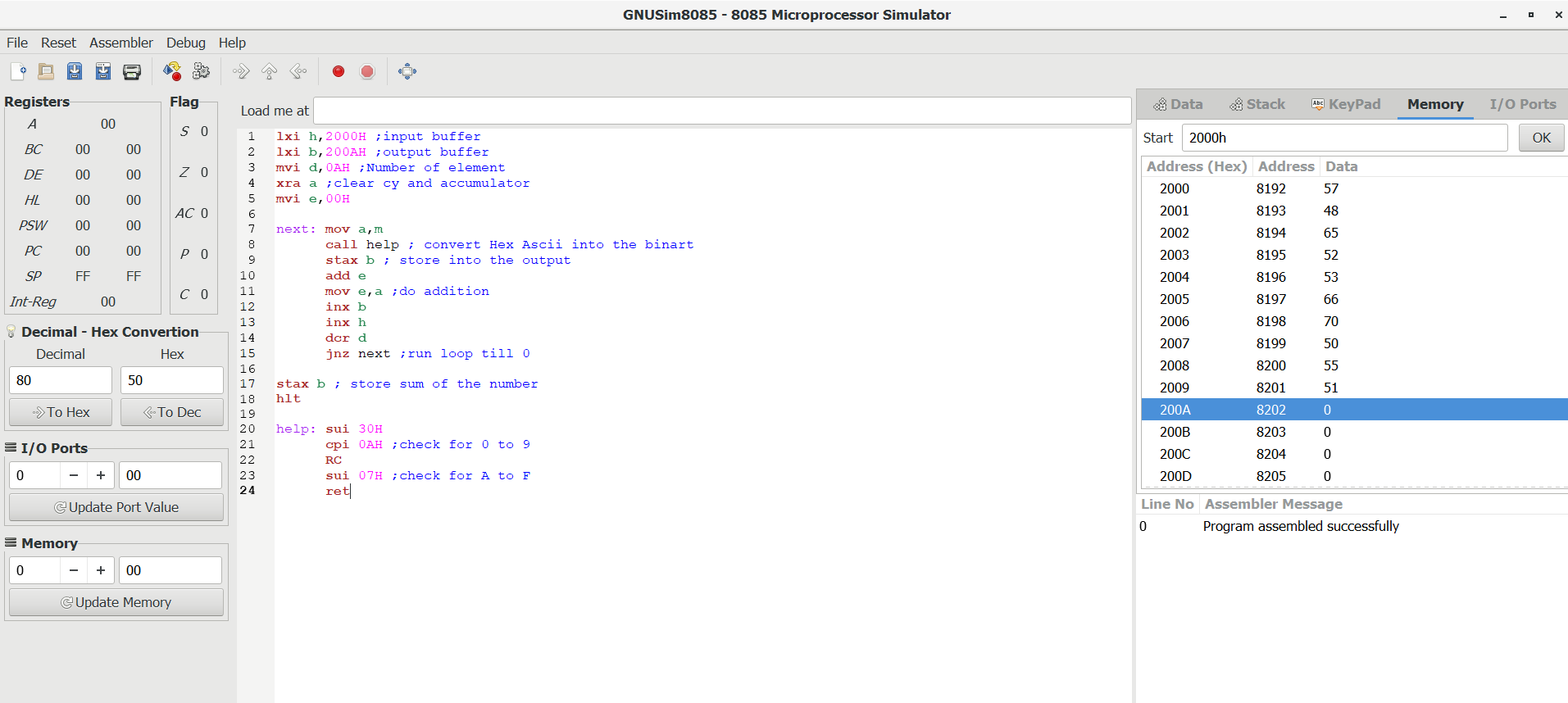
**cpi 0AH ;check for 0 to 9**

**RC**

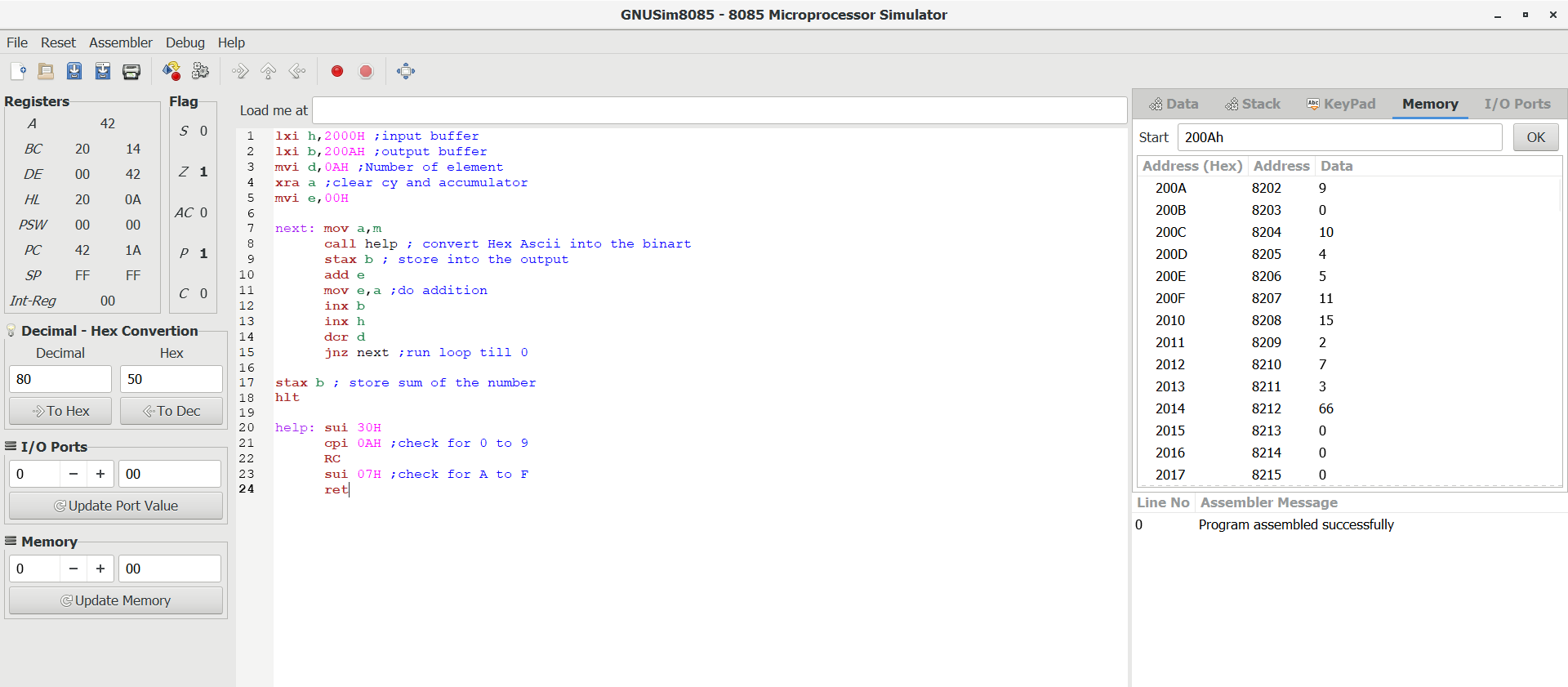
**sui 07H ;check for A to F**

**ret**

**Before Executing :-**

****

**After Executing :-**

****