EXAMPLE 8

OBJECTIVE
This exercise has the following objectives:

- Introduce decrease (SUBA)
- Review decimal borrow concepts using decrease
- Introduce hex borrow using decrease
- Introduce 2’s complement negative hex numbers as decrease of zero with free borrow
- Show the 2-digit 2’s complement negative hex numbers
- Show the decimal equivalent of the 2-digit hex decrease of zero and of 2-digit 2’s complement negative hex numbers

Note: To facilitate understanding, the same numbers are used in both decimal and the hex cases.

PROGRAM
The program ex8.asm performs the following arithmetic operations:

i) \( 7 - 1 = 6 \) (decimal) using the opcode mnemonics LDAA, SUBA
ii) \( \$07 - \$01 = \$06 \) (hex) using the opcode mnemonics LDAA, SUBA
iii) \( 10 - 1 = 9 \) (decimal) using the opcode mnemonics LDAA, SUBA
iv) \( \$10 - \$01 = \$0f \) (hex) using the opcode mnemonics LDAA, SUBA
v) \( \$00 - \$01 = \$ff \) (hex) using the opcode mnemonics LDAA, SUBA
vi) \( 0 - 1 = 255 \) (decimal equivalent of 2-digit hex arithmetic) using the opcode mnemonics LDAA, SUBA

CODING AND EXECUTION
Open THRSim11. Maximize THRSim11 window. Close the Commands window. Open ex8.asm. Assemble, tile the windows, set the breakpoint, and reset the registers. Your screen looks like this:
i) **SECTION: 7 – 1 = 6 (DECIMAL) USING THE OPCODE MNEMONICS LDAA, DECA**

Set accA display to decimal. Step through section i) of the code. At the end of this section, your screen looks like this:

During the execution of this section, decimal number 7 was loaded into accA, and decreased by 1 to become 6. During the decrease, decimal borrow did not need to happen.
ii) **SECTION: $07 - $01 = $06 (HEX) USING THE OPCODE MNEMONICS LDAA, DECA**

Set accA display to hex. Step through section ii) of the code. At the end of this section, your screen looks like this:

During the execution of this section, hex number $07 was loaded into accA, and decreased by 1 to become $06. During the decrease, hex borrow did not need to happen.
iii) **SECTION: 10 - 1 = 9 (DECIMAL) USING THE OPCODE MNEMONICS LDAA, DECA**

Set accA display to decimal. Step through section iii) of the code. At the end of this section, your screen looks like this:

During the execution of this section, decimal number 10 was loaded into accA, and decreased by 1 to become 9. During the decrease, decimal borrow happened because the LSD was 0.
iv) **SECTION: $10 - $01 = $0F (HEX) USING THE OPCODE MNEMONICS LDAA, DECA**

Set accA display to hex. Step through section vi) of the code. At the end of this section, your screen looks like this:

During the execution of this section, hex number $10 was loaded into accA, and decreased by 1 to become $0f. During the decrease, hex borrow happened because the LSD was 0.
v) **SECTION: $00 - $01 = $FF (HEX) USING THE OPCODE MNEMONICS LDAA, DECA**

Set accA display to hex. Step through section v) of the code. At the end of this section, your screen looks like this:

During the execution of this section, hex number $00 was loaded into accA, and decreased by 1 to become $ff. During the decrease, hex borrow happened because the LSD was 0. Since MSD was also 0, a ‘free borrow’ beyond the MSD happened. Hence, the result of $00-$01 is the same as the result of $000+$100-$001=$0ff, i.e. $ff.
vi) **SECTION: 0 - 1 = 255 (DECIMAL EQUIVALENT OF 8-BIT HEX ARITHMETIC) USING THE OPCODE MNEMONICS LDAA, DECA**

Set accA display to decimal. Step through section vi) of the code. At the end of this section, your screen looks like this:

During the execution of this section, decimal number 0 was loaded into accA, and decreased by 1 to become 255. We notice that the operation is equivalent to 0+256-1=255. This indicates that, although the operation was displayed in decimal, a hex free borrow actually happened, since 256=$100.

vii) **USING CALCULATOR TO OBTAIN HEX NEGATIVE NUMBERS FOR –100**

Use your calculator to find the HEX equivalent of negative decimal number –100. Type –100 into your calculator, and convert to hex. After you have done this, you will see hex number F…FFF9C in your screen. (The number of F depends on the maximum display length of your calculator.) The last two digits, 9C, are the hex negative number of –100.
WHAT YOU HAVE LEARNED
In this exercise, you have learned:

- In decimal arithmetic, decrease results in the value of the number being reduced by one unit.
- In hex arithmetic, decrease has the same meaning as in decimal arithmetic: the value of the number is being reduced by one unit.
- For a 2-digit decimal number with the LSD zero, e.g., 10, decrease required a borrow to be performed from the MSD into the LSD. As a result, we could perform the decrease and the result was 9. The result showed that, because of the borrow, the MSD decreased from 1 to 0, and the LSD increase from 0 to 9, i.e., from the bottom of the decimal symbols sequence to the top of the sequence.
- For a 2-digit hex number with the LSD zero, e.g., $10, decrease also required a borrow to be performed from the MSD into the LSD. As a result, we could perform the decrease and the result was $0f. The result showed that, because of the borrow, the MSD decreased from 1 to 0, and the LSD increase from 0 to f, i.e., from the bottom of the hex symbols sequence to the top of the sequence.
- The difference between 10 (ten) and $10 (hex one-zero = decimal sixteen).
- For a 2-digit hex number with zero LSD but also zero MSD (i.e., $00), decrease is still possible using the ‘free-borrow’ concept. The free-borrow concept, which is implemented on most microcontrollers, allows us to freely borrow a value of one from the next higher digit beyond the precision of the machine. In other word, if we extend the range to 3 hex digits and write $ff as $0ff, incrementing by one yields $0ff + $001 = $100. Conversely, decreasing $100 by one yields $0ff, i.e., $ff. This leads directly to the concept of 2's complement negative numbers.
- 2-digit 2's complement negative hex numbers are the numbers that appear to the left of the hex zero in the following sequence: …, $fd, $fe, $ff, $00, $01, $02, $03, …. We notice that:
  - $ff is the negative of $01, and $ff + $01 = $00
  - $fe is the negative of $02, and $fe + $02 = $00
  - $fd is the negative of $03, and $fd + $031 = $00
  - etc.
- New words and notations: decrease, free borrow, 2's complement negative numbers.
(This page is left intentionally blank)