Goal: Learning how to implement 64x64 bit integer multiplication on a 32-bit cpu.

Implementing 64x64 integer multiplication in assembly

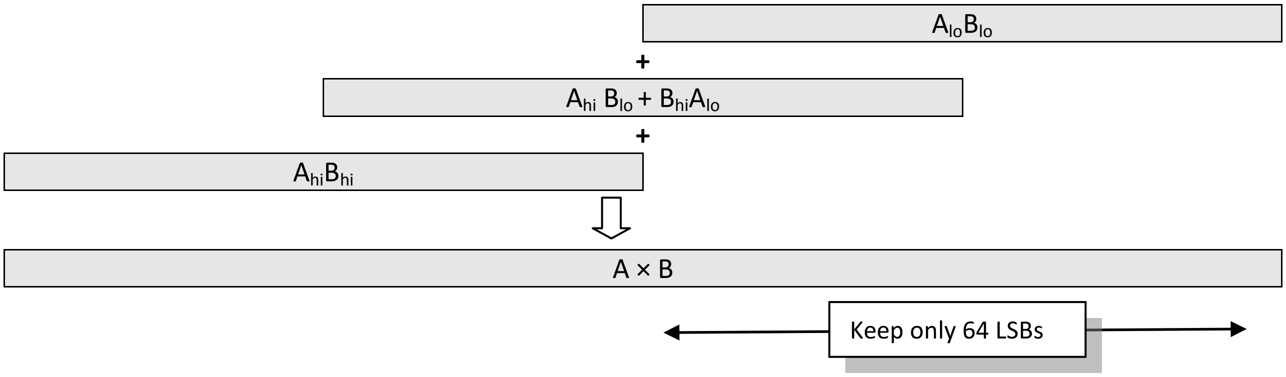
Objective: To replace the C functions defined in Integer64x64.c by equivalent assembly code in Integer64x64.s.

Background: Most 32-bit CPUs can multiply two 32-bit integers in a single instruction, but 64x64-bit multiplication must be implemented as a sequence of instructions or as a subroutine. When you multiply two 64-bit integers in C, the compiler generates code that calls a library function to do the multiplication. The purpose of this assignment is write our own version of that routine in assembly.

Approach: Multiplying two 64-bit unsigned integers on a 32-bit CPU requires breaking each number into two 32-bit halves, i.e.: $N = 2^{32} \times N_{hi} + N_{lo}$, where N_{hi} and N_{lo} are respectively the most and least significant 32-bit halves. Thus:

$$A \times B = (2^{32}A_{hi} + A_{lo}) \times (2^{32}B_{hi} + B_{lo}) = 2^{64}A_{hi}B_{hi} + 2^{32}(A_{hi}\,B_{lo} + B_{hi}A_{lo}) + A_{lo}B_{lo}$$

Each partial product term requires computing the 64-bit product of two 32-bit numbers. After shifting left 0, 32, or 64 bits according to the corresponding power of 2 scale factor, these partial products are then added to compute the final 128-bit product.



Multiplication of 64-bit integers in C keeps only the least significant 64 bits of the product, so one entire partial product term and the most significant half of two others are not needed. Note that one algorithm works for both signed and unsigned multiplication, since the only difference is in the discarded most-significant half of the product.

PART I: Preparation

1. Find the folder called “Lab Assignments” on your desktop and open it.
2. Find a double click on the file called “COEN 20.eww” to open the IAR Embedded Workbench program.
3. If step 2 did not open IAR Embedded Workbench, find the program on the Start Menu and open it. Once open, click on “File > Open > Workspace”, navigate to the “COEN 20.eww” file inside the “Lab Assignments” folder and open it.

PART II: Multiplication (using the code provided in file Integer64x64.c)

1. In the Workspace panel on the left side of the screen, right-click on “Lab09Integer64Multiplication” and select “Set As Active”.
2. Click on the “+” sign next to “Lab09Integer64Multiplication”.
3. Click on the “+” sign next to “Source”.
4. Set each of the following source code files as follows:

a.	Integer64x64.c	Include (do NOT check)
b.	Integer64x64.s	exclude (CHECK)
c.	main.c	Include (do NOT check)

5. To compile the program, right-click on “Lab09Integer64Multiplication” and select “Rebuild All”. The build should complete with no errors or warnings.
6. Connect the LM3S811 board to a USB port on your computer. This provides both power (as indicated by the power LED) and a download connection to the device. From the drop-down menus, select “Project > Download > Download active application” to start the download.
7. To run the program, press the reset button (the one closest to the thumbwheel) on the LM3S811 evaluation board. Press the other button to sequence through all the test cases. Verify that it behaves as expected.

PART III: Multiplication (implementing your ARM code in file Integer64x64.s)

1. Change each of the following source code files as follows:

a.	Integer64x64.c	exclude (CHECK)
b.	Integer64x64.s	Include (do NOT check)

2. Edit Integer64x64.s to implement the algorithm described in Integer64x64.c.
3. Recompile the program, download it to the board, and verify that it works properly.
4. Demonstrate your working program and submit your source code of Integer64x64.s to the teaching assistant according to his instructions.