

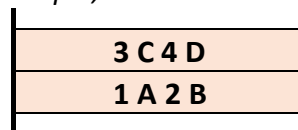
Overview

Software integer multiplication (and division) can be implemented using *shifting* and *addition*. An efficient extended multiplication algorithm is described in the class notes:

<http://users.cis.fiu.edu/~pestaina/Multiplication.pdf>

The term *extended* means that the multiplication algorithm allows for a double-word product, but single-word multiplier and multiplicand.

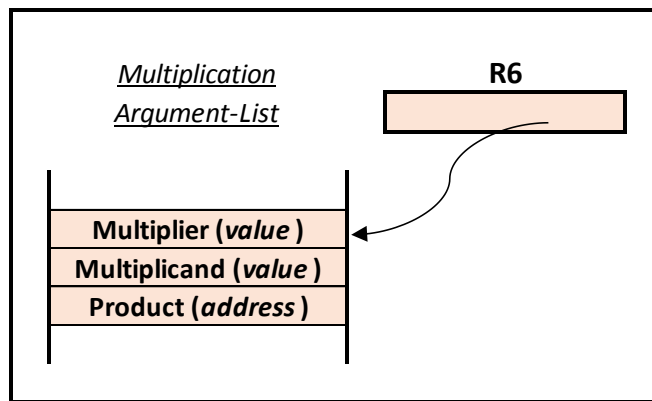
Note: A double-word is stored in 2 contiguous words; the least significant bits are stored at the lower-address word. For example, **x1A2B 3C4D** is stored as



Specific Requirements

You will need the right shift subroutine from the previous assignment. You can copy it into the start-up code (make sure it works correctly).

- 1) Implement the extended multiplication subroutine described in the class notes (link above). The parameter-passing mechanism is an argument list located by **R6**. The IN-parameters are passed *by value*; the OUT-parameters are passed *by address*. Prior to return, the subroutine must store its OUT-parameter value at the address in the argument-list.



- 3) Your subroutines must **save/restore** their working registers in separate save-areas.
- 4) Complete the main to test your XMULT subroutine by calling it twice as indicated by the comments, once to calculate A x B, next to calculate X x Y.
- 5) Include a **Program Identification Paragraph** at the beginning of your program.
- 6) Upload your **.asm** source code in SCIS Moodle by the due date.