

ELEC 2220 Computer Systems
Homework #8
Due: Friday, June 11

In RAM, define two 12-byte arrays, ARY1 and ARY2, initializing each with 12 unique data values (use the DC.B pseudo-op), and with different values in the two arrays. Write and test four versions of a program to copy ARY1 to ARY2, with the versions differing as follows.

1. Copy one byte at a time, using indexed addressing, but without using the “auto-increment” feature.
2. Copy one byte at a time, using indexed addressing with “auto-increment”.
3. Copy one word (16 bits) at a time, using either form of indexed addressing.
4. Copy one byte at a time, but reverse the order of the bytes within the array (first byte should be last after the move, etc.) Use either form of indexed addressing.

You must use a “loop” in your program that will be executed N times if copying N bytes (or N/2 times if copying words). **Do not** simply write 12 pairs of load/store instructions. The program should be able to be modified to copy arrays of arbitrary size by simply changing N.

Print and turn in your four assembly language source programs, each with the corresponding “debug” screen captured from *CodeWarrior*, showing the final results of each program. In the debug screen, highlight the two data arrays in the Memory window so we can see that the first array was correctly copied to the second.

Suggestions:

- Review how “constant” bytes are defined in memory using the “dc.b” directive.
- Create a “loop counter”, either in a register or a memory variable. Before entering the loop, this counter should be initialized to N, where N is the number of loops to be executed. At the end of the loop, the count should be decremented, with the program branching back to the start of the loop if the count is non-zero, and exiting the loop if the count is zero. After decrementing the count, test the result by executing a conditional branch instruction, either “beq” (branch if zero flag = 1) or “bne” (branch if zero flag <> 1).