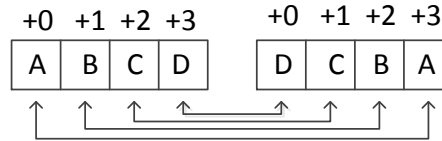


EE3220 Worksheet – 5

- Given a data array DAT1 with N 32-bit data store in BIG endian format. Write a program to store these data in array DAT2 in Little endian format.



Set-up an address pointer pointed to byte A of the Big-endian data

Set-up an address pointer pointed to byte A of the Little-endian data

Loop N times {

Load byte A to a register with post-index addressing offset of “+1” to update the address pointer.

Store byte A to memory location pointed by the Little-endian data address pointer with post-index addressing offset “-1” to update the address pointer.

Repeat these load-store actions for bytes B, C and D.

Update Little-endian data address pointer to point to byte A of the next data.

Repeat this action for N times

}

- Given 2 data arrays DAT1 and DAT2, stored with the same number of 16-bit signed numbers, write a program to find the sum of the corresponding pair of numbers if their signs are the same, i.e., compute the sum if both A_i or B_i are of the same sign. Determine the number of matched pair. Store the individual sums to a third array DAT3.

Set-up address pointer points to DAT1

Set-up address pointer points to DAT2

Set-up address pointer points to DAT3 (with 32-bit)

Set-up number counter

Matched pair counter = 0

Loop {

Load the 16-bit signed data to registers from the two arrays, with post-index addressing offset +2

Use MOVS Rn,Rn to check for sign (N = 1 if -ve number, N = 0 otherwise)

Use two registers to store the sign information for each data

Compare these two registers

Z flag = 1 if both number have the same signs, Z = 0 otherwise

Case Z=1 {

Increase the matched pair counter

Use ADD to compute the sum of both numbers

Store the sum to DAT3 array with post-index addressing offset of +4

}

Decrease number counter by 1

Repeat Loop if number counter is not “0”, END otherwise

}

END