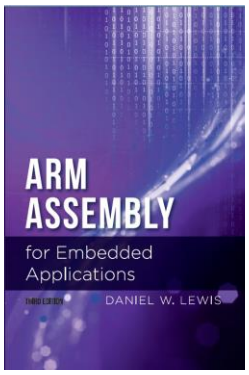


***Programming Lab #3***

# Copying Data Quickly

Prerequisite Reading: Chapters 1-4
Revised: October 10, 2017

Create five functions in ARM Cortex-M4 assembly all contained in a single source code file. Each function copies 512 bytes of data from one array to another.

Each function should use the .REPT and .ENDR directives shown in Listing 4-1 to copy the data without a loop using a straight-line sequence of instructions. The main program (download here) will display the relative execution  time of the three functions.

**`void UseLDRB(void *dst, void *src)`**

Copies **1** byte at a time using LDRB and STRB, and optimize the execution time by updating the address using the Post-Indexed addressing mode shown in Table 4-6.

**`void UseLDRH(void *dst, void *src)`**

Copies **2** bytes at a time using LDRH and STRH, and optimize the execution time by updating the address using the Post-Indexed addressing mode shown in Table 4-6.

**`void UseLDR(void *dst, void *src)`**

Copies **4** bytes at a time using LDR and STR, and optimize the execution time by updating the address using the Post-Indexed addressing mode shown in Table 4-6.

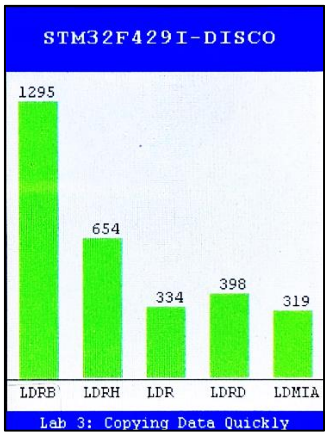
**`void UseLDRD(void *dst, void *src)`**

Copies **8** bytes at a time using LDRD and STRD, and optimize the execution time by updating the address using the Post-Indexed addressing mode shown in Table 4-6.

**`void UseLDMIA(void *dst, void *src)`**

Copies **32** bytes at a time using LDMIA and STMIA, and optimize the execution time by updating the address using the write-back flag (!) shown in Table 4-7.

If your code is correct, the display should look similar to the image at right with each function's execution time shown in clock cycles at the top of each bar graph. (Your numbers may differ.)


STM32F429I-DISCO
LDRB
LDRH
LDR
LDRD
LDMIA
Lab 3: Copying Data Quickly