

Goal: Learning how to implement Reverse Byte Order when no such instruction exists.

PART I: Background

The ability to reverse the order of bytes in a 32-bit word is a useful operation needed when converting between Little and Big Endian representation. ARM and Intel use Little Endian, while Motorola and everything on the Internet is Big Endian. While our ARM processor has an instruction (REV) that does this, many other processors do not. The purpose of this assignment is to discover how long a sequence of other instructions would have to be to accomplish what REV does in a single instruction.

Note: You are not allowed to use any of the following instructions in your solution:

BFI, BFC, SBFX, UBFX, CLZ, RBIT, REV, REV16, REVSH, SXTB, SXTB, UXTB, UXTH

PART II: Preparation

1. Download the ZIP file called "Lab Assignments.zip" from the course website on Camino.
2. Unzip the file to your desktop. This should create a folder called "Lab Assignments". Open the folder.
3. Find and double click on the file called "COEN20.eworkspace". This will open the EmBitz Integrated Development Environment (IDE) and display the projects for all the lab assignments.
4. If step 3 did not open EmBitz, find the program on the Start Menu and open it. Once EmBitz is open, click on "File > Open" in the upper left-hand corner. In the dialog window that opens, find the pull-down menu in the bottom right and select "EmBitz workspace files". Then in the middle of the dialog window, navigate to your "Lab Assignments" folder, select the file "COEN20.eworkspace", and click on "Open".

PART III: Creating Your Solution

1. Find the project (lab assignment) name in the "Management" panel on the left side of the screen. Make sure that the name is in **boldface**, which indicates that the project is Active. If not, right-click on it and select "Activate project".
2. Expand the project by clicking on the "+" sign immediately to the left of its name. Do the same for any subgroups found within it.
3. Double click on both main.c and ReverseByteOrder.s to view both files. Complete the assembly language function ReverseByteOrder using the C function RBIT as a guide.
4. To compile the program, right-click on the project name and select "Build". If there are any error or warning messages displayed, correct the source code of the function and recompile.

Hint: Function key F7 is a short-cut for "Build" for the Active project.

5. Connect the STM32F4 Discovery board to a USB port on your computer. This provides both power and a download connection to the device. To download the program to the board, click on "Debug" → "Start/stop Debug Session".

Hint: Function key F8 is a short-cut for "Debug" → "Start/stop Debug Session".

Reverse Byte Order

6. To run the program, click on “Debug” → “Run”. When the program begins to run it will display the first test case and pause. Press the left button to sequence through all the test cases. Verify that your program behaves as expected.
7. To end the debug session, click on “Debug” → Start/stop Debug Session” again.
8. Demonstrate your working program to the Teaching Assistant.
9. Upload your final version of file ReverseByteOrder.s to Camino.