Computing Machinery I
Assignment 2

Bit Reversal using Shift and Bitwise Logical Operations

Create an ARMv8 assembly language program that implements the following program:

```
#include <stdio.h>

int main()
{
  register unsigned int x, y, t1, t2, t3, t4;

  // Initialize variable
  x = 0x07FC07FC;

  // Reverse bits in the variable
  // Step 1
  t1 = (x & 0x55555555) << 1;
  t2 = (x >> 1) & 0x55555555;
  y = t1 | t2;

  // Step 2
  t1 = (y & 0x33333333) << 2;
  t2 = (y >> 2) & 0x33333333;
  y = t1 | t2;

  // Step 3
  t1 = (y & 0x0F0F0F0F) << 4;
  t2 = (y >> 4) & 0x0F0F0F0F;
  y = t1 | t2;

  // Step 4
  t1 = y << 24;
  t2 = (y & 0xFF00) << 8;
  t3 = (y >> 8) & 0xFF00;
  t4 = y >> 24;
  y = t1 | t2 | t3 | t4;

  // Print out the original and reversed variables
  printf("original: 0x%08X    reversed: 0x%08X\n", x, y);

  // Return 0 back to OS
  return 0;
}
```

Be sure to use 32-bit registers for variables declared using *int*. Use *m4* macros to name the registers to make your code more readable. Name the program *assign2a.asm*. Optimize your code so that it uses as few instructions as possible.

Also run the program in *gdb*, displaying the contents of key registers as the program executes; you should show that the algorithm is working as expected. Also print out the original and reversed values both in hexadecimal and in binary, just before the program exits. Capture the *gdb* session using the Unix command *script* and name the output file *script.txt*.

Create a second version of the program (called *assign2b.asm*) that uses 0x7F807F80 for the *x* variable. Also create a third version of the program (called *assign2c.asm*) that uses 0x01FF01FF *x* variable.

Other Requirements

Make sure your code is properly formatted into columns, is readable and fully documented, and includes identifying information at the top of each file. You must comment each line of assembly code. Your code should also be well designed: make sure it is well organized, clear, and concise.

New Skills Needed for this Assignment:

- Use of bitwise logical and shift operations
- Understanding of hexadecimal and binary numbers

Submit the following:

1. Your assembly source code files for the 3 programs, and your script output file. Use the *Assignment 2* Dropbox Folder in D2L to submit electronically. The TA will assemble and run your programs to test them. Be sure to name your programs and script file as described above.

Computing Machinery I
Assignment 2 Grading

Student:____

Functionality:		
Bit reversal step 1	3	______
Bit reversal step 2	3	______
Bit reversal step 3	3	______
Bit reversal step 4	5	______
Display to screen	3	______
Optimization	2	______
Script showing *gdb* session	3	______
Complete documentation and commenting	4	______
Formatting (use of columns and white space)	4	______
Design quality	2	______
Version 2	2	______
Version 3	2	______

Total **36** ______ _____**%**