CSE 3113 - Microprocessor and Assembly Lab
Lab Report

Tazkia Malik
Class Roll: 07

May 20, 2025

1 Write assembly language to perform a simple Boolean operation to calculate the bitwise calculation of F = W.X + NOT(Y.Z)

Build Status

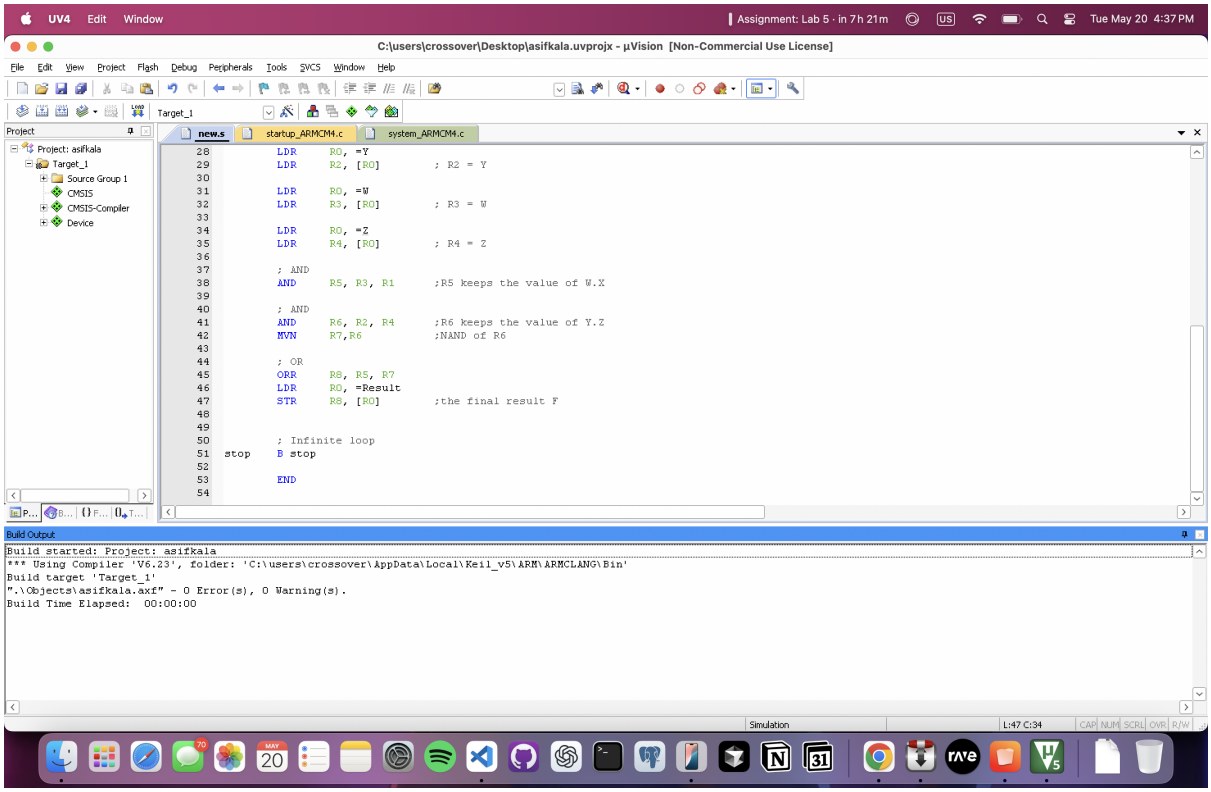

Figure 1: State of the system after the project is built.

Register Status After Execution

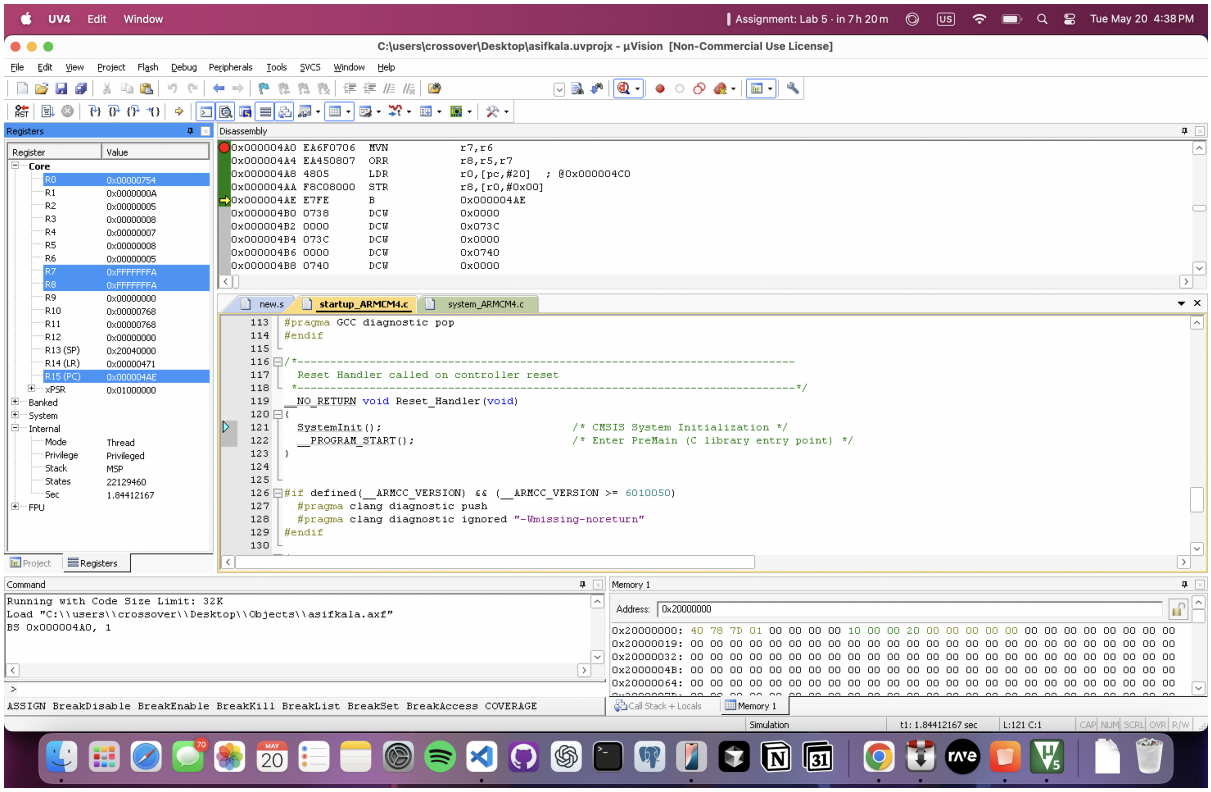

Figure 2: State of the system after the code has been executed.

2 Suppose we have three words P, Q and R. We are going to apply logical operations to subfields (bit fields) of these registers. We'll use 16-bit arithmetic for simplicity. Suppose that we have three 6-bit bit fields in P, Q, and R as illustrated below. The bit fields are in green and are not in the same position in each word. A bit field is a consecutive sequence of bits that forms a logic entity. Often they are data fields packed in a register, or they may be graphical elements in a display (a row of pixels). However, the following example demonstrates the type of operation you may have to perform on bits.

Build Status

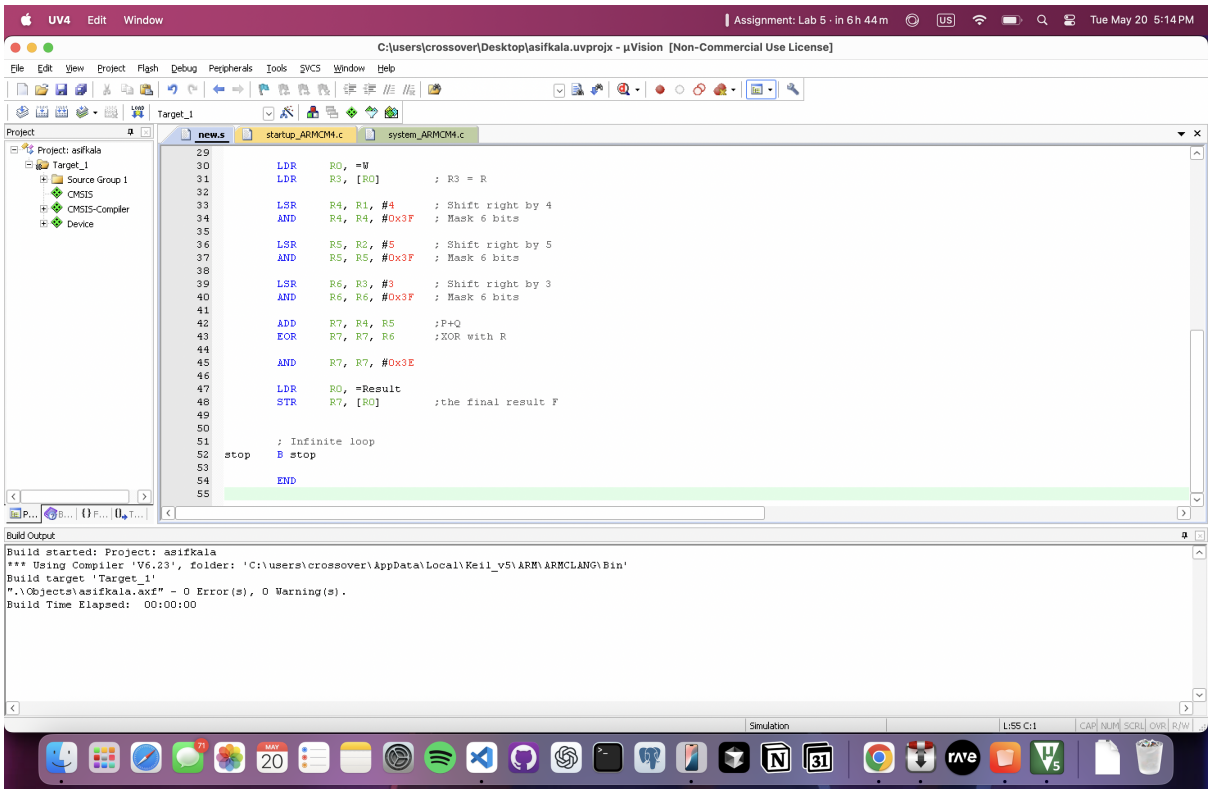

Figure 3: State of the system after the project is built.

Register Status After Execution

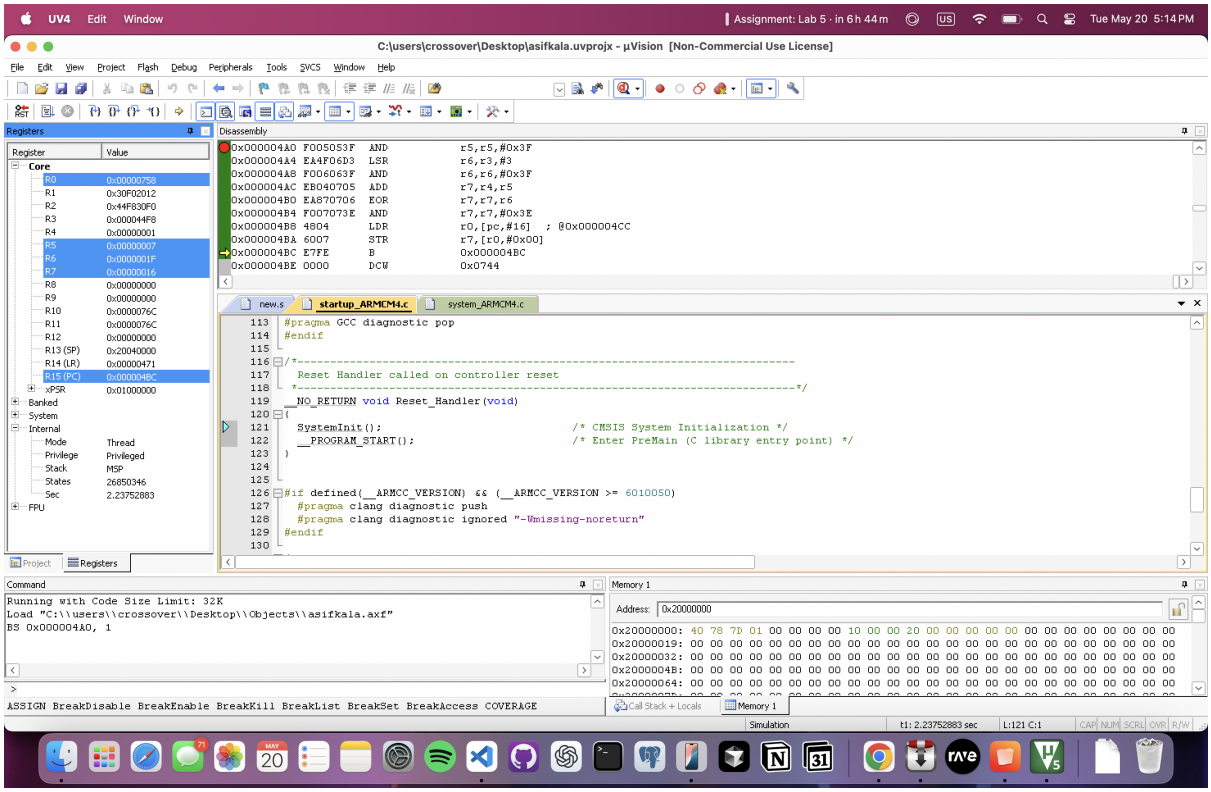

Figure 4: State of the system after the code has been executed.

3 Write an assembly language to find the one's complement of a number.

Build Status

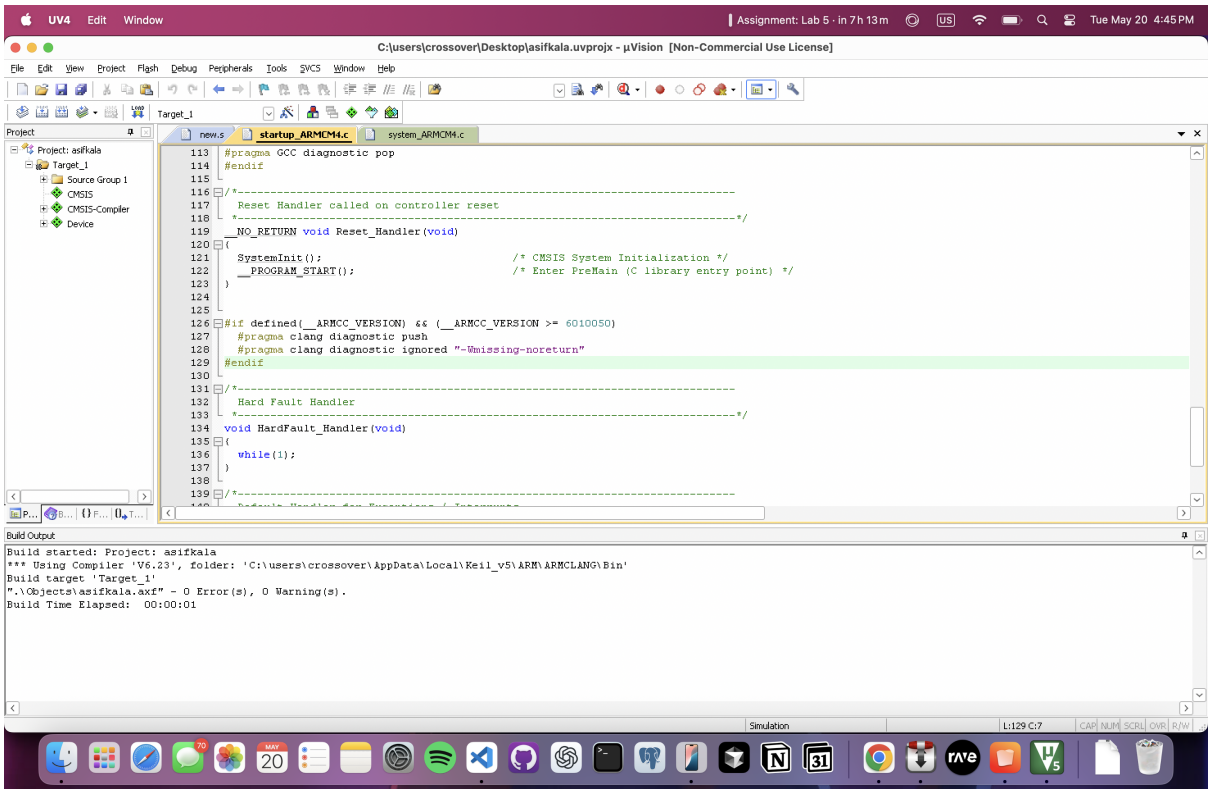

Figure 5: State of the system after the project is built.

Register Status After Execution

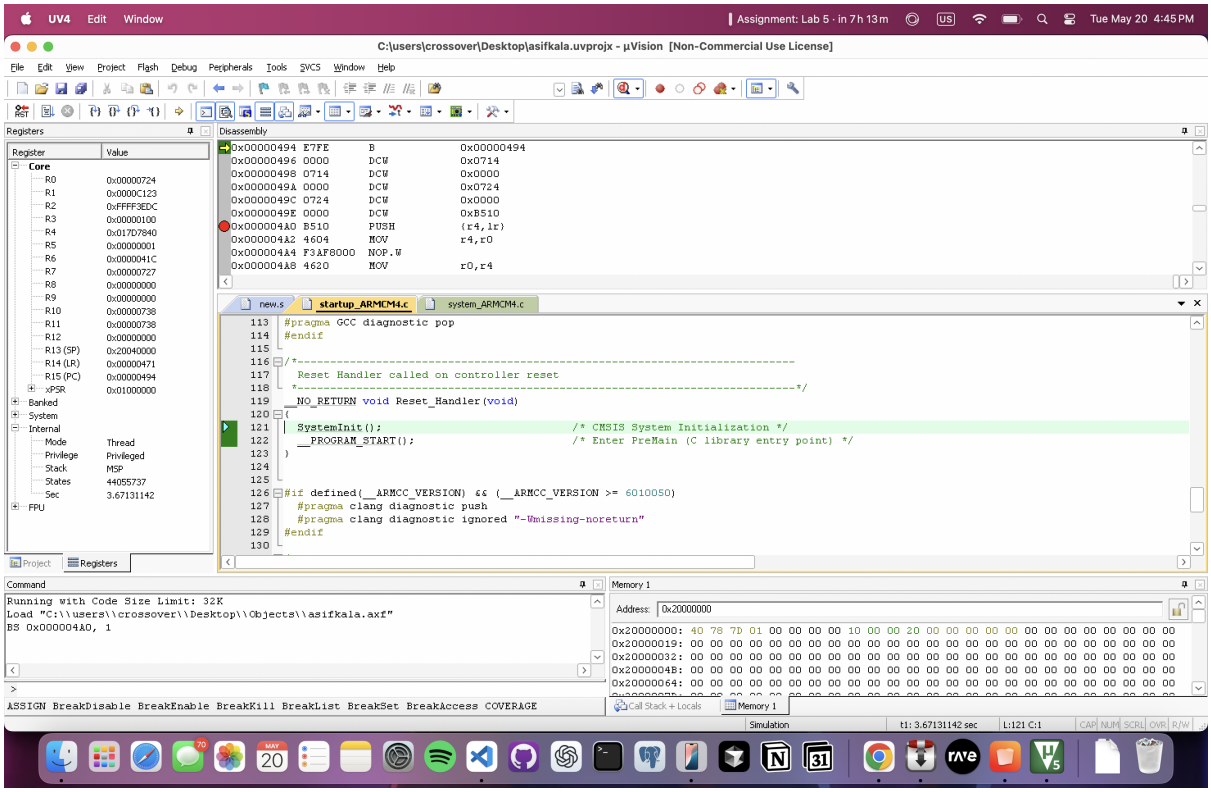

Figure 6: State of the system after the code has been executed.

4 Write an assembly language program to disassemble a byte into its high and low order nibbles. Divide the least significant byte of the 8-bit variable Value into two 4-bit nibbles and store one nibble in each byte of the 16-bit variable Result. The low-order four bits of the byte will be stored in the low-order four bits of the least significant byte of Result. The high-order four bits of the byte will be stored in the low-order four bits of the most significant byte of Result.

Build Status

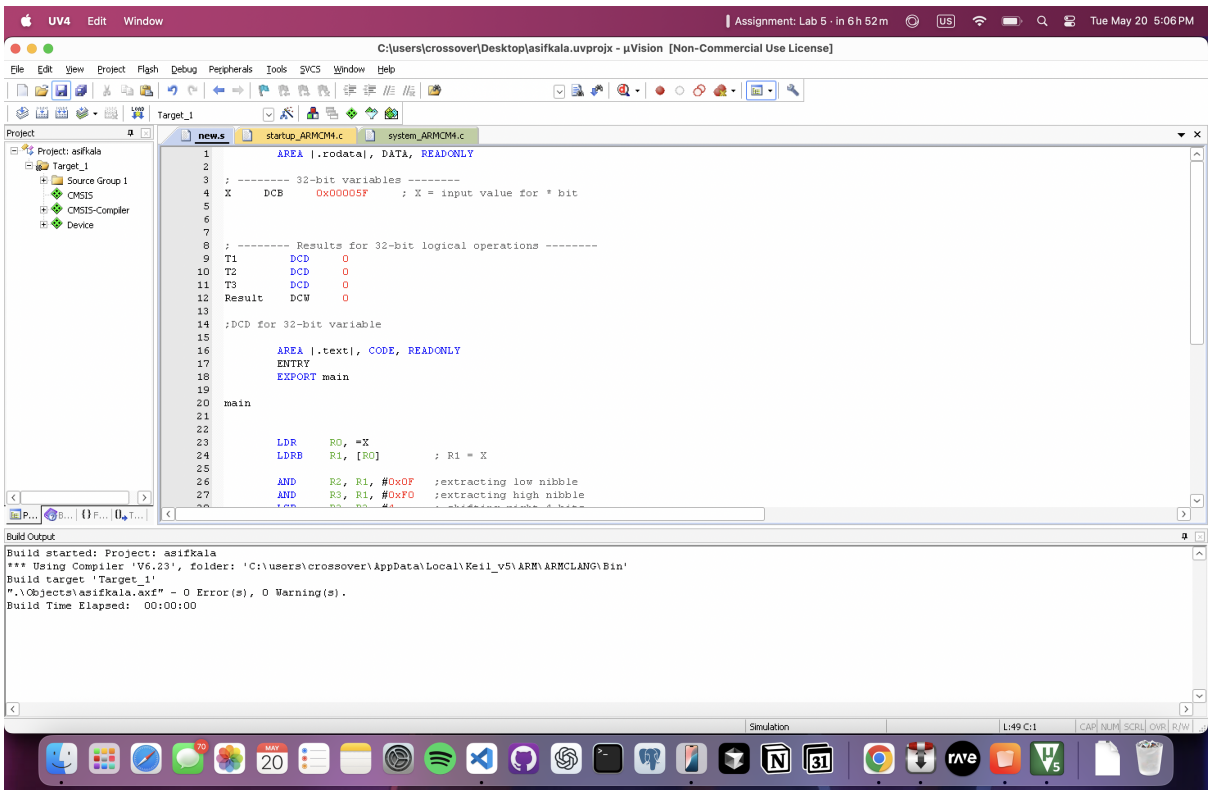

Figure 7: State of the system after the project is built.

Register Status After Execution

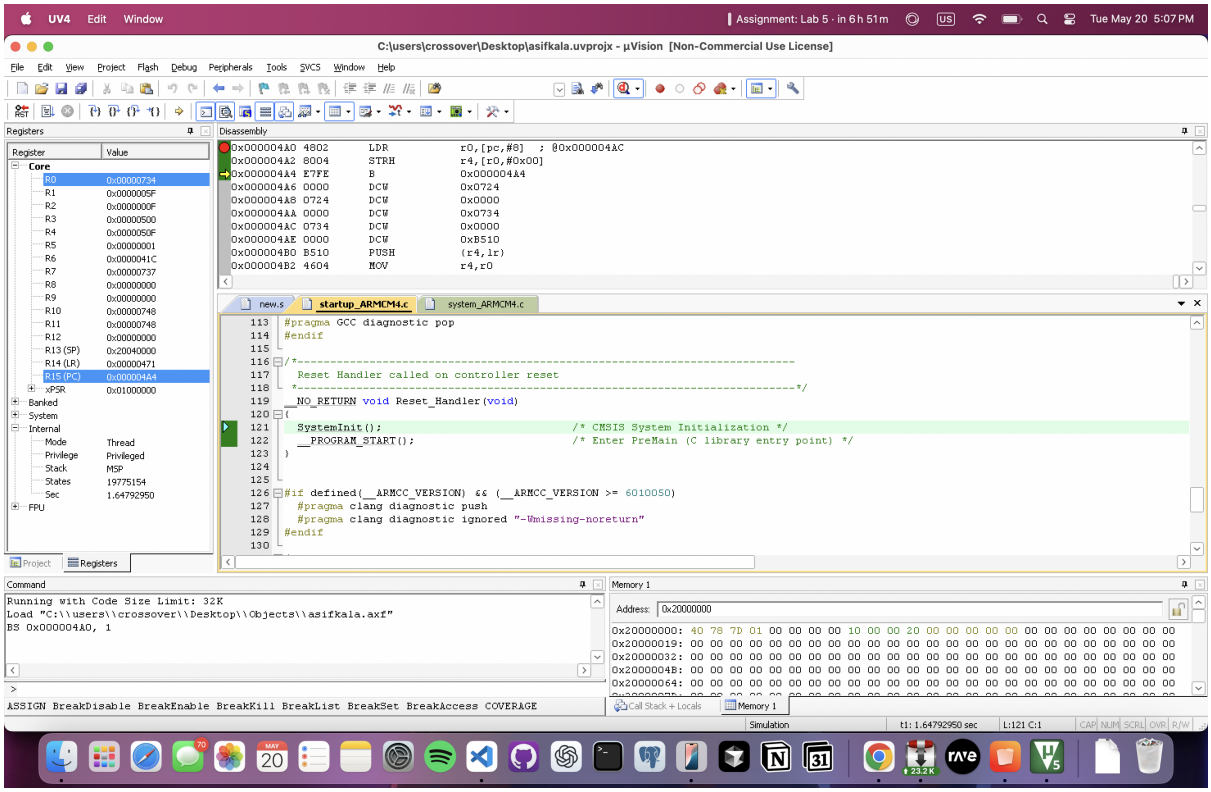

Figure 8: State of the system after the code has been executed.