Assignment Overview:

Names: Hamza El Sousi && Mansi Joshi

Student IDs: 040-982-818 && 041091664

Course: CST8216\_301 Processor Architecture

Professor: David Hailey

Group: 7

Assignment Description:

The assignment focused on implementing a dual-mode counter project using 68HCS12 Assembly Language. The goal was to create a counter that alternates between counting in Binary-Coded Decimal (BCD) and hexadecimal (HEX) formats. The assignment required a structured approach, incorporating concepts such as addressing modes, iteration, and the use of arrays.

Challenges Faced:

Hamza El Sousi:

One of the challenges I encountered was related to properly implementing the BCD counting logic. Understanding and correctly using the ‘daa’ instruction for BCD adjustment was crucial. Ensuring that the counter incremented and displayed the correct values on the HEX displays posed an initial challenge. Debugging the code in the simulator was crucial to identifying and resolving issues in the BCD counting section.

Additionally, managing the stack for temporary storage of values required careful consideration. Efficiently using the stack for pushing and pulling values while switching between BCD and HEX counting modes was a crucial aspect of the implementation. The resources provided on Brightspace was very helpful in understanding what I needed to do.

Furthermore, when asked to change the values of START\_COUNT and END\_COUNT during the demo, we realized yet another issue with how the program was count the HEX values, BCD was working fine and displaying the correct numbers while HEX display didn’t display the new values correctly.

Mansi Joshi:

My partner and I faced challenges in synchronizing the BCD and HEX counting modes to ensure a smooth transition. Coordinating the display on the HEX segments with the count value was essential for the project's success. Debugging in the simulator helped us identify synchronization issues and fine-tune the timing to achieve the desired results.

Another challenge was adapting the provided starter code and integrating the necessary library routines. Understanding how to use the API subroutines, particularly Extract\_Msb and Extract\_Lsb, posed some difficulties initially. However, referring to the API booklet and collaborating with my partner allowed us to overcome these challenges.