Lab 1 Basic Debugging with Keil: Lab Report

Student: Jeremiah Webb

Student ID: 2545328

Instructor: Dr. Jianhua Liu

Section #2

Introduction

This lab was used to analyze and understand more advanced ways to debug C programs and even compare assembly programmed functions to see the difference in speeds. Using the same algorithms in C and assembly we can see clock speed differences too. Using such performance evaluations can also help debug code and see how we can make programs overall faster and more efficient.

Tasks

Task 1

Beginning Clock Cycle:

Text, letter

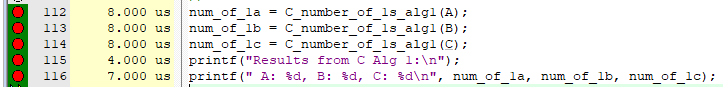
Description automatically generated

Ending Clock Cycles:

Text

Description automatically generated

Complete Difference: 5362 clocks

Total Time spent: 35 us

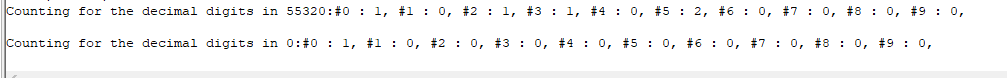
Task 2

Table

Description automatically generated   
Lower Clock amounts means faster.

As shown, Assembly is significantly faster, for example between Alg2, C and Alg2,asm, Alg2,asm is 190% faster than its C counterpart. Overall, Assembly is faster, even in small algorithms.

Task 4



Results

Text

Description automatically generated

Code

Code Snippets

Task 4

void C\_number\_of\_0\_to\_9s(uint32\_t x, uint32\_t result\_arr[]) {

for(uint32\_t i = 0; i < 10; i++){

result\_arr[i] = 0;

}

if(x == 0){

result\_arr[0] = 1;

}

else{

while (x != 0){

uint32\_t digit = x % 10;

result\_arr[digit]++;

x = x / 10;

}

}

for(uint32\_t i = 0; i < 10; i++){

printf("#%d : %d, ", i, result\_arr[i]);

}

printf("\n");

}

Narrative

The Lab did go well, seeing the difference between Assembly and C is a good insight into how our future labs when programming Assembly. Visualizing the speed difference is also a good insight into why programmers even program in Assembly, than a higher language like C.

Results

One can see that between C and Assembly, Assembly is significantly faster. In small algorithms one could argue that the difference is negligible, however in longer, more intricate functions and algorithms, Assembly is faster. Using clock speeds is additionally a great way to debug and understand how to make algorithms more efficient, in both languages.