**Lab1 – Arrays and Functions**

ECSE-324 Computer Organization

Sen Wang (260923645)

Friday 16, October 2020

McGill University

Montreal Quebec, Canada

# Part 1\_1: Square root of an integer using the SGD technique (iterative solution)

## Brief Description:

This assembly program takes an integer as an input and calculates its square root using the SGD technique. For our program, we used 100 iterations. The code can be separated into 6 main parts. The first part of the program is variable declaration. Here, it used MOV to store values/constants into their respective registers. The second part is the main loop where the program computes the step variable and compares step to t. The third part is the GreaterThan branch which assign step variable to t if step is greater than t. The fourth part of the program is LesserThan branch which assign step variable to -t if step is lesser than -t. It branches to GreaterThan or LesserThan using CMP inside the loop. The fifth part of the program decrements xi by step at each loop iteration and decrements the counter for the loop. Finally, the last part is the Stop branch which calls itself recursively. It branches to Stop when we are outside of the loop.

## Approach Taken

As described in the brief description, the approach taken for this program is the use of Branch Instructions and conditional execution. For example, in the main loop, I used CMP to compare values. If step is greater than t, then the program branches to GreaterThan using BGT. Similarly, if step is smaller than -t, the program branches to LesserThan using BLT. To implement the loop logic, the program uses BGT to loop after we decrement the loop counter. More precisely, the loop counter starts and 100 and decrements by 1 at each iteration. If the counter is greater than 0, then the program branches to Loop.

## Challenges Faced and Solutions

Since the first part of the lab is relatively simple compared to the other parts, no particular challenges faces. The only difficulty I faced for this first part of the lab would be the use of conditional execution. At first, I couldn’t really figure out how the comparison worked. The issue was quickly solved after I revised the notes and after some internet searches.

## Possible Improvement to the Program

This program works but several improvements to the program could be made. For example, this program used a lot of registers. Some registers are only used as temporary variables to store intermediate calculation values. Reusing some registers can then reduce the total number of registers. Another improvement would be the use of less branches. In this program, I used an additional branch to just decrement the values.