Integer Overflow and Floating Point:

Overflow and Point----------------------------------------------------

What is Integer Overflow:

Occurs when an addition operation causes a number to fall outside of the range that a certain variable can hold.

Ex: If you can only hold 32 bits, but you’re operation leaves you with an answer that is more than 32 bits, than there will be an overflow.

Certain types such as int, float, double, and so on can only hold a certain amount of digits and this capacity of these numbers can be seen in this document: <https://docs.google.com/document/d/1eJj4oXB9IiNQ2gDJMOQTHeFICFuZPy-glFcpXBP0f40/edit?usp=sharing>

Examples:

#include <iostream>

using namespace std;

int main(){  
int i = 100;

i

i=i+1000000000000000000000000;

cout << i << endl;

}

This would cause an overflow because an integer is only 32 bits and its capacity is -2,147,483,648 to 2,147,483,647

In ARM, there is a way to modify your operations so that they automatically carry for you, without having to keep track of it yourself.

For example, ADD is a function used in ARM to simply add two values. But as discussed, there can be an overflow error. To combat this overflow error, we use ADCS in place of ADD. ADCS does the same thing that ADD does, but it sets a carry bit, and performs the operation with the carry bit.

The “C” in ADCS is what does the carry. If ADD adds together X and Y, ADDC adds X, Y, and the carry bit (what is leftover).

The “S” in ADCS sets the status flags. This means that the operation will set the carry status flag to 1, allowing for the program to realize it needs to keep hold of the carry information.

Example in Theory:

Adding 101 + 110 in binary

Using ADCS will give you the answer 1011 while ADD would return 011.

Example Code:

LDR r3, [r1,#4]

LDR r4, [r2, #4]

ADCS r5, r3, r4

STR r5, [r0,#4]

Floating Point Error:

Real numbers cannot be accurately represented within a fixed space and this problem can not be eliminated, rather only managed.

Contains three parts:

1. Sign bit:
2. Exponent
3. Mantissa: part of number in scientific notation or floating point number that consist of significant digits

Example:

#include <iostream>

using namespace std;

int main(){

for(float value = -2.0; value <= 2.0; value += 0.2)

std::cout << value << std::endl;

}