

COL216 Computer Architecture

Lab Assignment 1 : Evaluation of Expressions Without Parentheses

Write a program in ARM assembly language for evaluating arithmetic expressions involving integer constants and operators add (+), subtract (-) and multiply (*). The program takes an expression represented as a null terminated ASCII string. Evaluation is performed in left to right order. There is no operator precedence. Assume that the program is to work only with expressions that do not cause an overflow at any stage.

Use .asciz assembler directive to define the expression to be evaluated. For example,

```
myExp: .asciz "31+12*20-18"
```

This should evaluate to $((31+12)*20)-18 = 842$

The result is to be left in some register. Note that the constant operands will need to be converted from ASCII to binary. For example, take a constant "342". It is stored as a sequence of three bytes containing ASCII codes of '3', '4' and '2'. You need to extract the values 3, 4 and 2 and compute $((3 * 10) + 4) * 10 + 2$, to get the binary value of 342.

This assignment will help you to get started with ARM assembly language programming using ARMSim# simulator. It will also familiarize you with how the external representation of numbers differs from their internal representation.

The relevant ASCII codes are given below in hexadecimal.

Character(s)	ASCII code
'0' – '9'	30 - 39
'+'	2B
'-'	2D
'*'	2A