

FACULTY OF ELECTRICAL ENGINEERING AND
COMPUTING
DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING

Computer Architecture 1R

1st Laboratory Exercise

October 2024

1 Task

In memory at address **0x700**, there is a data array where each data item is a structure consisting of three **32-bit** numbers. At the beginning of the structure, there is a 32-bit number that specifies an arithmetic operation, as follows:

- 0 - addition
- 1 - subtraction
- 2 - multiplication
- 3 - division

Following the arithmetic operation code, there are two 32-bit numbers stored in **2's complement format**. The number of data entries in the block is not predefined, but it is known that it is terminated by the data entry **0x88888888** at the operation position within the structure. An example of part of the memory is shown in Table 1.

Table 1: Memory Representation

#	Address	Description	Data
1	0000 0700	Operation	0000 0003
	0000 0704	1st operand	FFFF FEFF
	0000 0708	2nd operand	0000 0010
2	0000 070C	Operation	0000 0001
	0000 0710	1st operand	0000 01F4
	0000 0714	2nd operand	FFFF FD44
3	0000 0718	Operation	0000 0002
	0000 071C	1st operand	FFFF FFFE
	0000 0720	2nd operand	0000 000A
4	0000 0724	Operation	0000 0003
	0000 0728	1st operand	FFFF F000
	0000 072C	2nd operand	FFFF FFC0
			8888 8888

Write a program for the ARM processor that processes all data in the block by performing the arithmetic operation specified at the beginning of each structure on the two data values in the structure. After performing the operation, the program stores the **32-bit 2's complement** result in memory starting at address **0x2000**. The resultant block should be terminated with the entry **0xFFFFFFFF**. You can assume that the result of the operation will never match the value used to

terminate the resultant block. An example of the resultant block for the data from Table 1 is shown in Table 2.

For subtraction and division operations that are not commutative, the 1st operand represents the minuend or dividend, and the 2nd operand represents the subtrahend or divisor.

Table 2: Resultant Memory Block

Address	Result
0000 2000	FFFF FFF0
0000 2004	0000 04B0
0000 2008	FFFF FFEC
0000 200C	0000 0040
0000 2010	FFFF FFFF

Write a subroutine **DIVIDE** that performs integer division of two numbers using the method of successive subtraction. The subroutine receives parameters **via the stack** and returns the result **in register R10**. Use the DIVIDE subroutine in the main program solution for the division operation of two data values in the structure. In case of division by zero, the subroutine returns 0. The multiplication operation can be implemented using mnemonics available for the ARM processor. The multiplication and division operations must preserve the sign of the data (e.g., multiplying a positive and a negative number results in a negative number). You can assume that all operations will yield a valid result within 32 bits.

2 Submission

Submit your **own** solutions by sending it by e-mail either in their original format named **lab1.a** or in a .zip archive at the e-mail address **matko.batos@fer.hr** by Friday, November 08th 2024 at 23:59.

For any questions, we are available via email.