Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

University of Zagreb

Faculty of Electrical Engineering and Computing

Arhitektura računala 1

*Computer Architecture 1*

**Laboratorijska vježba broj 1**

*Laboratory exercise no. 1*

Dora Mlinarić 0036480184

Kristina Vučenik 0036476704

# Zadatak / Exercise

In the memory on the address 50016 there`s a block of data in 64-bit format with one bit for sign (a format with a bit for sign in double precision). Lower word of the 64-bit number is located on lower memory locations and the higher word on higher locations (whole 64-bit data is written in little-endian order). Size of the block is written in first 32-bit data on address 50016. Write a program for FRISC-3 which will turn data from this block into 16-bit format of the 2-complement and write it down in the new block which starts at the address 70016. Data in first block should be: -100, +210, -410, +3710, -7310. The new block has to be closed with 16-bit value 800016.

# Rješenje / Solution

Copy your assembly code between lines. Include code required for verification of the solution. Please comment the code. Use Paste -> Keep text only when copying to preserve original formatting.

`ORG 0

MOVE 1000, SP ;stack pointer initialization

MOVE 504, R0 ;adress of the beginning of the block (without number of

; data)

MOVE 700, R1 ;adress of the destination block

MOVE LCDDATA, R5 ;adress of the location for lcd preview

LOAD R2, (R0-4) ;load the number of data in the block

LOOP LOAD R3, (R0) ;load lower 4 bytes of the number

LOAD R4, (R0+4) ; load higher 4 bytes of the number

ADD R0, 8, R0 ;increase adress pointer

SHL R4, 1, R4 ;shift left for one place to get the highest bit of the data in ;carry flag

JP\_C NEGATIVE ;if that bit is 1, the number is negative

STOREH R3, (R1) ;store the number in the destination block

STOREH R3, (R5) ;also store it on a location for lcd preview

ADD R1, 2, R1 ;increase the adress in the destination block

ADD R5, 2, R5 ;increase the lcd preview adress

JP NEXT ;skip turning the number into the negative one

NEGATIVE XOR R3, -1, R3 ;if it is negative turn it into its 2 complement

ADD R3, 1, R3

STOREH R3, (R1)

STOREH R3, (R5)

ADD R1, 2, R1

ADD R5, 2, R5

NEXT SUB R2, 1, R2 ;decrease data number counter

JP\_NZ LOOP ;if there is more data avaliable go to the beginning of the

; loop and repeat everything

MOVE 8000, R2 ;last data in the block

STOREH R2, (R1)

STOREH R2,(R5)

JP PRINTDATA ;o to the part for lcd preview

HALT

;part of the memory with defined data

`ORG 500

DW 5, 00000001, 80000000, 00000002, 00000000, 00000004, 80000000, 00000025, 00000000, 000000049, 80000000

PRINTDATA MOVE %B 001, R0

STORE R0, (%H FFFF0000) ; GPIO configuration for SW – input without int

LOAD R0, (%H FFFF0004) ; load SW state

SHL R0, 4, R0

MOVE LCDDATA, R1

ADD R0, R1, R2

MOVE 4, R3

PRINTLOOP CALL LCDWRITE

ADD R2, 4, R2

SUB R3, 1, R3

CMP R3, 0

JP\_NE PRINTLOOP

JP PRINTDATA

LCDWRITE LOAD R0, (%H FFFFF008)

CMP R0, 0

JR\_EQ LCDWRITE

MOVE 1, R0

STOREB R0, (%H FFFFF004)

LOAD R0, (LCDCURS)

ADD R0, 1, R0

STORE R0, (LCDCURS)

CMP R0, 3

JP\_EQ ROW2

CMP R0, 6

JP\_EQ ROW1

JP NOMOVE

ROW1 MOVE %B 00000001, R0

STOREB R0, (%H FFFFF005)

MOVE 0, R0

STORE R0, (LCDCURS)

JR LCDWRITE

ROW2 MOVE %B 10000001, R0

STOREB R0, (%H FFFFF005)

JR LCDWRITE

NOMOVE LOAD R0, (R2)

STORE R0, (%H FFFFF000)

RET

LCDCURS DW %H 00000000

LCDDATA `DS %D1024

# Ispitivanje rješenja / Solution Verification

Write detailed explanation how is the solution verified:

Using the LCD preview we can see the result of our program. We can check our result by looking for the correct solution in the following table:

