

Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА №2  
з дисципліни “Архітектура комп’ютерів 3”  
на тему “Основні інструкції 32-бітного ARM процесора для мікроконтролерів”

Виконала:  
студентка III курсу ФІОТ  
групи ІВ-81  
Дьяченко Тетяна

Перевірив:  
Нікольський С. С.

**Мета:** Навчитися використовувати асемблерні інструкції ядра Cortex-M4, працювати з процедурами і базово зрозуміти архітектуру ядра.

### **Завдання**

Номер залікової книжки: 8113

$8113 \% 5 = 3$

$$\begin{cases} (a + b)/c, & (a - b) \geq 0 \\ (a - b)/c, & (a - b) < 0 \end{cases}$$

### **Виконання роботи**

start.S

```
.syntax unified
.cpu cortex-m4
//.fpu softvfp
.thumb
// Global memory locations.
.global vtable
.global __hard_reset__
/*
 * vector table
 */
.type vtable, %object
.type __hard_reset__, %function
vtable:
    .word __stack_start
    .word __hard_reset__ + 1
    .size vtable, . - vtable
__hard_reset__:
// initialize stack here
// if not initialized yet
    bl lab1
    _loop: b _loop
    .size __hard_reset__, . - __hard_reset__
```

lscript.ld

MEMORY

{

FLASH ( rx )

: ORIGIN = 0x08000000, LENGTH = 1M

RAM ( rxw )

: ORIGIN = 0x20000000, LENGTH = 128K

}

\_\_stack\_start = ORIGIN(RAM) + LENGTH(RAM);

lab1.S

.global lab1

.syntax unified

#define A #11

#define B #5

#define C #2

lab1:

```
    push {lr}
    //calculate
    mov r0, A
    mov r1, B
    mov r2, C
    bl var3
    pop {pc}
```

var3:

```
    push {r0, r1, r2, lr}
    cmp r0, r1
    ITE GE
    addGE r3, r0, r1
    subLT r3, r0, r1
    sdiv r3, r2
    pop {r0, r1, r2, pc}
```

## Makefile

SDK\_PREFIX?=arm-none-eabi-

CC = \$(SDK\_PREFIX)gcc

LD = \$(SDK\_PREFIX)ld

SIZE = \$(SDK\_PREFIX)size

OBJCOPY = \$(SDK\_PREFIX)objcopy

QEMU = qemu-system-gnuarmeclipse

BOARD ?= STM32F4-Discovery

MCU=STM32F407VG

TARGET=firmware

CPU\_CC=cortex-m4

TCP\_ADDR=1234

deps = \

start.S \

lscript.ld

all: target

target:

\$(CC) -x assembler-with-cpp -c -O0 -g3 -mcpu=\$(CPU\_CC) -Wall start.S -o

start.o

\$(CC) -x assembler-with-cpp -c -O0 -g3 -mcpu=\$(CPU\_CC) -Wall lab1.S -o

lab1.o

\$(CC) start.o lab1.o -mcpu=\$(CPU\_CC) -Wall --specs=nosys.specs -nostdlib -

lgcc -T./lscript.ld -o \$(TARGET).elf

\$(OBJCOPY) -O binary -F elf32-littlearm \$(TARGET).elf \$(TARGET).bin

qemu:

\$(QEMU) --verbose --verbose --board \$(BOARD) --mcu \$(MCU) -d

unimp,guest\_errors --image \$(TARGET).bin --semihosting-config

enable=on,target=native -gdb tcp::\$(TCP\_ADDR) -S

clean:

-rm \*.o

-rm \*.elf

-rm \*.bin

flash:

st-flash write \$(TARGET).bin 0x08000000

## Результати виконання

$a=11, b=5, c=2$

$r3 = (a+b)/c = 8$

```
Register group: general
r0      0xb      11
r1      0x5       5
r2      0x2       2
r3      0x8       8
r4      0x0       0
r5      0x0       0

0x8000002a <var3+8> subs    r3, r0, r1
0x8000002c <var3+10> sdiv   r3, r3, r2
>0x80000030 <var3+14> pop    {r0, r1, r2, pc}
0x80000032      movs     r0, r0
0x80000034      movs     r0, r0
0x80000036      movs     r0, r0

extended-r Thread 1 In: var3                                L24    PC: 0x80000030
var3 () at lab1.S:18
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) 
```

$a=5, b=5, c=2$

$r3 = (a+b)/c = 5$

```
Register group: general
r0      0x5       5
r1      0x5       5
r2      0x2       2
r3      0x5       5
r4      0x0       0
r5      0x0       0

0x8000002a <var3+8> subs    r3, r0, r1
0x8000002c <var3+10> sdiv   r3, r3, r2
>0x80000030 <var3+14> pop    {r0, r1, r2, pc}
0x80000032      movs     r0, r0
0x80000034      movs     r0, r0
0x80000036      movs     r0, r0

extended-r Thread 1 In: var3                                L24    PC: 0x80000030
var3 () at lab1.S:18
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) 
```

a=5, b=11, c=2  
r3 = (a-b)/c = -3

```
Register group: general
r0      0x5      5
r1      0xb      11
r2      0x2      2
r3      0xffffffffd -3
r4      0x0      0
r5      0x0      0

0x8000002a <var3+8> subs    r3, r0, r1
0x8000002c <var3+10> sdiv   r3, r3, r2
>0x80000030 <var3+14> pop    {r0, r1, r2, pc}
0x80000032      movs     r0, r0
0x80000034      movs     r0, r0
0x80000036      movs     r0, r0

extended-r Thread 1 In: var3          L24    PC: 0x80000030
(gdb) step
var3 () at lab1.S:18
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb)
```

## **Висновок**

В ході виконання лабораторної роботи було вивчено базові асемблерні інструкції ядра Cortex-M4 та створено проект на асемблері, що виконує розрахунок за заданою формулою. Виконання проекту було перевірено за допомогою відлагоджувача gdb. Також було створено Makefile для автоматизації створення прошивки.