НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



**ЗВІТ**

**Про виконання лабораторної роботи № 12**

**з дисципліни «Архітектура комп’ютера»**

**Лектор:**

Доцент Крук О.Г.

**Виконав:**

студ. групи ПЗ-23

Михалевич П.-І.В.

**Прийняв:**

Доцент Крук О.Г.

«\_\_\_» \_\_\_\_\_\_ 2021 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2021

**Тема роботи:** Розроблення програми для арифметичного співпроцесора мікроконтролера Cortex-M4F

**Мета роботи:** Розвинути навички складання програм для арифметичного співпроцесора ARM-процесорів мовою асемблера для обчислення математичного виразу, відтранслювати і виконати в режимі відлагодження програму, складену відповідно до свого варіанту, обчислити заданий вираз мовою C та порівняти результати.

**Теоретичні відомості**

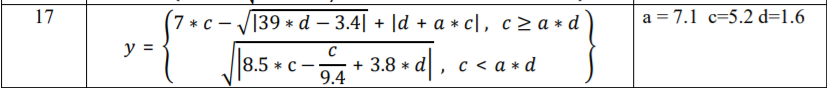
**Арифметичний співпроцесор** - цифровий пристрій, призначений для апаратного виконання арифметичних операцій над дійсними числами. Такий співпроцесор значно пришвидшує виконання програм, які працюють з дійсними числами.

Такий співпроцесор **FPU (Floating Point Unit)** вже інтегрований до процесора **Cortex-M4F**. Він підтримує стандарт арифметики чисел з плаваючою комою IEEE 754-2008. FPU на відміну від співпроцесорів Intel використовує не стек для роботи з дійсними числами, а спеціальні регістри. Таких регістрів є 32 від s0 до s31, якщо брати операнди з одинарною точністю, або 16 регістрів від d0 до d15, якщо брати операнди з подвійною точністю.

Кожен із регістрів подвійної точності складається з двох регістрів одинарної. Всі команди арифметичного співпроцесора повинні починатися з літери V.

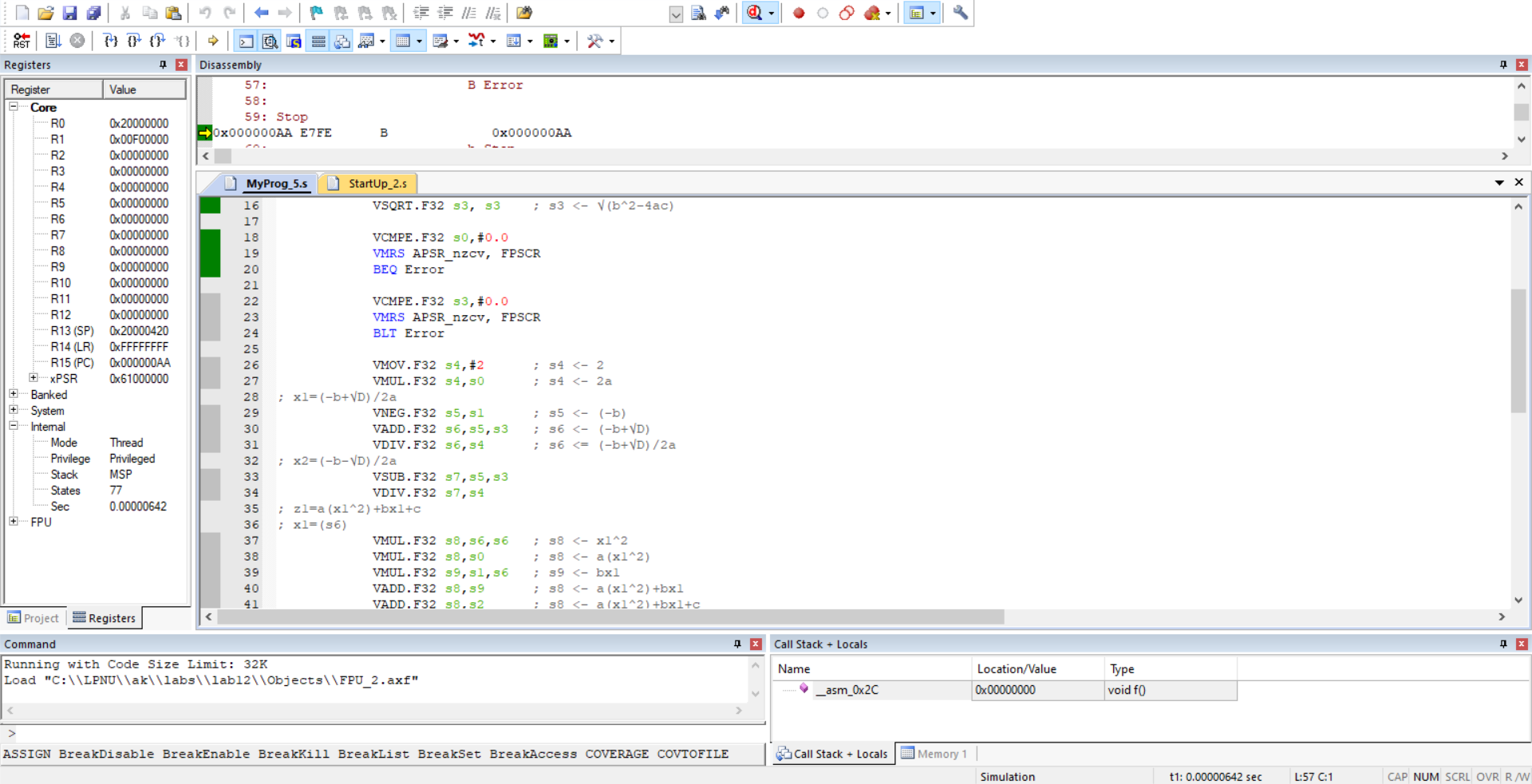
Для роботи з регістрами одинарної точності (32-бітовими регістрами) використовують суфікс **.F32**.

**Індивідуальне завдання**

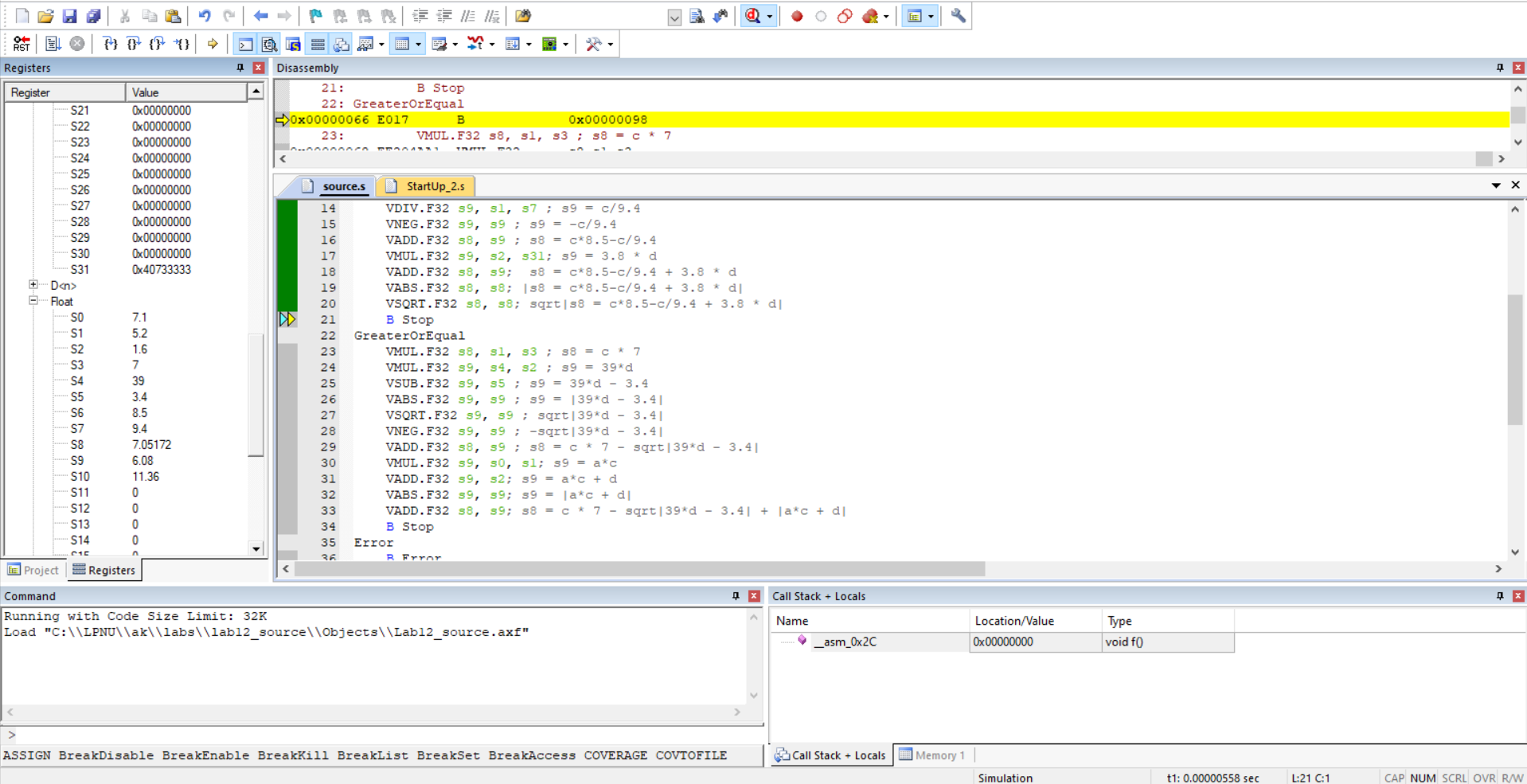


**Хід роботи:**

В середовищі Keil µVision виконано проект FPU\_2 покроково в режимі відлагодження, уважно простежено за змінами всіх регістрів, детально проаналізовано код програм.

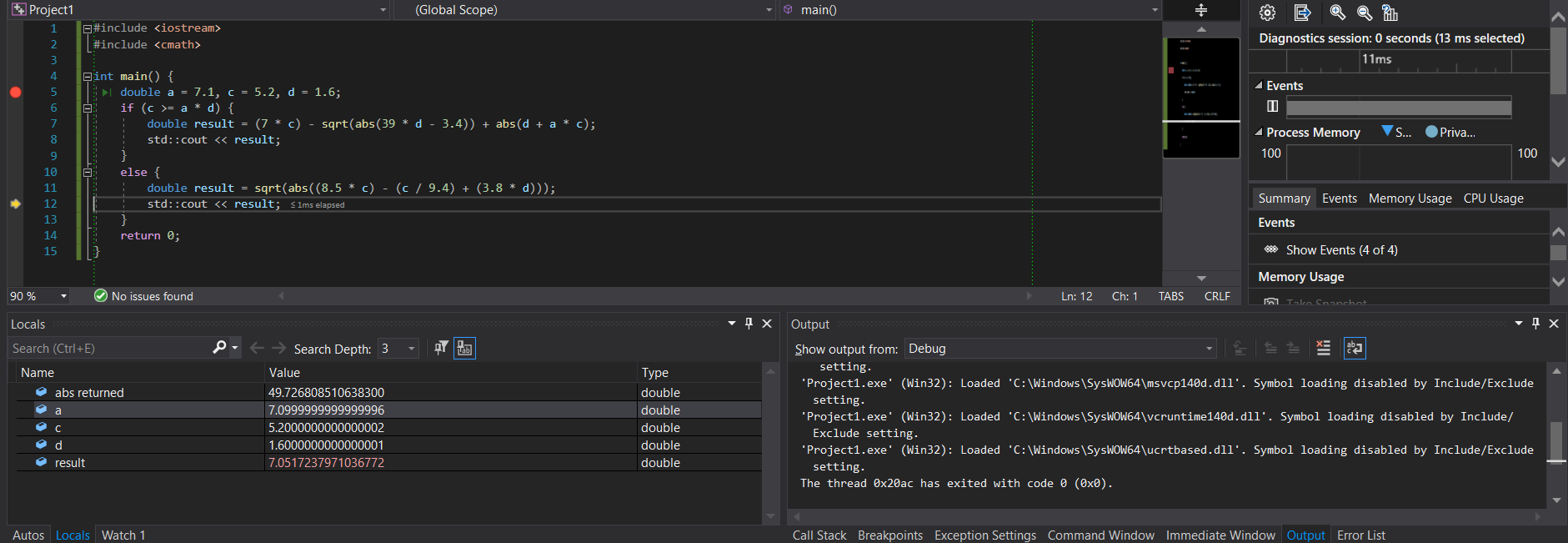


Складено програму обчислення виразу індивідуального варіанту за допомогою арифметичного співпроцесора ARM процесорів. Відтрансльовано програму StartUp\_2.s. Скомпоновано проект, виконано відлагодження в покроковому режимі та простежено за змінами регістрів.



В регістр s8 занесено значення 0x40A87DA9, який відповідає значенню +7.05172.

Перевірено результат роботи програми мовою асемблера аналогічною програмою мовою C, встановлено, що програма правильно обчислює вираз.



Код програми:

Назва файлу source.s:

AREA myCode, CODE, READONLY

MyProg

EXPORT MyProg

LDR r0, =a

VLDM r0, {s0-s7}; s0 = a, s1 = c, s2 = d, s3 = 7, s4 = 39, s5 = 3.4, s6 = 8.5, s7 = 9.4

LDR r0, = threeDotEight

VLDM r0, {s31}

VMUL.F32 s10, s0, s2 ; s10 = a\*d

VCMPE.F32 s0, s10; compare c and a\*d

VMRS APSR\_nzcv, FPSCR

BGE GreaterOrEqual

;c < a\*d

VMUL.F32 s8, s1, s6 ; s8 = c\*8.5

VDIV.F32 s9, s1, s7 ; s9 = c/9.4

VNEG.F32 s9, s9 ; s9 = -c/9.4

VADD.F32 s8, s9 ; s8 = c\*8.5-c/9.4

VMUL.F32 s9, s2, s31; s9 = 3.8 \* d

VADD.F32 s8, s9; s8 = c\*8.5-c/9.4 + 3.8 \* d

VABS.F32 s8, s8; |s8 = c\*8.5-c/9.4 + 3.8 \* d|

VSQRT.F32 s8, s8; sqrt|s8 = c\*8.5-c/9.4 + 3.8 \* d|

B Stop

GreaterOrEqual

VMUL.F32 s8, s1, s3 ; s8 = c \* 7

VMUL.F32 s9, s4, s2 ; s9 = 39\*d

VSUB.F32 s9, s5 ; s9 = 39\*d - 3.4

VABS.F32 s9, s9 ; s9 = |39\*d - 3.4|

VSQRT.F32 s9, s9 ; sqrt|39\*d - 3.4|

VNEG.F32 s9, s9 ; -sqrt|39\*d - 3.4|

VADD.F32 s8, s9 ; s8 = c \* 7 - sqrt|39\*d - 3.4|

VMUL.F32 s9, s0, s1; s9 = a\*c

VADD.F32 s9, s2; s9 = a\*c + d

VABS.F32 s9, s9; s9 = |a\*c + d|

VADD.F32 s8, s9; s8 = c \* 7 - sqrt|39\*d - 3.4| + |a\*c + d|

B Stop

Error

B Error

Stop

b Stop

a DCFS 7.1

c DCFS 5.2

d DCFS 1.6

seven DCFS 7

thirtyNine DCFS 39

threeDotFour DCFS 3.4

eightDotFive DCFS 8.5

nineDotFour DCFS 9.4

threeDotEight DCFS 3.8

ALIGN

AREA MyData, DATA, ReadWrite

x1 SPACE 4

x2 SPACE 4

z1 SPACE 4

z2 SPACE 4

END

Назва файлу Source.cpp:

#include <iostream>

#include <cmath>

int main() {

double a = 7.1, c = 5.2, d = 1.6;

if (c >= a \* d) {

double result = (7 \* c) - sqrt(abs(39 \* d - 3.4)) + abs(d + a \* c);

std::cout << result;

}

else {

double result = sqrt(abs((8.5 \* c) - (c / 9.4) + (3.8 \* d)));

std::cout << result;

}

return 0;

}

**Висновок**

На даній лабораторній роботі поглиблено знання мови асемблера процесорів Cortex-M4F. Написано програму мовою асемблера для процесорів Cortex-M4F та мовою Сі , що обраховує вираз згідно з індивідуальним завданням. Відтрансльовано, відлагоджено та виконано в покроковому режимі код програми. Виконано в покроковому режимі програми StartUp\_2.s, source.s, sorce.cpp. Досліджено принципи роботи даних програм та відслідковано зміни вмісту регістрів ЦП.