Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА № 4**

**з дисципліни «Архітектура комп’ютера-3» на тему**

**«РОЗРОБЛЕННЯ ПРОГРАМ ОБРОБКИ ДАНИХ ДЛЯ ПРОЦЕСОРНОГО ЯДРА CORTEX М4.**

**ВИВЧЕННЯ КОМАНД ПЕРЕДАВАННЯ УПРАВЛІННЯ»**

ВИКОНАВ:

студент ІІІ курсу ФІОТ

групи ІО-82

Вербовський Ілля

ПЕРЕВІРИВ:

Каплунов А. В.

Київ – 2021

**Варіант:** 820510 = 10 0000 0000 11012

**Посилання на репозиторій з кодом:**

<https://github.com/IllyaVerb/Lab1_4_AK-3>

**Завдання 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| h4 | h3 | h2 | h1 | Функція\* | | | |
| 1 | 0 | 1 | 1 |  | | | |
| h2 | h1 | X1 | X2 | X3 | X4 |
| 0 | 1 | 12 | 2 | –10 | 15 |

**Завдання 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| h4 | h5 | | h1 | | Номер точки переходу на першу підпрограму |
| 1 | 0 | | 1 | | VI |
| h2 | h1 | | Номер точки переходу на другу підпрограму | |
| 0 | 1 | | X | |

**Виконання роботи**

Розподіл пам’яті зображено на таблиці 1.

*Таблиця 1. Розподіл пам’яті*

|  |  |
| --- | --- |
| … | … |
| Векторна таблиця | 8000000h |
| 8000004h |
| Головна програма | 8000008h |
| … |
| 8000010h |
| Дані в пам’яті для завдання 1 | 8000012h |
| … |
| 8000032h |
| Код завдання 1 | 8000036h |
| … |
| 8000138h |
| Код завдання 2 | 800013Ch |
| … |
| 80001A0h |
| … | … |

Операнди містяться у комірках пам’яті (*0x8000012…0x8000032*).

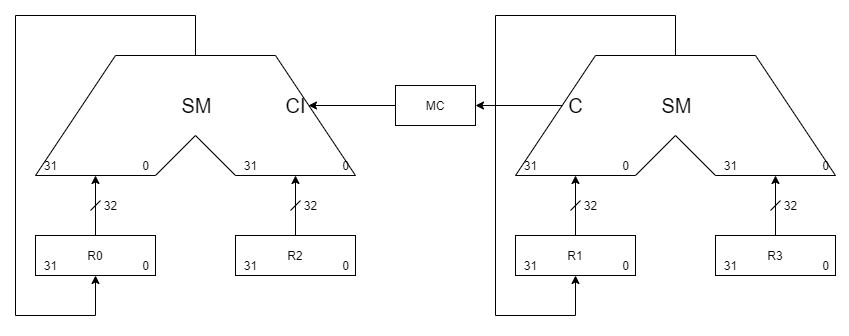
Читання даних з пам’яті у регістри здійснюється за допомогою команди *ldm*, а запис – *stm*.

Запис програми до оперативної пам’яті здійснюється методом інкрементної регістрової адресації з використанням команд *ldm, stm*.

Структурна схема суми операндів подвійної довжини зображена на рисунку 1, а приклад коду наведений у наступних рядках.

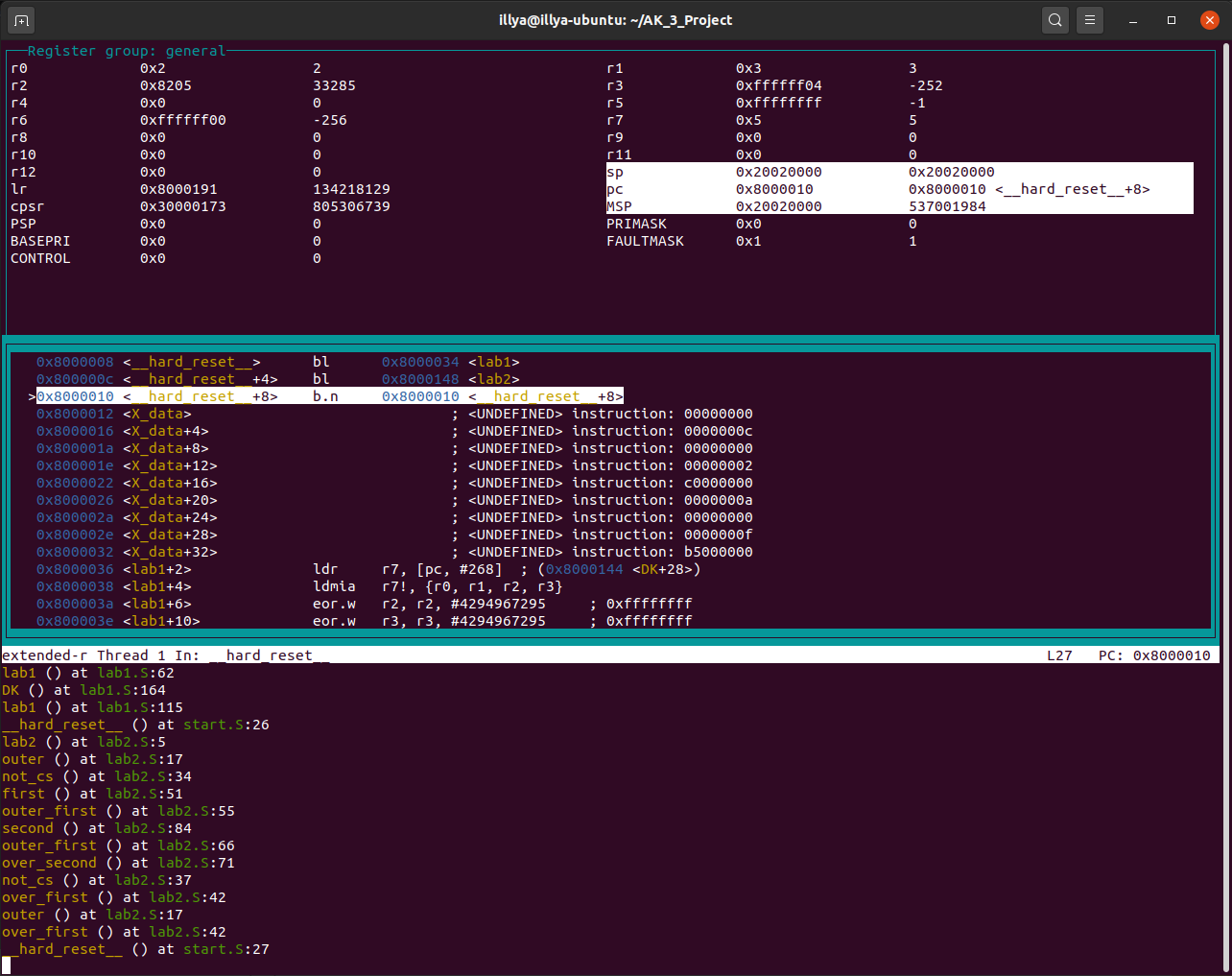
*adds r1, r3*

*adc r0, r2*



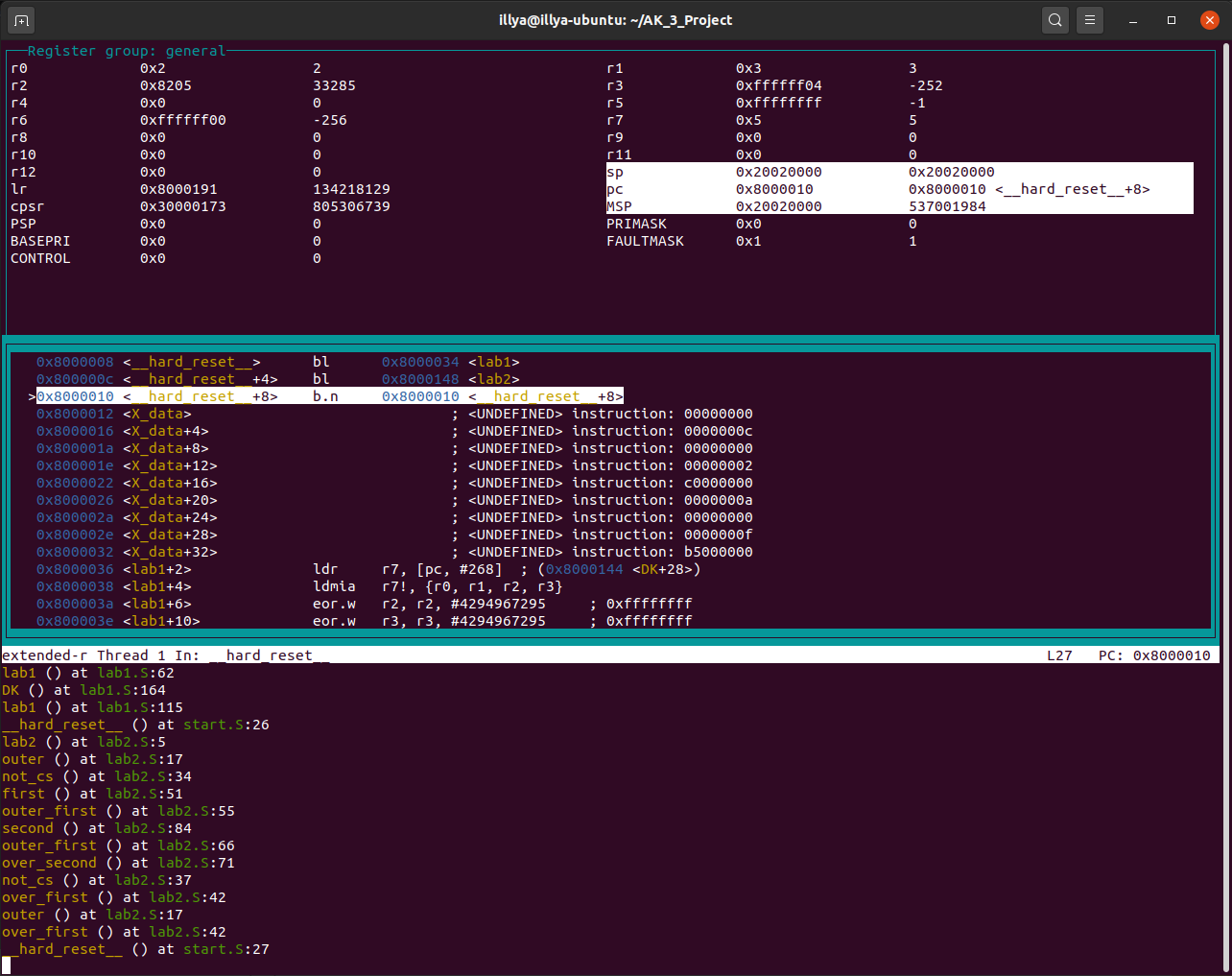
*Рисунок 1. Структурна схема суми операндів подвійної довжини*

Регістри після тестування програми з результатом функції з корекцією в регістрі *R2* та *R3* (в *R2* знаходиться номер заліковки – крок 4 корекції), зображені на рисунку 2.



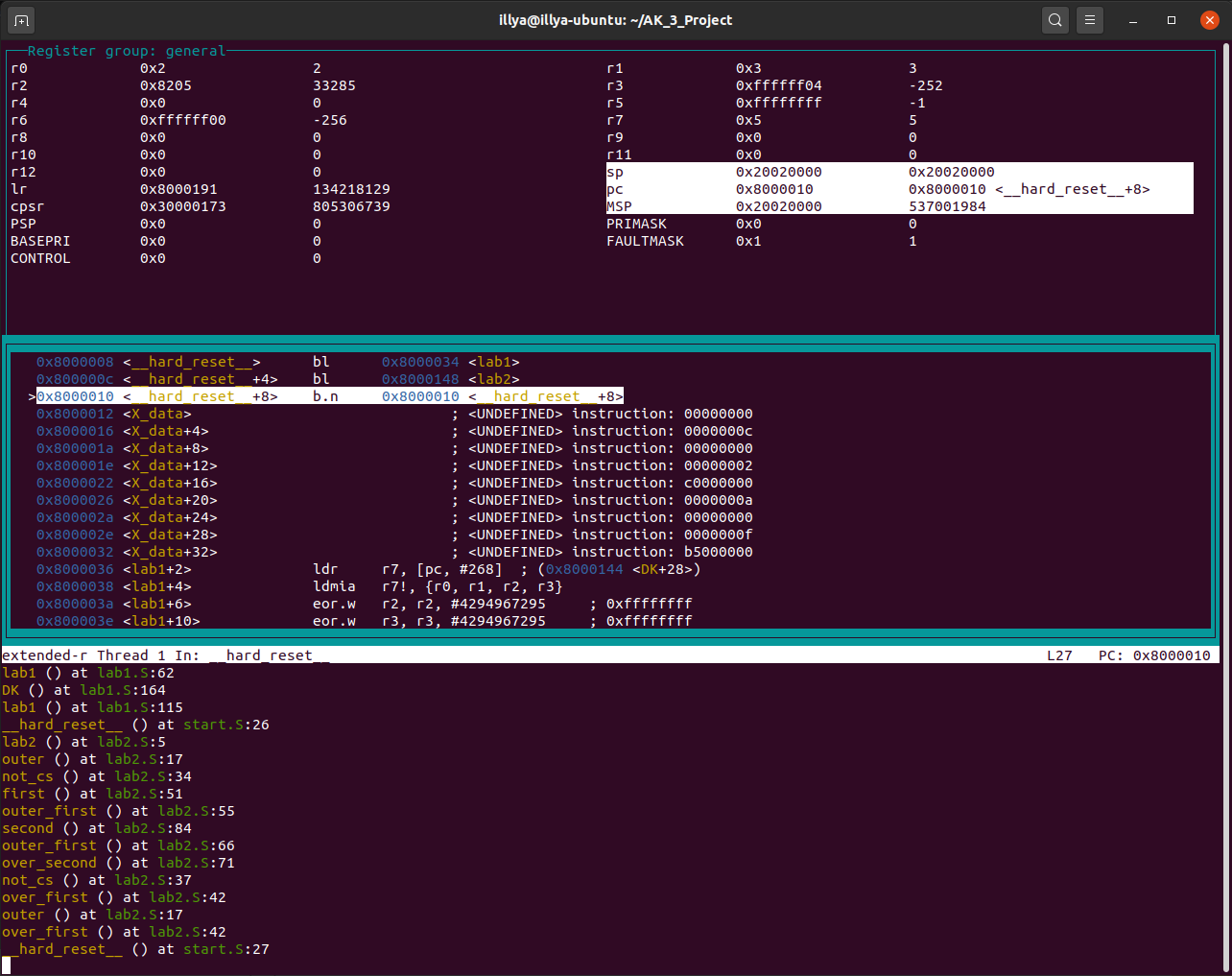
*Рисунок 2. Регістри з результатом функції*

Розміщення операндів в пам’яті зображено на рисунку 3.



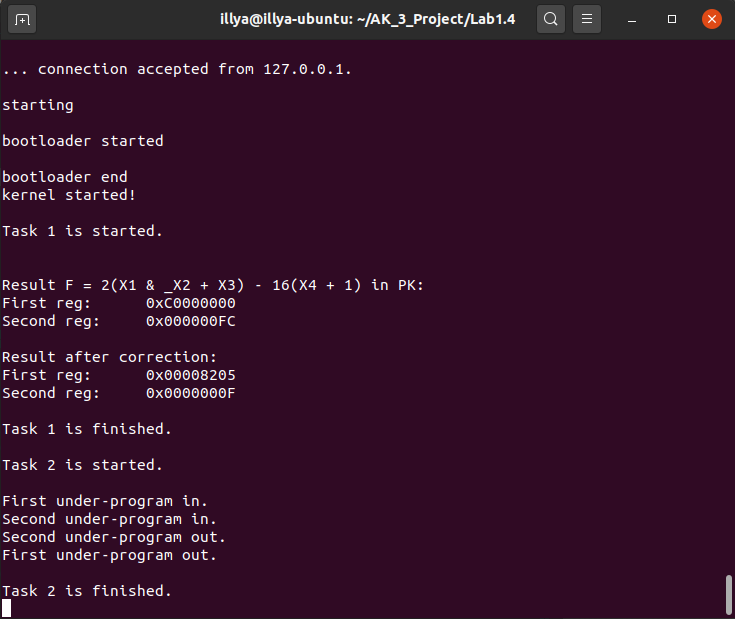
*Рисунок 3. Розміщення операндів в пам’яті*

Переходи між мітками в асемблерному коді, де *lab1* це завдання 1, а *lab2* та інші (крім *\_\_hard\_reset\_\_*) – завдання 2, зображені на рисунку 4.



*Рисунок 4. Переходи між мітками протягом виконання*

Результат виконання програми з завантажувачем та виводом у термінал значень в шістнадцятковій формі, зображений на рисунку 5.



*Рисунок 5. Виконання програми з виводом у термінал*

Наявні повідомлення:

* початок роботи програми – «*starting»*;
* початок та кінець роботи завантажувача – «*bootloader start / end»*;
* початок виконання завдань – «*kernel started!»*;
* початок та кінець 1 завдання – «*Task 1 is started /ended»*;
* результат функції у ПК – «*Result F…»*;
* результат функції після корекції у ДК – «*Result after…»*;
* початок та кінець 2 завдання – «*Task 2 is started / ended»*;
* переходи на першу та другу підпрограми і виходи з них – «*First / Second under-program in / out»*.