Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра інформаційної безпеки та комп’ютерної інженерії

**Звіт**

**з лабораторної роботи № 1**

**на тему: «Програмування циклу з переадресацією»**

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

ПЕРЕВІРИВ: ВИКОНАВ:

доцент студент групи

Бабенко В. Г. Кулинич О. М.

Черкаси 2016

Тема: Програмування циклу з переадресацією

Мета: Вивчення роботи програм з використанням умовної адресації при вирішенні задач пов’язаних з обробкою масивів.

**Завдання**

**Варіант 3**

1. Розробити програму що знаходить добуток всіх чисел в масиві.
2. Скласти граф-схему алгоритму вирішення задачі
3. Розподілити пам’ять ОЗП.
4. Записати програму в вигляді таблиці.
5. Зберегти результат виконання програми

**Теоретичні відомості**

Від програм, які мають просту лінійну структуру, як правило, дуже мало користі. Практично кожна програма, яка має будь-яке практичне застосування, повинна аналізувати деякі умови, і залежно від виконувати ті чи інші дії, тобто містити розгалуження та умовні переходи.

У багатьох випадках виникає необхідність багатократного повторення якихось однорідних операцій, тобто виконання циклу. Класичним прикладом циклічних програм є програми, що здійснюють обробку масивів (цикл по елементах масиву), файлі (цикл по записах файлу), керують подіями (цикл обробки подій), обчислюють щось за ітераційними або рекурентними формулами.

Для підтримки програм з циклічною структурою у складі системи команд процесорів 80х86 є спеціальні команди організації циклів. Але ці команди не володіють логічною повнотою (не дозволяють організувати всі можливі і необхідні види циклів), тому в ряді випадків доводиться імітувати циклічну структуру вручну з допомогою умовних і безумовних переходів по параметру циклу.

Засоби мов високого рівня для організації циклів всіх типів спираються на команди циклів процесора та на команди умовного і безумовного переходів (при компіляції переводяться саме в ці команди).

Найчастіше використовуються такі команди умовного переходу:

- JNS — перехід якщо вміст акумулятора не від’ємний

- JS — перехід якщо вміст акумулятора від’ємний

- JNZ — перехід якщо вміст акумулятора не рівний нулю

- JZ — перехід якщо вміст акумулятора рівний нулю

Також використовують команду безумовного переходу JMP.

**Хід роботи:**

**Варіант №3**

Розробити програму що знаходить добуток всіх чисел в масиві

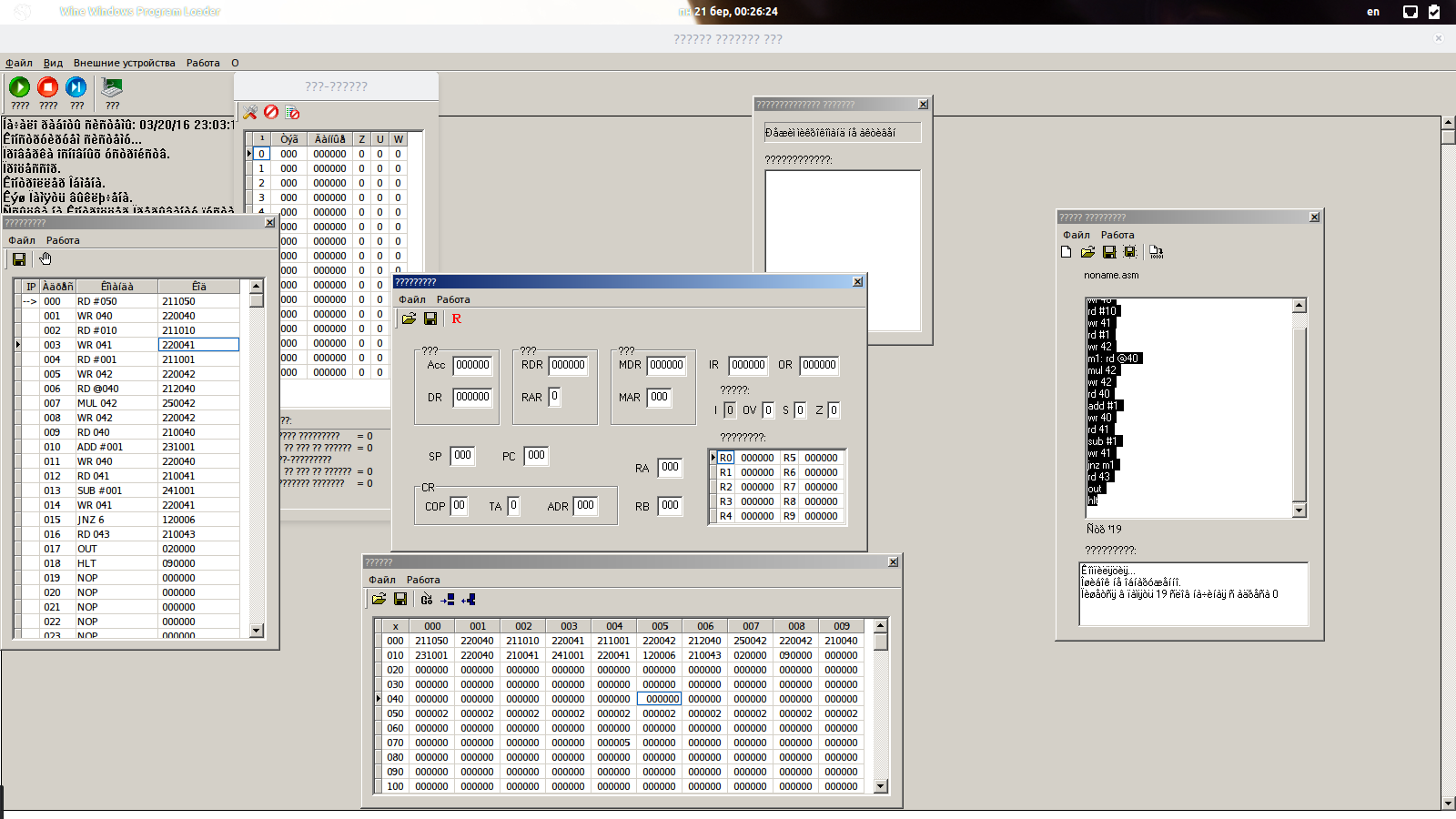
**Граф-схема вирішення задачі:**

diagsanya3.png

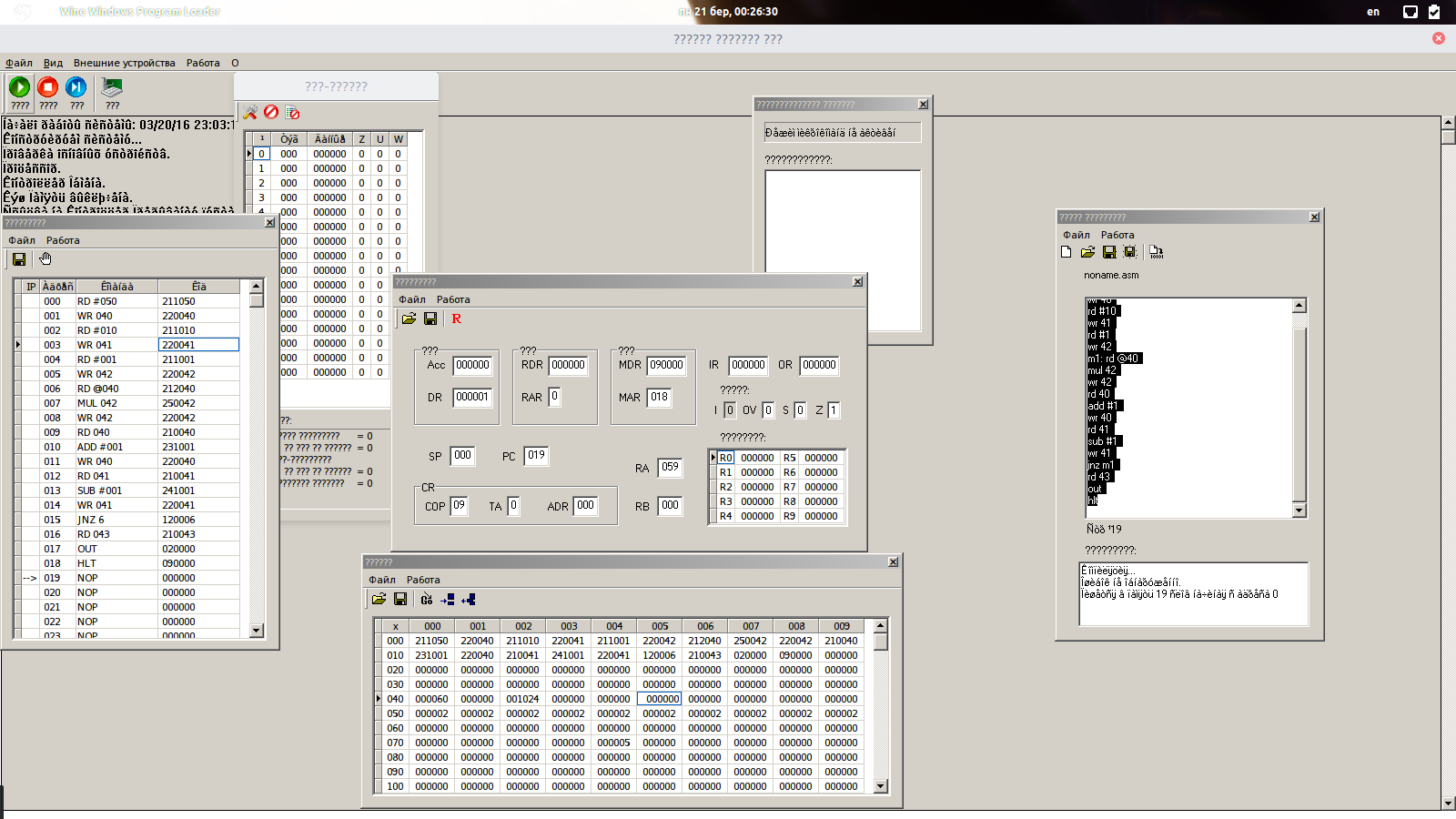
|  |  |  |
| --- | --- | --- |
| № | Команда | Опис команди |
| 000 | rd #50 | Зчитування числа — адресу 1 елементу масиву |
| 001 | wr 40 | Запис його в 40 комірку |
| 002 | rd #10 | Зчитування довжини масиву |
| 003 | wr 41 | Запис її в 41 комірку |
| 004 | rd #1 | Зчитування одиниці як початкового числа для добутку |
| 005 | wr 42 | Запис її в 42 комірку |
| 006 | m1: rd @40 | Зчитування поточного елементу масиву |
| 007 | mul 42 | Множення його на поточний добуток |
| 008 | wr 42 | Перезапис добутку |
| 009 | rd 40 | Зчитування адресу поточної комірки масиву |
| 010 | add #1 | Збільшення його на 1 |
| 011 | wr 40 | Перезапис |
| 012 | rd 41 | Зчитування довжини масиву |
| 013 | sub #1 | Зменшення її на 1 |
| 014 | wr 41 | Перезапис |
| 015 | jnz m1 | Якщо не досягнуто кінця масива, перехід до m1 |
| 016 | rd 42 | Інакше зчитування результату |
| 017 | out | Виведення |
| 018 | hlt | Вихід з програми |

***Таблиця 1 : Текст прорами***

**Вид моделі до запуску програми:**



**Результат виконання програми:**



**Висновок**

На лабораторній роботі я створив програму що оброблює масиви, використовуючи цикли в моделі комп’ютера. Я програмував цикли використовуючи оператори умовного переходу та мітки. Моя програма знаходить добуток всіх елементів масиву.