**Міністерство освіти і науки України Національний технічний університет України «Київський Політехнічний Інститут імені Ігоря Сікорського» Кафедра конструювання електронно-обчислювальної апаратури**

Звіт з виконання лабораторної роботи №2

з дисципліни “Основи мікропроцесорної техніки”

Виконав: студент групи ДК-82

Рудюк Б. Б.

Перевірила: ст.вик.

Бондаренко Н. О.

**Київ – 2020**

**Хід роботи**

**Варiант 17**

1. Очистити регістр CX

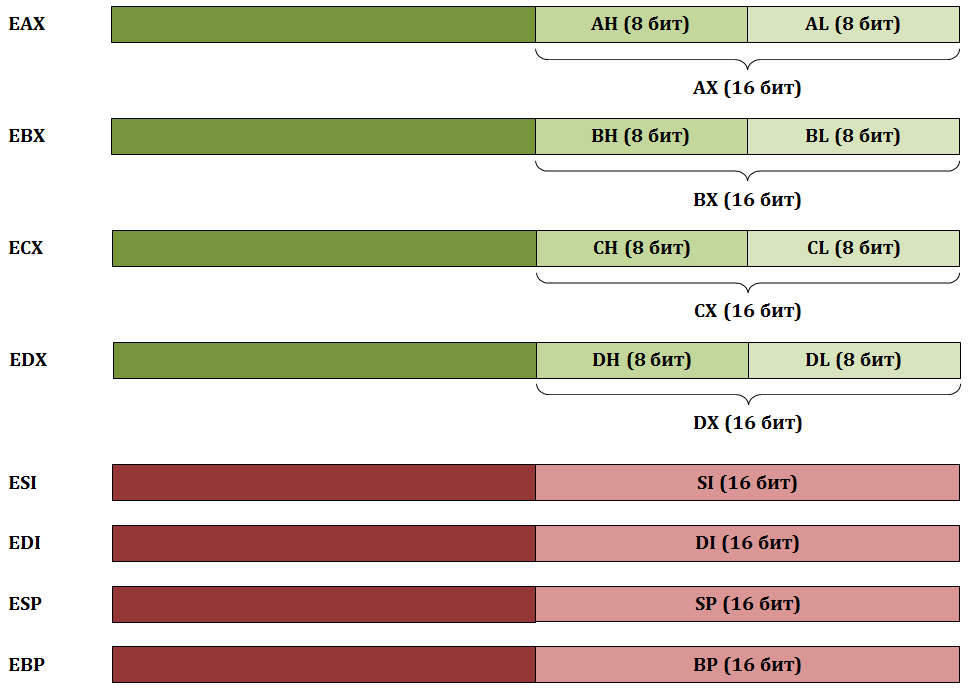
2. Переслати число 18H у регістр CL

3. Додати число 8559H до регістра CX

4. Переслати регістр CX у регістр AX

5. Додати регістр ЕAX до регістра ЕCX

6. Відняти регістр AX із регістра CX

Які регістри є в наявності для використання при написанні програми:

До регістрів загального призначення відноситься група з 8 регістрів, які можна використовувати в програмі на мові асемблера. Всі регістри мають розмір 32 біти і можуть бути розділені на 2 або більше частин.

В регістрах EAX, EBX, ECX, EDX є можливість записувати дані як в повний обсяг регістру(EAX), так і в молодші 16біт (АХ), молодший байт (AL) та старший байт (AH) молодших 16біт (AX). В регістри ESI , EDI , ESP , EBP можна записувати дані тільки в повний обсяг регістру або в молодші 16біт.

Процесор має регістр прапорців, що містить набір прапорців, який відображає поточний стан процесора.

*Прапорець* – це біт, приймаючий значення 1 («флаг встановлено»), чи виконує деяку умову, і значення 0 («флаг скинуто»), якщо навпаки.

**«Прапорці»  
 FLAGS (CPU) DEBUGGER (VStudio)**

OF прапорець переповнювання OV

DF прапорець напряму UP

IF прапорець переривання EI

TF прапорець перехоплення -

SF прапорець знаку PL

ZF прапорець нуля ZR

AF прапорець додаткового перенесення AC

PF прапорець парності PE

CF прапорець перенесення CY

Відмітимо, що для правильного виконання лабораторної відразу необхідно налаштувати Visual Studio, для того, щоб коректно компілювався програмний код і мати змогу перевіряти всі кроки роботи коду. Тому, перш за все беремо шаблон, в який вставимо написаний код. Наступний крок - це налаштування дебагування коду, щоб виводити на екран всю інформацію про зміни, які відбудуться в регістрах. Таким чином, вид робочого вікна Visual Studio при дебагуванні коду буде наступним (рис.1):

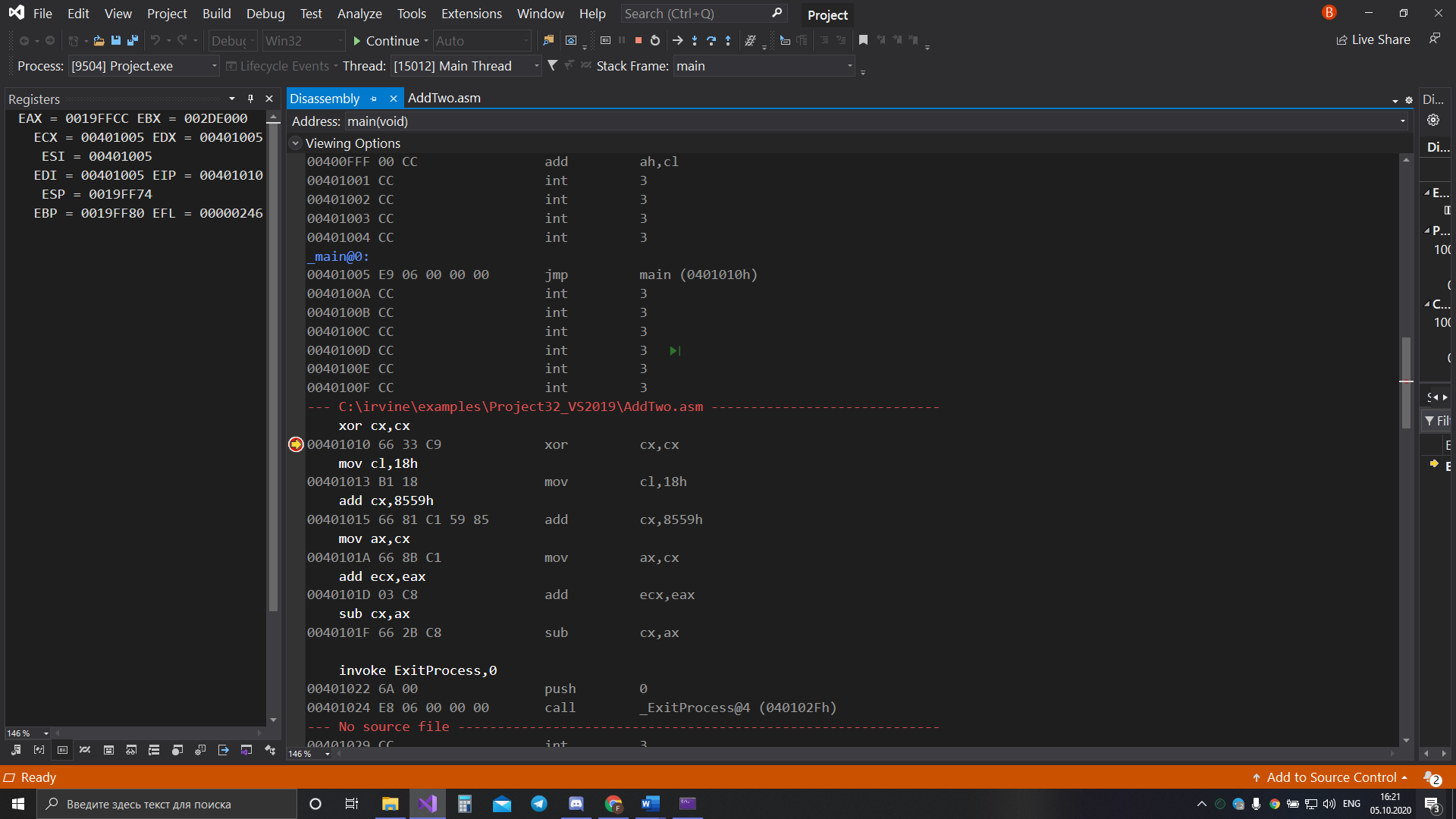


Рис.1 Робоче вікно Visual Studio

Як бачимо, ми можемо відслідковувати всі етапи виконання коду за рахунок наглядних змін у регістрах та прапорцях.

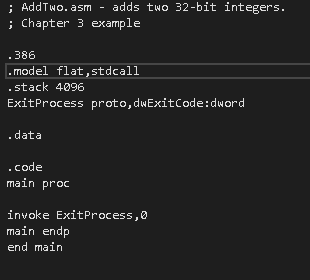
Тепер переходимо до написання самого коду. Спершу візьмемо шаблон, який виглядає наступним чином (рис.2):

Рис.2 Шаблон коду

Переходимо до написання коду.

1. **Очищення регістру CX**

Для цього використовуємо команду “ xor cx, cx ”. При виконанні даної команди бачимо, що вміст регістра CX обнулився (рис.3). Також слід відмітити, що зміни відбулись в регістрі EIP, який фактично є вказівником команд і вміщує зміщення наступної команди, що підлягає виконанню.

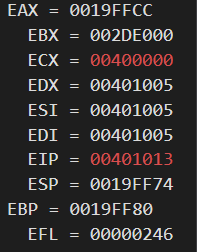


Рис.3 Очистка регістру CX

1. **Реалізація перенесення числа 18H у регістр CL**

Використовуємо команду “mov cl, 18h” . Перевіряємо які зміни відбулись в регістрах. Бачимо, що в регістр CL записалось число 18Н (рис.4). Ну і так само, як і у попередньому завданні, у нас змінився регістр EIP.

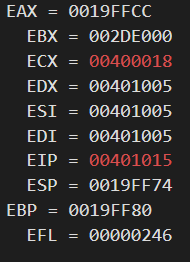


Рис.4 Запис числа 18Н у регістр CL

Відмітимо, що CX та ЕCХ є один і той самий базовий регістр, але коли ми говоримо про регістр CX, то маємо на увазі, що звертаємось до двох молодших байтів регістра ЕCХ. Положення прапорців не змінилось, бо операція не є ні арифметичною, ні логічною.

1. **Додамо число 8559Н до регістру CХ**

При виконанні команди “add cx, 8559h “ ,бачимо, що змінився вміст регістрів ЕCХ, EIP та EFL. Якщо за перші два регістри зрозуміло за попередніми завданнями, то про EFL хотілось би нагадати, що він відповідає за прапорці (EFLAGS), що приймають значення 1 («прапор встановлений»), якщо виконана деяка умова, і значення 0 («прапор скинутий») в іншому випадку, тому поточний стан процесора: (рис.5 та рис.6).

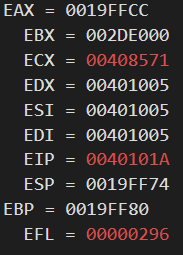
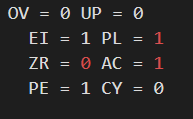
 

Рис.5 Додавання числа Рис.6 Положення прапорців після додавання

Бачимо, що до регістру CX додалось число 8559Н. Результат операції, регістр СХ буде рівний 8571H (1000 0101 0111 0001). Варто відмітити, що прапорці змінюють своє значення тільки при арифметичних та логічних операціях. Прапор нуля ZR встановлюється в 1, якщо результат операції дорівнює нулю, у нашому випадку результат операції не дорівнює нулю і тому прапорець ZR = 0. Прапор парності PЕ встановлюється в 1, якщо молодший байт результату містить парне число одиничних бітів, у нашому випадку прапорець PE = 1, бо молодший байт отриманого регістра СХ, містить парну . Прапор знаку PL вказує, якого знаку наше число. AC прапорець додаткового перенесення, встановлюється, коли при виконанні арифметичної операції з 8-розрядним операндом виконується перенос з третього біта в четвертий.

1. **Пересилаємо регістр СХ у AХ**

За допомогою команди “mov ax, cx” ми пересилаємо регістр CX у регістр AX (рис.7 та рис.8)

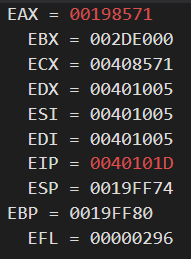
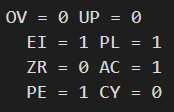
 

Рис.7 Пересилання одного регістра в інший Рис.8 Положення прапорців після

пересилання

Як бачимо положення прапорців не змінилось, бо операція не є ні арифметичною, ні логічною.

1. **Додавання регістру EAХ до регістра ECХ**

Реалізація даної операції здійснюється за допомогою команди “add ecx, eax”.Операція арифметична, тому прапорець PL змінив своє значення( рис.9 та рис.10).

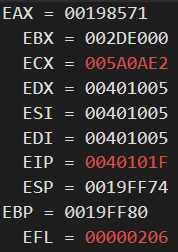
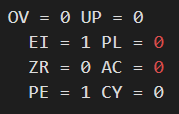
 

Рис. 9 Додавання одного регістра в інший Рис.10 Положення прапорців після додавання

1. **Віднімання регістру AХ від регістра CХ**

За допомогою команди “sub cx, ax”(рис.11 та рис.12). Операція арифметична, тому прапорці PL і CY змінились.

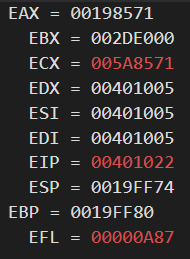
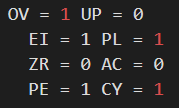
 

Рис.11 Віднімання одного регістра від іншого Рис.12 Положення прапорців після

віднімання

Отже, аналізуючи всі етапи виконання програми, програма працює коректно.OV прапорець переповнювання, встановлюється у випадку, коли при виконані арифметичної операції, зі знаком, виходить число, розрядність якого перевищує розрядність виділеного для нього поля результата. CY прапорець перенесення, встановлюється у випадку, коли при виконані беззнакової арифметичної операції виходить число, розрядність якого перевищує розрядність виділеного для нього поля результата.

1. **Текст програми з необхідними коментарями:**

TITLE addition and subtraction ; Назва програми

; Lab1 ;Коментарі

; Last update: 22.09.2020

.386 ;Ця директива визначає тип процесора, для якого створюється програма

.MODEL flat, stdcall ;Model - вказує компілятору, що потрібно генерувати код для захисного режиму роботи процесора,

stdcall - надає можливість в програмі викликати функції системи MS Win.

.STACK 4096 ; Резервація області пам’яті для стека

ExitProcess PROTO, dwExitCode:dword ; ExitProcess – функція системи Win, яка завершує виконання поточної програми.

.code ;Початок сегмента кода

main PROC ; PROC - позначає початок процедури, яка має назву main

xor cx, cx ;Очистити регістр CX

mov cl, 18h ;Переслати число 18H у регістр CH

add cx, 8559h ;Додати число 8559H до регістра CX

mov ax, cx ;Переслати регістр CX у регістр AX

add ecx, eax ;Додати регістр ЕAX до регістра ЕCX

sub cx, ax ;Відняти регістр AX із регістра CX

INVOKE ExitProcess, 0 ;Викликання функції ExitProcess, яка передає значення 0

main ENDP ;ENDP - позначає кінець процедури main

END main ;END - остання строка програми, у ній вказується назава точки входи в програму (main)

**Висновки:**

Виконавши дану лабораторну роботу, ми створили коректно працюючу програму, яка виконує прості операції, серед яких очищення регістру, додавання, віднімання, пересилання. Ми розібрались з роботою регістрів та прапорців, і зрозуміли, яку функцію виконує кожен з них. Засвоїли операції, що впливають на прапорці.