МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА КОНСТРУЮВАННЯ ЕОА

ЗВІТ

з лабораторної роботи №1

по курсу «Основи мікропроцесорної техніки - 1»

Виконав:

студент гр. ДК-82

Сопіра Р. Я.

Перевірив:

доц. Корнєв В. П.

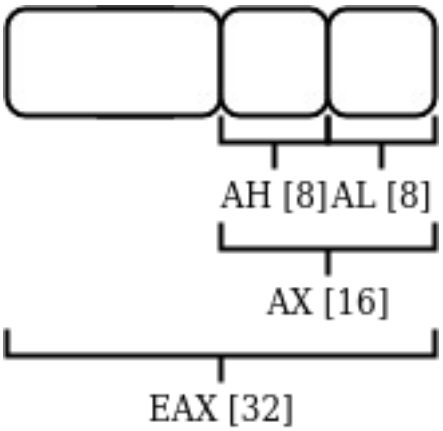
ст. в. Бондаренко Н. О.

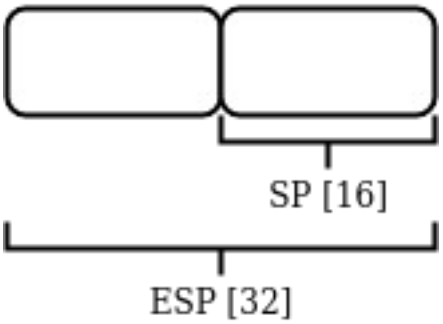
Київ – 2020

Теоретичні відомості

Регістром називають ділянку високошвидкісної пам’яті, розташовану всередині ЦП і призначену для оперативного збереження даних з швидким доступом до них зі сторони внутрішніх компоненті процессора.

Регістри загального призначення використовуються в основному для виконання арифметичних операцій і передачі даних. До кожного з таких регістрі можна звертатися як до 32- чи 16-розрядному, а до деяких, наприклад, до AX можна звертатися як до двух 8-розрядних — старші 8-розрядів знаходяться в регістрі AH, молодші 8-розрядів у AL.





Регістри загального призначення:

- EAX/AX/AH/AL (accumulator register) — аккумулятор;
- EBX/BX/BH/BL (base register) — регістр бази;
- ECX/CX/CH/CL (counter register) — лічильник;
- EDX/DX/DH/DL (data register) — регістр данних;

Наступні регістри зазвичай використовуються для команд передачі даних в пам’яті:

- ESI/SI (source index register) — індекс джерела;
- EDI/DI (destination index register) — індекс одержувача;

Наступними регістрами виконується звертання до даних, що зберігаються в стеці:

- ESP/SP (stack pointer register) — регістр вказівника на верх стеку;
- EBP/BP (base pointer register) – регістр вказівника бази стеку;

Даний регістр зберігає адресу наступної команди:

- EIP — вказівник команд;

Кожен із бітів цього регістру відповідає за особливості виконання деяких команд ЦП або результати виконання команд АЛУ:

- EFL — регістр “прапорців”;

Регістр “прапорців”:

- OF - прапорець переповнювання;
- DF - прапорець напряму;
- IF прапорець переривання;
- TF прапорець перехоплення;
- SF прапорець знаку;
- ZF прапорець нуля;
- AF прапорець додаткового перенесення;
- PF прапорець парності;
- CF прапорець перенесення;

Flat модель пам’яті — один із методів організації адрес оперативной пам’яті обчислювальної техніки. В такій моделі і код, і дані використовують один адресний простір. Для 16-бітних процесорів дана модель пам’яті дозволяє теоретично адресувати 64 кілобайт оперативної пам’яті; для 32-бітних процессоров — 4 гігабайт, для 64-бітних — до 16 эксабайт.

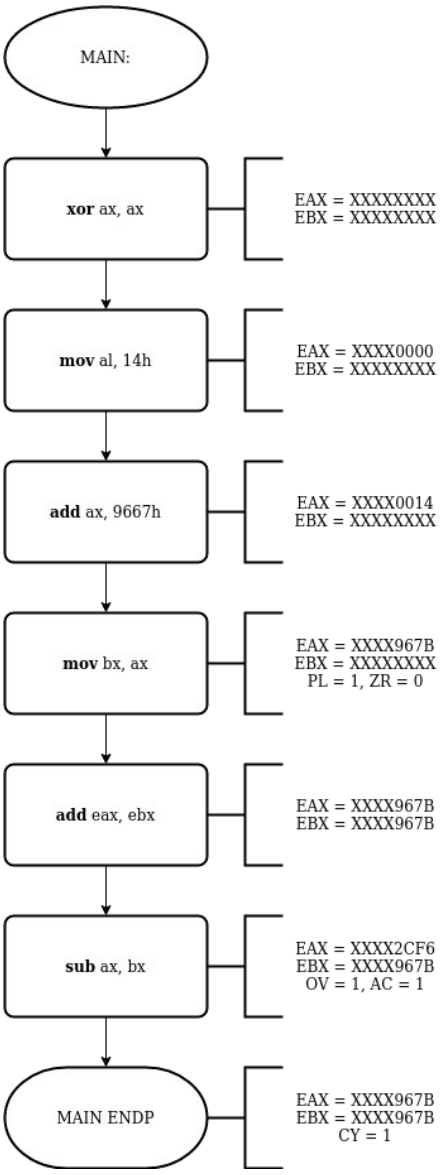
Завдання

Варіант 18:

1) Очистити регістр AX;
2) Переслати число 14H у регістр AL;
3) Додати число 9667H до регістра AX;
4) Переслати регістр AX у регістр BX;
5) Додати регістр EBX до регістра EAX;

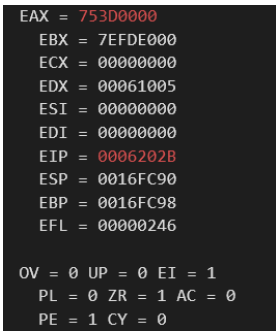
Код і блок-схема програми

```
1   .386
2   .model flat, stdcall
3   .stack 4096
4
5   ExitProcess PROTO, dwExitCode:DWORD
6
7   .code
8   main PROC
9           xor ax, ax
10          mov al, 14h
11          add ax, 9667h
12          mov bx, ax
13          add eax, ebx
14          sub ax, bx
15
16          invoke ExitProcess, 0
17  main ENDP
18
19  END main
```

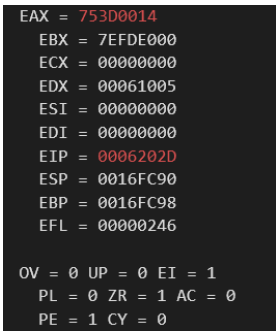


Покрокове виконання

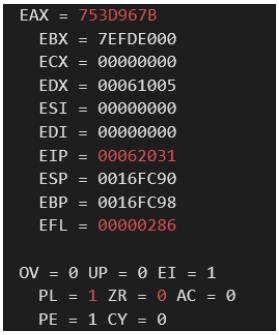
1. **xor** ax, ax — обнуляє регістр AX



2. **mov** al, 14h — записує число 14_{16} в регістр AL



3. **add** ax, 9667h — додає до регістра AX число 9667_{16}, прапорець нуля ZR скидається, бо результат не рівний нулю, а прапорець PL взводиться, бо результат інтерпретується як від’ємний



4. **mov** bx, ax — копіює значення регістра AX у регістр BX

```
EAX = 753D967B
  EBX = 7EFD967B
  ECX = 00000000
  EDX = 00061005
  ESI = 00000000
  EDI = 00000000
  EIP = 00062034
  ESP = 0016FC90
  EBP = 0016FC98
  EFL = 00000286

OV = 0 UP = 0 EI = 1
  PL = 1 ZR = 0 AC = 0
  PE = 1 CY = 0
```

5. **add** eax, ebx — додає значення регістра EBX до EAX, взводяться прапорці OV — переповнення, бо результат додавання має на розряд більше ніж розрядність регістру і AC

```
EAX = F43B2CF6
  EBX = 7EFD967B
  ECX = 00000000
  EDX = 00061005
  ESI = 00000000
  EDI = 00000000
  EIP = 00062036
  ESP = 0016FC90
  EBP = 0016FC98
  EFL = 00000A96

OV = 1 UP = 0 EI = 1
  PL = 1 ZR = 0 AC = 1
  PE = 1 CY = 0
```

6. **sub** ax, bx — віднімає значення регістра BX від регістра AX, взводячи прапорець переносу CY

```
EAX = F43B967B
  EBX = 7EFD967B
  ECX = 00000000
  EDX = 00061005
  ESI = 00000000
  EDI = 00000000
  EIP = 00062039
  ESP = 0016FC90
  EBP = 0016FC98
  EFL = 00000A97

OV = 1 UP = 0 EI = 1
  PL = 1 ZR = 0 AC = 1
  PE = 1 CY = 1
```

Висновок

У процесі виконання даної лабораторної роботи було створено програму, що виконує прості задані операції очищення регістру, пересилання з регістру в регістр, додавання, віднімання тощо, мовою ассемблеру архітектури x86. Також вдалося ознайомитися з роботою регістрів загального призначення та прапорців, їх призначенням та функціями, впливом операцій, які впливають на зміну їхнього стану.

Вихідний код програм доступний за посиланням:

[===]