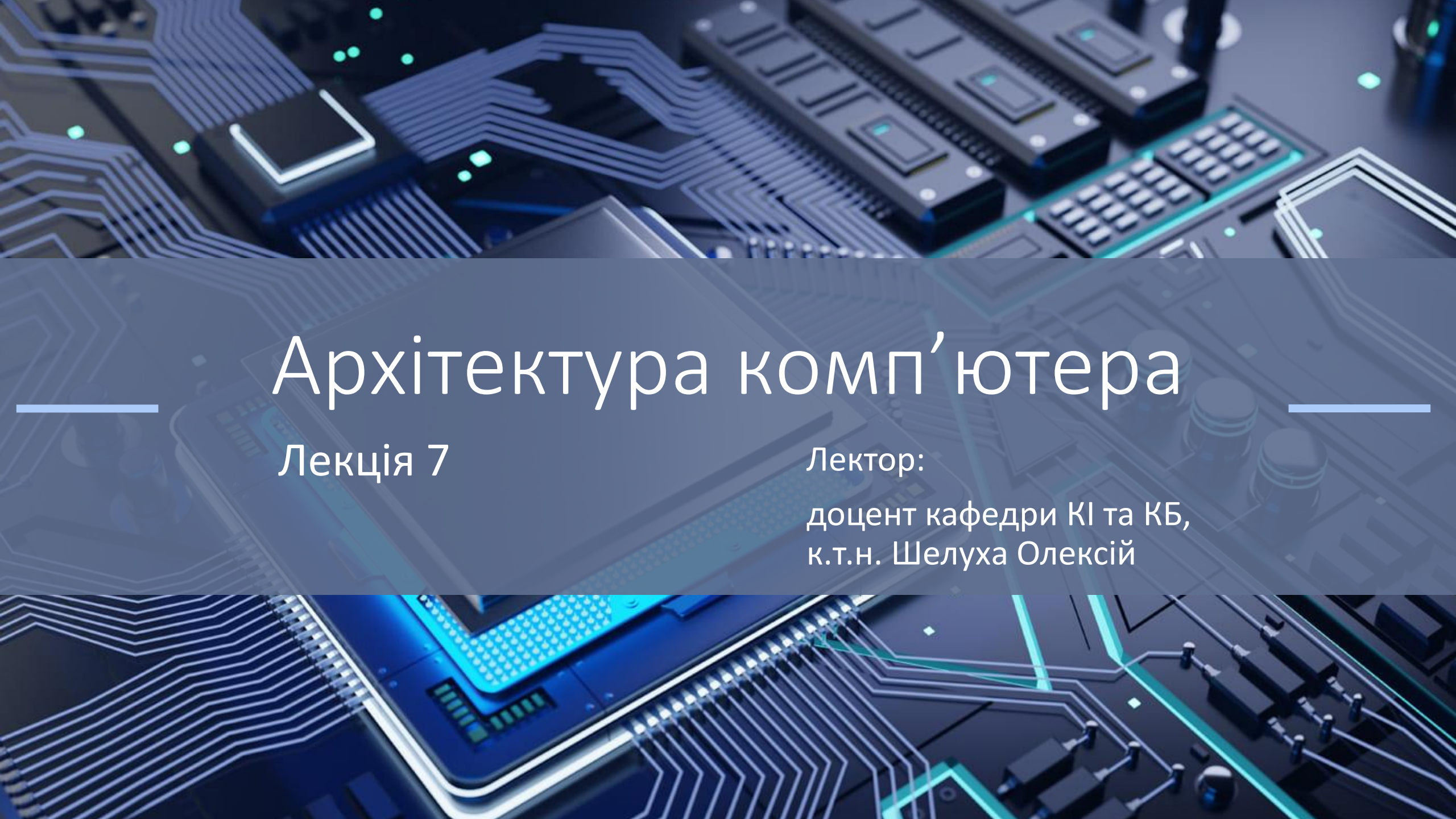


Архітектура комп’ютера

Лекція 7

Лектор:

доцент кафедри КІ та КБ,
к.т.н. Шелуха Олексій

Зміст лекції

- Прості команди
- Приклад розрахунку простих 16-бітних виразів
- Умови та ресурси для виконання операцій множення та ділення
- Команди множення, ділення та зміщення
- Приклад розрахунку складних виразів

Команда пересилки MOV

- Синтаксис:
  MOV Операнд 1 (Приймач), Операнд 2 (Джерело)
- ДОВЖИНИ ОБОХ операндів повинні бути однаковими

Команда обміну даними XCHG

- Синтаксис:
  XCHG Операнд 1 (Приймач), Операнд 2 (Джерело)
- ДОВЖИНИ ОБОХ операндів повинні бути однаковими.
- Існує три варіанта команди XCHG:
  XCHG Register Register
  XCHG Register Memory
  XCHG Memory Register
- Приклади використання команди XCHG:
  1. Обмін вмісту 16-розрядних регістрів: xchg AX, BX
  2. Обмін вмісту 8-розрядних регістрів: xchg AH, AL
  3. Обмін вмісту 16-розрядного операнда в пам’яті і регістра BX: xchg VAR1, BX
  4. Обмін вмісту 32-розрядних регістрів: xchg EAX, EBX

Стан прапорців після виконання команд додавання

Прапорець	Призначення	Пояснення
CF=1	прапорець перенесення	Результат додавання не помістився в операнд-приймач
PF=1	прапорець парності	Результат додавання має парну кількість бітів із значенням 1
SF=1	прапорець знаку	Копіюється СТАРШИЙ (ЗНАКОВИЙ) біт результату додавання
ZF=1	прапорець нуля	Результат додавання дорівнює нулю
OF=1	прапорець переповнення	При додаванні двох чисел з ОДНАКОВИМ знаком (два додатні або два від’ємні) результат додавання вийшов БІЛЬШЕ ніж допустиме значення. В цьому випадку приймач МІНЯЄ ЗНАК.

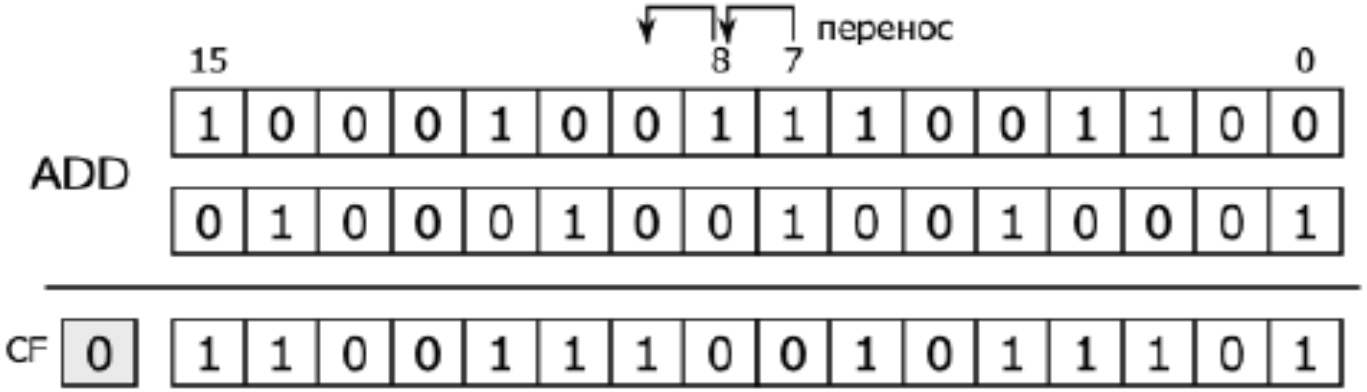
Команда ADD

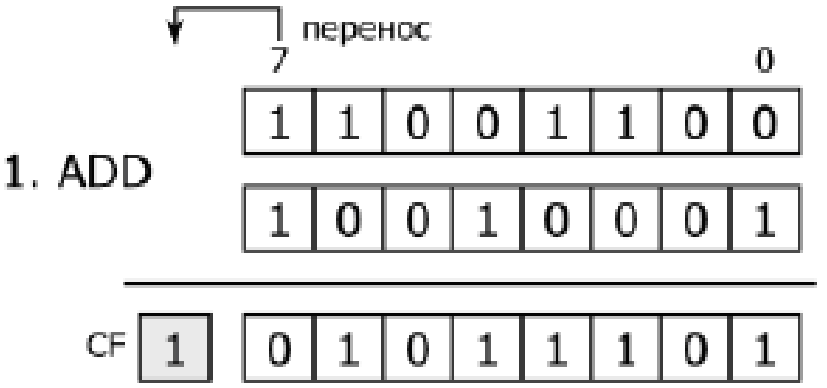
- Синтаксис:
  ADD Приймач, Джерело
- Нехай у нас є два числа – 120 і 100. Щоб їх додати. Необхідно виконати наступні дії:
  mov al, 120 ; al <=== 120
  mov cl, 100 ; cl <=== 100
  add al, cl ; <al>:=<al>+<cl>
  mov x, al ; x <=== <al>

Команда ADC

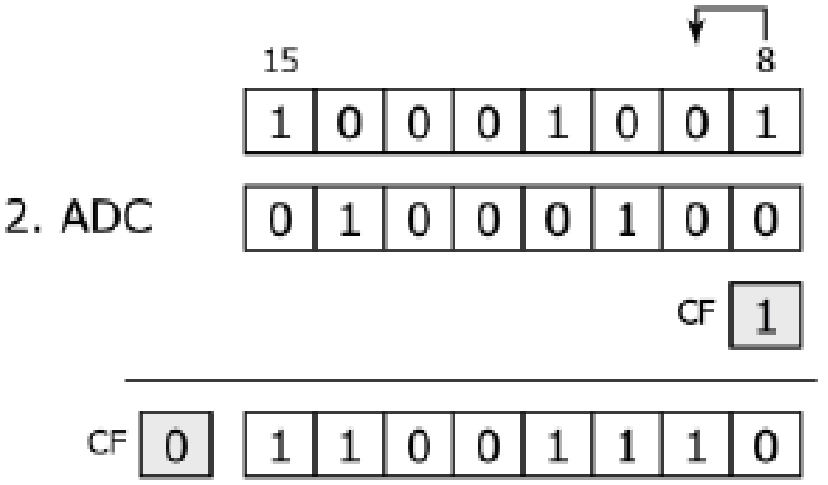
- Синтаксис:
  ADC Приймач, Джерело
- Логіка роботи:
  <Приймач> = <Приймач> + <Джерело> + <CF>

ADD та ADC





Додавання двох 16-розрядних чисел
35276 + 17553 = 52829 командою ADD.



Додавання двох 16-розрядних чисел
35276 + 17553 = 52829 командою ADC.

Команда INC

- Синтаксис:
  INC Операнд
- Логіка роботи команди:
  <Операнд> = <Операнд> + 1

- Наприклад:
  inc I ; I++
  inc ax ; <ax> = <ax> + 1
  inc cl ; <cl> = <cl> + 1

Команда SUB

- Синтаксис:
  SUB Приймач, Джерело
- Приклад використання команди:
  mov al, 10
  sub al, 7 ; al = 3; al – приймач, 7 – джерело.

Команда SBB

- Синтаксис:
  SBB Приймач, Джерело
- Логіка роботи:
  <Приймач> = <Приймач> – <Джерело> – <CF>

Команда DEC

- Синтаксис:
  DEC Операнд
- Вона еквівалентна команді:
  SUB Приймач, 1
- Логіка роботи команди:
  <Операнд> = <Операнд> - 1

- Приклад використання команди:
  mov al, 15
  dec al; al = 14

Команда NEG

- Синтаксис:
  NEG Операнд
- Логіка роботи команди:
  <Операнд> = - <Операнд>

- Наприклад, для обчислення <AL> = =50 - <AL> можна використати такий код:
  neg al
  add al, 50

Розрахунок 16 бітних виразів

- Розрахувати значення виразу:

x = (a + b +10) – (b+c+d-e) + 20,
де x, a, b, c, d, e – 16-бітні знакові.

- Потрібно звернути увагу, що у нас обмежені ресурси, і ми маємо лише чотири 16-бітні регістри (AX, BX, CX, DX).

Операнди множення та ділення

- При виконанні операції множення операнди повинні мати однакову довжину, а результат довжину в два рази більшу:

  8 біт * 8 бітів = 16 бітів
  16 бітів * 16 бітів = 32 бітів
  32 біти * 32 біти = 64 біти
  64 біти * 64 біти = 128 біти

- При виконанні операції ділення ділене повинне мати довжину в два рази більше ніж дільник та частка:

  16 бітів : 8 бітів = 8 бітів
  32 біти : 16 бітів = 16 бітів
  64 біти : 32 біти = 32 біти
  128 бітів : 64 біти = 64 біти

Ресурси

- Операції множення та ділення «прив’язані» до регістрів:

  8 бітне множення та ділення – регістри AL, AX
  16 бітне множення та ділення – регістри AX, <DX:AX> (EAX)
  32 бітне множення та ділення – регістри EAX, <EDX:EAX> (RAX)

Команди перетворення типів

- **cbw** – розширює регістр AL до регістра AX
- **cwd** – розширює регістр AX до комбінованого регістра <DX:AX> (EAX)
- **cwde** – розширює регістр AX до регістра EAX
- **cdq** – розширює регістр EAX до комбінованого регістра <EDX:EAX> (RAX)

- Синтаксис:
  CBW
  CWD
  CWDE
  CDQ

Приклад перетворення 8-бітного значення в 16-бітне

- Перетворення 8-бітного значення в 16-бітне +/- 7
- Для додатнього «7»
- AL: 00000111
- AX: AH:00000000 AL:00000111
- Для від’ємного «-7» (зберігається в додатковому коді)
- AL: 11111001 – 8 бітів знакове
- AX: AH: 11111111 AL:11111001 – 16 бітів знакове

Команди множення MUL та IMUL

- Беззнакове множення, синтаксис:
  MUL Операнд 2 (Джерело)
- Знакове множення, синтаксис:
  IMUL Операнд 2 (Джерело)

Неявні операнди команд MUL, IMUL

Довжина джерела	Множник	Добуток
Byte	AL AX	(<AH:AL>)
Word	AX	<DX:AX>
DWord	EAX	<EDX:EAX>

Логіка роботи команд множення

- Логіка роботи команди аналогічна в обох команд:
  Добуток = Множене х Операнд 2 (Джерело)
- **Множене** – знаходиться в регістрі AL (8 бітів), AX (16 бітів), EAX (32 біти)
- **Добуток** – зберігається в регістрі AX (пара регістрів AH:AL для 8-бітного множення), DX:AX (16 бітне множення) і, відповідно, EDX:EAX (32 бітне множення).
- **Операнд 2 (Джерело)** – або регістр, або пам’ять (змінна), константи (числа) не допускаються
- Довжина Добутку завжди в два рази більша, ніж у Множника. Причому старша частина Добутку знаходиться або в регістрі AH, або в DX або в EDX.
- Ці команди також виробляють прапорці CF=1 та OF=1. Але вони встановлюються тільки тоді, коли результат занадто великий для відведених йому регістрів призначення.

Команди ділення DIV та IDIV

- Беззнакове ділення, синтаксис:
  DIV Операнд 2
- Знакове ділення, синтаксис :
  IDIV Операнд 2

- **Неявні операнди команд DIV, IDIV**

Довжина джерела (Дільника)	**Ділене**	**Добуток**	
		Частка	Залишок
Byte	AX(<AH:AL>)	AL	AH
Word	<DX:AX>	AX	DX
DWord	<EDX:EAX>	EAX	EDX

Логіка роботи команд ділення

- Логіка роботи команди:
  <Частка:Залишок>=<Ділене> / Операнд 2 (Джерело_Дільник).
- **Ділене** – знаходиться в регістрі AX (8-бітне ділення), DX:AX (16 бітне ділення).
- **Результат** – зберігається в регістрі AL (8-бітне ділення), AX (16 бітне ділення)
- **Операнд 2** – або регістр, або пам’ять (змінна), константи (числа) не допускаються.
- Довжина діленого завжди в два рази більша, ніж у Дільника. Причому старша частина Діленого знаходиться в регістрі AH, або в DX, або в EDX.
- Ці команди також виробляють прапорці CF=1 та OF=1. Але вони встановлюються коли частка не поміщається в регістри AL, AX або EAX. Тут, на відміну від команд множення, завжди генерується переривання «Ділення на нуль», хоча на нуль і не ділимо.

Додаткові команди

- SHL (Logical Shift Left), SAL (Arithmetic Shift Left), SHR, SAR – команди логічного та арифметичного зсуву ліворуч та праворуч на певну кількість бітів
- Можуть застосовуватися для беззнакових чисел для виконання множення/ділення на 2^x(2, 4, 8).
- $00000100_2 = 4_{10}$
- Операція SAL AL, 2
- $00010000_2 = 16_{10}$
- ---
- $00000100_2 = 4_{10}$
- Операція SAR AL, 2
- $00000001_2 = 1_2$
-

- Для знакових чисел в командах логічного та арифметичного зсуву вправо є різниця в збереженні знаку.
- $11111001_2 = -7_{10}$ (в додатковому коді) (або 249_{10} в прямому)
- SHR AL, 1; сприймає дані як беззнакове число в прямому коді
- $01111100_2 = 124_{10}$
- SAR AL, 1; сприймає дані як від’ємне число в додатковому коді
- 10000111_2 –> $10000011_2 = -4_{10}$ –> 11111101_2
- При зміщенні додатнього знакового числа вліво ми отримуємо не коректний результат, оскільки втрачаємо старший біт

Приклади розрахунку складних виразів

- Приклад 1.
  Розрахувати значення виразу x = (a+5)/b + c/d, всі числа 8-бітні знакові

- Приклад 2.
  Розрахувати значення виразу x = (a+d+5)/b + a/(d*2), всі числа 16-бітні знакові.

Дякую за увагу!

Список використаних джерел

- 1. Тараракa В.Д. Архітектура комп'ютерних систем: навчальний посібник. Житомир: ЖДТУ, 2018. 383 с.
- 2. David Harris, Sarah Harris. Digital Design and Computer Architecture. 2nd Edition. Morgan Kaufmann. 2012. 720p.
- 3. Assembler. Підручник для ВНЗ. В.І. Юров. 2003. 637 с.
- 4. Мистецтво програмування на асемблері. Лекції та вправи. Н.Г.Голуб. 2002. 656 с.