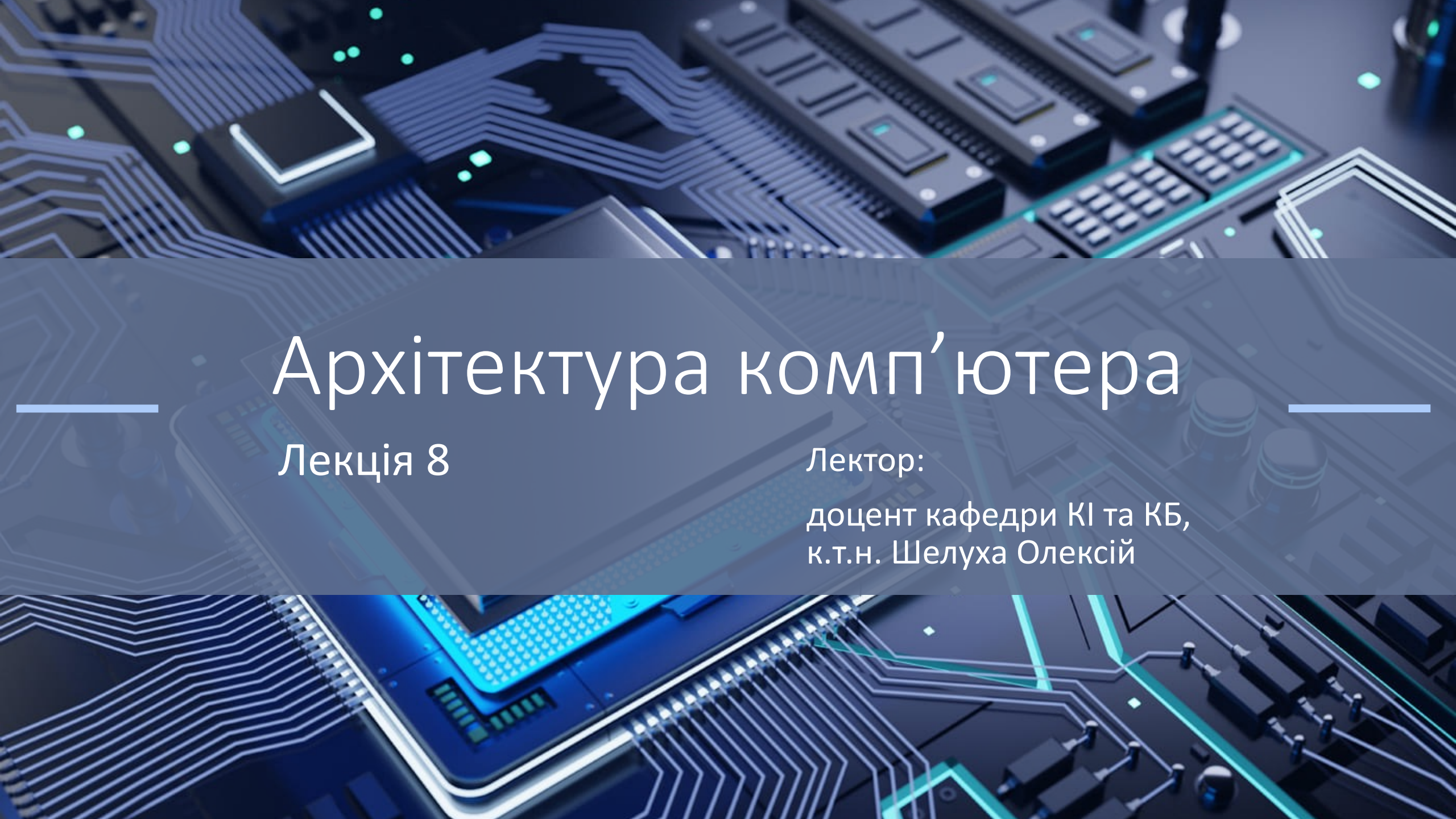


Архітектура комп’ютера

Лекція 8

Лектор:

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

Зміст лекції

- Безумовні переходи
- Умовні переходи
- Команди умовних переходів
- Приклад
- Нестандартні ситуації при виконанні арифметичних операцій та їх опрацювання в Ассемблер

Переходи в команді:

- Рішення про те, яка команда буде виконуватись наступною може бути:
  – безумовним: із визначеного місця необхідно передати управління не тій команді, яка йде наступною, а іншій, яка розміщується на деякому віддаленні від поточної команди;
  – умовним: рішення про те, яка команда буде виконуватись наступною, приймається за результатами аналізу деяких умов або даних.

- За принципом дії команди мікропроцесора, які забезпечують організацію переходів у програмі, можна поділити на три основні групи:
- 1. Команди безумовної передачі управління.
- 2. Команди умовної передачі управління.
- 3. Команди управління циклом.

Команди безумовного переходу

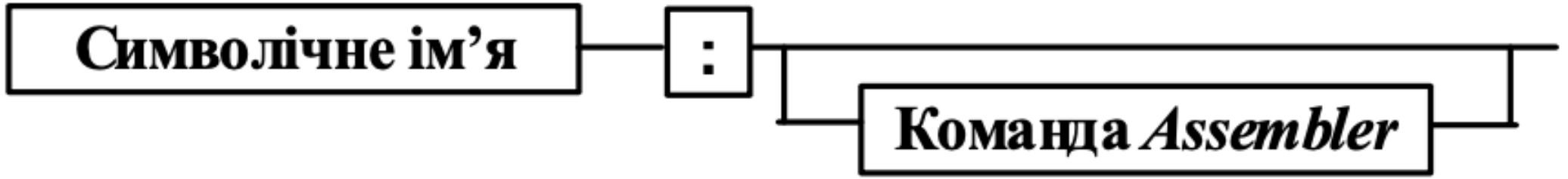
- команди безумовного переходу;
- виклик процедур та повернення з процедур;
- виклик програмних переривань та повернення з програмних переривань

Мітки

- **Мітка** – це символічне ім’я, яке позначає визначену комірку пам’яті, що призначена для використання як операнда в командах передачі управління. Тобто до якої потрібно перейти для продовження виконання коду програми.
- Основні атрибути:
  - – ім’я сегмента коду, де ця мітка описана;
  - – зміщення – відстань у байтах від початку сегмента коду, у якому описана мітка;
  - – тип мітки або атрибут відстані
- Атрибут відстані може набувати двох значень:
  - – near – перехід на цю мітку можливий лише в межах сегмента коду, де ця мітка описана. Фізично це означає, що для переходу на мітку достатньо змінити тільки вміст регістра eip/ip;
  - – far – перехід на цю мітку можливий тільки внаслідок міжсегментної передачі управління, для здійснення якої потрібна зміна вмісту як регістра eip/ip, так і регістра cs.

Визначення мітки

- – оператором : (двокрапка);
- – директивою label.



- Рис.1 – Синтаксис визначення мітки

- Команда Assembler може розташовуватись як в одному рядку з міткою, так і на наступному рядку. Таку мітку можна використовувати як операнд в командах умовного (jcc) та безумовного (jmp, call) переходів.

Приклад

```
__asm {
m1:  mov ax, bx
        inc bx
        …
        jmp m3
m2:
    mov ax, c
    imul ax
    …
    jmp ext
m3:  imul bx
        jmp m2
ext:
}
```

послідовність виконання команди m1-> m3-> m2-> ext

Команди умовної передачі управління

- Ці команди дають можливість перевірити виконання або не виконання певної умови або даних що сформувались перед виконанням переходу.

- – команди переходу за результатом виконання команди порівняння (CMP);
- – команди переходу за станом визначеного прапорця FLAGS/EFLAGS;
- – команди переходу за вмістом регістра ecx/cx.

Типова схема умовного переходу (if):

- порівняння за допомогою команди **CMP**
- перехід на мітку за результатами порівняння за допомогою команди Jc/Jcc/Jccc (c/cc/ccc – різні символи для різних порівнюваних значень).

Команда порівняння (CMP).

- Принцип роботи команди **CMP** (CoMPare) такий самий, як і команди віднімання **SUB** операнд_1, операнд_2. Команда cmp здійснює віднімання операндів і встановлює відповідні прапорці. Однак на відміну від команди sub, команда cmp не записує результат віднімання на місце першого операнда, а генеруються у вигляді відповідних ознак регістра FLAGS/EFLAGS

- Операнд 1 – регістр, змінна
- Операнд 2 – регістр, змінна, константа (число)

Синтаксис команди CMP:

- cmp операнд_1, операнд_2

- Команди умовних переходів **Jccc.**
- Літери команд «**ccc**» використовуються з наступної мнемоніки:

Позначення	Оригінал	Переклад	Тип операндів
e	Equal	дорівнює	операнди будь які
n	Not	не	операнди будь які
g	Greater	Більше	Числа зі знаком
l	Less	Менше	Числа зі знаком
a	Above	Вище (більше)	Числа без знаку
b	Below	Нижче (менше)	Числа без знаку

Приклад

Всі	Знакові	Беззнакові
!=	>=	>=
Не дорівнює	більше або дорівнює	більше або дорівнює
JNE	JGE (JNL)	JAE (JNB)
Jump if Not Equal	Jump if Greater or Equal	Jump if Above or Equal
	Jump if Not Less	Jump if Not Below

Перелік команд умовного переходу для команди CMP операнд_1, операнд_2

Тип операндів	Мнемокод команди	Критерій умовного переходу	Значення прапорців для здійснення переходу
Будь-які	je	операнд_1=операнд_2	zf = 1
Будь-які	je	операнд_1< >операнд_2	zf = 0
Зі знаком	jl / jnge	операнд_1<операнд_2	sf< > of
Зі знаком	jle / jng	операнд_1<=операнд_2	sf< > of or zf = 0
Зі знаком	jg / jnle	операнд_1>операнд_2	sf = of and zf = 0
Зі знаком	jge / jnl	операнд_1=>операнд_2	sf = of
Без знака	jb / jnae	операнд_1<операнд_2	cf = 1
Без знака	jbe / jna	операнд_1<=операнд_2	cf = 1 or zf = 1
Без знака	ja / jnbe	операнд_1>операнд_2	cf = 0 and zf = 0
Без знака	jae / jnb	операнд_1=>операнд_2	cf = 0

- Команди умовного переходу не змінюють прапорці, тому після команди **cmp** може бути декілька команд умовного переходу. Це може бути використано для того, щоб дослідити кожну з альтернатив: “більше”, “менше” або “дорівнює”.

- .....
- cmp ax, 5 *;(порівняти вміст* ax *з* 5*)*
- je eql *;(перехід, якщо* ax = 5)
- jl low *;(перехід, якщо* ax<5*)*
- jg grt *;(перехід, якщо* ax>5*)*
- egl: .....
- low: .....
- grt: .....

Команди умовного переходу та регістр ecx/cx

- Регістр ecx/cx виконує функцію лічильника в командах управління циклами й при роботі з ланцюгами символів.

- Синтаксис цієї команди умовного переходу такий:
- jcxz *мітка_переходу* (Jump if CX is Zero) – *перехід, якщо* cx *нуль*;
- jecxz *мітка_переходу* (Jump if ECX is Zero) – *перехід, якщо* ecx *нуль*.

Приклад

- Приклад. Розрахувати значення 16-бітного знакового виразу
- $$y = \begin{cases} \mathbf{1}, \mathbf{a} > \mathbf{b} \\ \mathbf{0}, \mathbf{a} = \mathbf{b} \\ \mathbf{-1}, \mathbf{a} < \mathbf{b} \end{cases}$$

```
MOV AX, a;
    MOV BX, b;
    CMP AX,BX ;
    JG m1; Перехід, якщо a>b
    JE m2 ; Перехід, якщо a=b
    JL m3 ; Перехід, якщо a<b
m1: MOV y,1
        JMP end
m2: MOV y, 0
        JMP end
m3: MOV y, -1
        JMP end
end:
```

Приклад

- Розрахувати приклад

- $y = \begin{cases} (a + b - 5) * c, a > b \\ 0, a = b \\ \frac{a+b-5}{c}, a < b \end{cases}$

Нестандартні ситуації при виконанні арифметичних операцій та їх опрацювання в Ассемблер

- Для цілочисельних арифметичних виразів:

1. Переповнення
2. Ділення на нуль
3. Вихід за межі діапазону

Переповнення

- a, b, y – 8 бітні беззнакові (0 … 255)
- y = a+b+1
- a = 150 – в діапазон потрапляє 0 … 255
- b = 180 – в діапазон потрапляє 0 … 255
- y = 150 + 180 !!! y = 330 переповнення

Ділення на нуль

- a, b, y – 8 бітні беззнакові (0 … 255)
- y = a/(b-1)
- a = 50 – в діапазон потрапляє 0 … 255
- b = 1 – в діапазон потрапляє 0 … 255
- y = 50/(1-1) !!! Ділення на нуль

Вихід за межі діапазону

- a, b, y – 8 бітні беззнакові (0 … 255)
- y = a – b – 1
- a = 5 – в діапазон потрапляє 0 … 255
- b = 6 – в діапазон потрапляє 0 … 255
- y = 5 – 6 !!! вихід за межі діапазону

Регістр ознак FLAGS/EFLAGS

31	...	21	20	19	18	17	16	15	14	13 12	11	10	09	08	07	06	05	04	03	02	01	00
0	...	ID	VIP	VIF	AC	VM	RF	0	NT	IOPL	OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF

Команди умовних переходів для нестандартних ситуацій

- Команди реагують на значення певних ознак (Flags).
  Виконання команди порівняння перед цими командами не обо’язкове, достатньо будь якої команди, що змінює ознаку.

Код команди	Стан прапорців	Код команди	Стан прапорців
JZ/JE	ZF=1	**JNZ/JNE**	ZF=0
JS	SF=1	**JNS**	SF=0
JC/JB/JNAE	CF=1	**JNC/JAE/JNB**	CF=0
JO	OF=1	**JNO**	OF=0
JP	PF=1	**JNP**	PF=0

Дякую за увагу!