

ДОДАТКОВА ЛАБОРАТОРНА РОБОТА №7*.

**виконується групами КІ-307, КІ-308 та КІ-309 у разі несвоєчасного виконання основних лабораторних*

Укладачі:

Мархивка В.С., ст. викладач,
Олексів М. В., асистент, к.т.н.,
Мороз І.В., ст. викладач, к.т.н.,
Акимішин О.І., доцент, к.т.н..

ОБЧИСЛЕННЯ ЕЛЕМЕНТАРНИХ ФУНКЦІЙ НА МАТЕМАТИЧНОМУ СПІВПРОЦЕСОРІ

Мета: познайомитися з принципами роботи математичного співпроцесора та оволодіти навиками використання вбудованих елементарних математичних функцій та реалізації розгалужень.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Команди порівняння чисел

У центральному процесорі команди умовних переходів виконуються відповідно до значень окремих бітів регістра прапорців процесора. У арифметичному співпроцесорі існують спеціальні команди порівнянь, за наслідками виконання яких, встановлюються біти кодів умов в регістрі стану:

FCOM – порівняння;

FICOM – порівняння цілих чисел;

FCOMP - порівняння дійсних чисел і вилучення зі стеку;

FICOMP - порівняння цілих чисел і вилучення зі стеку;

FCOMPP - порівняння і подвійне вилучення зі стеку (ST(0), ST(1));

FTST - порівняння операнда з нулем;

FXAM - аналіз операнда на тип числа (скінчене число, денормалізоване число, нуль, некінечність, ...).

Команда FCOM віднімає вміст операнда, розміщеного в оперативній пам'яті, від значення у вершині стеку ST(0). Результат віднімання нікуди не записується і покажчик вершини стеку ST не змінюється.

Позначимо операнд команди порівняння як "x". Нижче, приведемо значення бітів кодів умови після виконання команди "FCOMx":

$C3 = 0, C0 = 0 \quad ST(0) > x$

$C3 = 0, C0 = 1 \quad ST(0) < x$

$C3 = 1, C0 = 0 \quad ST(0) = x$

$C3 = 1, C0 = 1 \quad ST(0) \text{ і } x \text{ непорівнювані}$

Остання комбінація виникає при спробі порівняння не чисел, невизначеностей або нескінченості, а також в деяких інших випадках.

Операндами команди FICOM є 16- або 32-розрядні цілі числа, а в решті - аналогічна команді FCOM.

Команди FCOMP і FICOMP аналогічні, відповідно, командам FCOM і FICOM, за винятком того, що після виконання операнд вилучається зі стеку.

Команда FCOMPP виконує ті ж дії, що і FCOM, але вона після виконання вилучає зі стеку обидва операнди, що брали участь в порівнянні.

Команда FTST призначена для порівняння операнду з нулем. Після її виконання коди умов встановлюються згідно з наведеним нижче:

$C3 = 0, C0 = 0$ ST(0) > 0

$C3 = 0, C0 = 1$ ST(0) < 0

$C3 = 1, C0 = 0$ ST(0) = 0

$C3 = 1, C0 = 1$ ST(0) і 0 непорівнювані

Команда FXAM аналізує вміст ST(0). Після її виконання встановлюються коди умов, згідно яких можна визначити знак числа, його скінченність або нескінченність, нормалізованість і т.д. Біт C1 містить знак числа, що аналізується: 0 - додатний, 1 - від'ємний. За допомогою біта C0 можна визначити, є число скінченим або нескінченим: 0 - скінчене число, 1 - нескінчене. Для скінчених чисел подальша класифікація може проводитися за вмістом кодів умов C2 і C3:

$C3 = 0, C0 = 0$ Ненормалізоване число

$C3 = 0, C0 = 1$ Нормалізоване число

$C3 = 1, C0 = 0$ Нульове число

$C3 = 1, C0 = 1$ Число денормалізоване

Аналогічно, для нескінчених чисел коди умов C2 і C3 мають наступні значення:

$C3 = 0, C0 = 0$ Нечисло

$C3 = 0, C0 = 1$ Нескінчене число

$C3 = 1, C0 = 0$ Порожнє число

$C3 = 1, C0 = 1$ Порожнє число

Для реалізації порівняння чисел необхідно за допомогою команди "FSTSW AX" переписати вміст регістра стану співпроцесора в регістр AX центрального процесора. Далі вміст регістра АН переписати в регістр прапорів центрального процесора за допомогою команди SAHF. Біти кодів умов співпроцесора відображаються (проектуються) на регістр прапорів центрального процесора так, що без додаткових дій, можна використовувати команди умовних переходів, але тільки беззнакового типу.

Наприклад, в наступному фрагменті програми виконується перехід до мітки compute, якщо операнди рівні:

```
.586
...
fcom
fstsw ax
sahf
je compute
...
```

Трансцендентні команди

Трансцендентні команди призначені для обчислення наступних функцій:

- тригонометричні (sin, cos, tg...)
- зворотні тригонометричні (arcsin, arccos...)
- показникові (x^y , 2^x , 10^x , e^x)
- гіперболічні (sh, ch, th...)
- зворотні гіперболічні (arsh, arch, arcth...)

Ось список всіх трансцендентних команд математичного співпроцесора:

FPTAN - обчислення часткового тангенса;

FPATAN - обчислення часткового арктангенса;

FYL2X - обчислення $y \cdot \log_2(x)$;

FYL2XP1 - обчислення $y \cdot \log_2(x+1)$;

F2XM1 - обчислення $2^x - 1$;

FCOS - обчислення $\cos(x)$;

FSIN - обчислення $\sin(x)$;

FSINCOS - обчислення $\sin(x)$ і $\cos(x)$ одночасно.

Команда FPTAN обчислює частковий тангенс $ST(0)$, розміщуючи в стеку два числа x та y , такі що $y/x = tg(ST(0))$. Для сучасних співпроцесорів $x = 1$, а $y = tg(ST(0))$.

Після виконання команди число y розташовується в $ST(0)$, а число x заноситься у вершину стеку (тобто записується в $ST(1)$). Для старих співпроцесорів (Intel 8087, 80287) аргумент команди FPTAN повинен бути нормалізованим і знаходиться в межах: $0 \leq ST(0) \leq \pi/4$.

Таблиця 4.1.

Результати обчислень інструкції FPTAN

ST(0) джерело	ST(0) приймач
$-\infty$ або $+\infty$	виняткова ситуація
$-R$ або $+R$	від $-R$ до $+R$
-0	-0
+0	+0

нечисло	Нечисло
Примітка: R – скінченне дійсне число	

Для нових співпроцесорів аргумент команди FPTAN повинен бути поданий в радіанах в межах $\pm 2^{63}$. Результати обчислень в залежності від вхідних даних подані у таблиці нижче.

Якщо аргумент є більшим або меншим, ніж $\pm 2^{63}$, то тангенс необхідно шукати сумісним використанням команди FPREM і FPTAN перевіряючи прапорець C2 до повного знаходження значення тангенсу.

Користуючись знайденим значенням часткового тангенса, можна обчислити інші тригонометричні функції за наступними формулами:

$$\sin(z) = 2 \cdot (y/x) / (1 + 2 \cdot (y/x));$$

$$\cos(z) = (1 - 2 \cdot (y/x)) / (1 + 2 \cdot (y/x));$$

$$\operatorname{tg}(z/2) = y/x;$$

$$\operatorname{ctg}(z/2) = x/y;$$

$$\operatorname{cosec}(z) = (1 + 2 \cdot (y/x)) / 2 \cdot (y/x);$$

$$\sec(z) = (1 + 2 \cdot (y/x)) / (1 - 2 \cdot (y/x)).$$

Де z - значення, що знаходилося в ST(0) до виконання команди FPTAN, x і y - значення в регістрах ST(0) і ST(1), відповідно.

Команда FPATAN обчислює частковий арктангенс $z = \operatorname{arctg}(\operatorname{ST}(1)/\operatorname{ST}(0)) = \operatorname{arctg}(x/y)$. Перед виконанням команди числа x і y розташовуються в ST(1) і ST(0), відповідно. Аргументи команди FPATAN повинні знаходитися в межах: $0 \leq x < y < +\infty$. Результат записується в ST(1), а операнд в ST(0) вилучається зі стеку. Таким чином результат після виконання інструкції розміщується у ST(0).

Команда FYL2X обчислює вираз $y \cdot \log_2(x)$, операнди x і y розміщуються, відповідно в ST(0) та ST(1). Операнди вилучаються зі стеку, а результат записується в стек. Параметр x повинен бути додатнім числом.

Користуючись результатом виконання цієї команди, можна обчислити логарифмічні функції, наступним чином:

$$\text{Логарифм за основою два: } \log_2(x) = \text{FYL2X}(x).$$

Натуральний логарифм:

$$\log_e(x) = \log_e(2) \cdot \log_2(x) = \text{FYL2X}(\log_e(2), x) = \text{FYL2X}(\text{FLDLN2}, x).$$

Десятковий логарифм:

$$\log_{10}(x) = \log_{10}(2) \cdot \log_2(x) = \text{FYL2X}(\log_{10}(2), x) = \text{FYL2X}(\text{FLDLG2}, x).$$

Функція FYL2XP1 обчислює вираз $y \cdot \log_2(x + 1)$, де x відповідає ST(0), а y - ST(1). Результат записується в ST(0), обидва операнди вилучаються зі стеку та втрачаються. На операнд x накладається обмеження: $0 < x < 1 - 1 / \sqrt{2}$.

Команда F2XM1 обчислює вираз $2x - 1$, де x - ST(0). Результат записується в ST(0), параметр повинен знаходитися в наступних межах: $0 \leq x \leq 0,5$.

Команда FCOS обчислює $\cos(x)$. Параметр x винен знаходитися в ST(0), туди ж записується результат виконання команди.

Команда FSIN аналогічна команді FCOS, але обчислює значення синуса ST(0).

Команда FSINCOS обчислює одночасно значення синуса і косинуса параметра ST(0). Значення синуса записується в ST(1), косинуса - в ST(0).

Команди керування

Команди, керування, призначені для роботи з нечисловими регістрами співпроцесора. Деякі команди мають альтернативні варіанти. Мнемоніки цих команд можуть починатися з FN або F. Перший варіант відповідає командам “Без очікування”. Для таких команд процесор не перевіряє, чи зайнятий співпроцесор виконанням команди, тобто біт зайнятості В не перевіряється. Особливі випадки також ігноруються.

Варіанти команд “З очікуванням” діють так само, як і звичайні команди співпроцесора.

Ось список команд керування для співпроцесора:

FNSTCW (FSTCW) - записати управляюче слово (записує вміст регістра управління в оперативну пам'ять).

FLDCW - завантажити управляюче слово (завантажує регістр управління з оперативної пам'яті і, як правило, використовується для зміни режиму роботи співпроцесора).

FNSTSW (FSTSW) - записати слово стану (записує вміст регістра стану в оперативну пам'ять).

FNSTSW AX (FSTSW AX) - записати слово стану в AX (записує вміст регістра стану в регістр AX центрального процесора, де можливий аналіз вмісту за допомогою команд умовних переходів).

FNCLEX (FCLEX) - скинути особливі випадки (скидає прапорці особливих випадків в регістрі стану співпроцесора, також скидаються біти ES і В).

FNINIT (FINIT) - ініціалізувати співпроцесор (ініціалізує регістр стану, регістр управління, і регістр тегів таким чином:

регістр управління - проектна нескінченність, округлення до найближчого, розширена точність, всі особливі випадки замасковані;

регістр стану - В=0 (біт зайнятості скинутий), код умови не визначений, ST=ES=0, прапорці особливих випадків встановлені в нуль;

регістр тегів - усі поля регістру тегів містять значення 11 (порожній регістр));

FNSTENV (FSTENV) - записати оточення (записує в пам'ять вміст всіх регістрів, окрім числових, у визначеному форматі. Команда корисна при обробці особливих випадків);

FLDENV - завантажити оточення (завантажує регістри, збережені командою FNSTENV);

FNSAVE (FSAVE) - записати повний стан (діє аналогічно команді FNSTENV, але додатково зберігає вміст числових регістрів);

FRSTOR - відновити повний стан (діє аналогічно команді FLDENV, але додатково відновлює вміст числових регістрів);

FINCSTP - збільшити показчик стека SP на 1;

FDECSTP - зменшити показчик стека SP на 1;

FFREE - звільнити регістр (визначає числовий регістр ST, вказаний як операнд, як порожній, записуючи у відповідне поле регістра тегів значення 11);

FNOP - порожня команда, немає операції (не робить жодних дій);

FSETPM - встановлює захищений режим роботи (переводить співпроцесор в захищений режим роботи).

КОНТРОЛЬНІ ПИТАННЯ

1. Для чого використовується математичний співпроцесор?
2. Які формати даних може обробляти математичний співпроцесор?
3. Які числові регістри має співпроцесор?
4. Які регістри керування має співпроцесор?
5. Які групи команд має співпроцесор?
6. Які арифметичні команди співпроцесора ви знаєте?
7. Які команди керування співпроцесора ви знаєте?
8. Які трансцендентні команди FPU ви знаєте?

ЛІТЕРАТУРА

1. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC XT и AT. - М. "Финансы и статистика", 1992, стор. 13-31.
2. Березко Л.О., Троценко В.В. Особливості програмування в турбо-асемблері. - Київ, НМК ВО, 1992.
3. Дао Л. Программирование микропроцессора 8088. Пер. с англ. - М.: "Мир", 1988.
4. Абель П. Язык ассемблера для IBM PC и программирования. Пер. з англ. - М., "Высшая школа", 1992.
5. Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 2 (2A & 2B): Instruction Set Reference, A-Z. – 2011. Режим доступу:

ЗАВДАННЯ

- Створити *.exe програму, яка реалізовує обчислення, заданого варіантом виразу. Вхідні дані повинні вводитися з клавіатури, як дійсні числа. Програма повинна складатися з двох модулів:
головний модуль – створюється мовою C і має забезпечити ввід необхідних даних, виклик асемблерної процедури для обчислення виразу та вивід результату обчислень;
модуль безпосередніх обчислень – здійснює всі необхідні арифметичні дії з використанням математичного співпроцесора.
- Переконалися у правильності роботи кожного модуля зокрема та програми загалом.
- Скласти звіт про виконану роботу з приведенням тексту програми та коментарів до неї.
- Дати відповідь на контрольні запитання.

Таблиця 4.1.

Варіанти завдань

варіант	Завдання	Варіант	Завдання
1.	$X_i = \begin{cases} \frac{2 * c - d + \sqrt{23 * a_i}}{4} & c > d \\ \frac{a_i - 1}{c + 4 * d - \sqrt{123 * c}} & c \leq d \end{cases}$	16.	$X_i = \begin{cases} \frac{a_i + \frac{c}{d} - \sqrt{28 * d}}{4 * b * a + 1} & c > d \\ \frac{\frac{c}{d} - \sqrt{24 + d} + a_i}{2 * a_i * c - 4} & c \leq d \end{cases}$
2.	$X_i = \begin{cases} \frac{-2 * c + d * 82}{\lg(\frac{a_i}{4} - 1)} & c > d \\ \frac{\lg(2 * c - a_i) + d - 152}{\frac{a_i}{4} + c} & c \leq d \end{cases}$	17.	$X_i = \begin{cases} \frac{2 * c - \ln(a_i + d) * c}{c / 4 - 1} & c > d \\ \frac{41 - d / 4 - 1}{c / \lg(d + a_i) - d} & c \leq d \end{cases}$
3.	$X_i = \begin{cases} \frac{\lg(a_i + c / 4) - 12 * d}{a_i * b - 1} & c > d \\ \frac{-2 * c - \sin(a_i / d) + 53}{\frac{a_i}{4} - b} & c \leq d \end{cases}$	18.	$X_i = \begin{cases} \frac{a - c * 4 - 1}{c / 31 + \lg(a_i * d)} & c > d \\ \frac{\lg(d / a_i + 4) + d}{c - d + 1} & c \leq d \end{cases}$

4.	$X_i = \begin{cases} \frac{2 * c - \lg(d/4)}{a_i * a_i - 1} & c > d \\ \frac{\lg(c) - d * 23}{2 * a_i - 1} & c \leq d \end{cases}$	19.	$X_i = \begin{cases} \frac{\lg(21 - a_i) * c/4}{1 + c/a_i + d} & c > d \\ \frac{c - \ln(33 + d)/4}{a_i - c/d - 1} & c \leq d \end{cases}$
5.	$X_i = \begin{cases} \frac{2 * c - d/23}{\ln(1 - \frac{a_i}{4})} & c > d \\ \frac{4 * c + d - 1}{c - \lg \frac{a_i}{2}} & c \leq d \end{cases}$	20.	$X_i = \begin{cases} \frac{2 * b - 38 * c}{\arctg(d + a_i)/c + 1} & c > d \\ \frac{\arctg(c/4 + 28) * d}{a_i/d - c - 1} & c \leq d \end{cases}$
6.	$X_i = \begin{cases} \frac{2 * c - d * \sqrt{\frac{42}{d}}}{c + a_i - 1} & c > d \\ \frac{\sqrt{\frac{25}{c}} - d + 2}{d + a_i - 1} & c \leq d \end{cases}$	21.	$X_i = \begin{cases} \frac{a_i * c/4 - 1}{\sqrt{41 - d} - c * a_i} & c > d \\ \frac{1 + a_i - d/2}{\sqrt{24 + d - c} + a_i} & c \leq d \end{cases}$
7.	$X_i = \begin{cases} \frac{\arctg(c - d/2)}{2 * a_i - 1} & c > d \\ \frac{4 * \lg(c) - d/2 + 23}{a_i * a_i - 1} & c \leq d \end{cases}$	22.	$X_i = \begin{cases} \frac{\ln(a_i * d + 2) * c}{41 - d/c + 1} & c > d \\ \frac{\lg(4 * d - c) * a_i}{c/d - 1} & c \leq d \end{cases}$
8.	$X_i = \begin{cases} \frac{c * \lg(d + 23)}{a_i/2 - 4 * d - 1} & c > d \\ \frac{c/d + \ln(3 * a_i/2)}{c - a_i + 1} & c \leq d \end{cases}$	23.	$X_i = \begin{cases} \frac{2 * c + \lg(a_i - 21)}{c/a_i + d + 1} & c > d \\ \frac{4/c + \lg(3 * a_i)}{c/a_i - d - 1} & c \leq d \end{cases}$
9.	$X_i = \begin{cases} \frac{2 * c + \lg(d) * 51}{d - a_i - 1} & c > d \\ \frac{2 * c + \ln(d/4) + 23}{a_i * a_i - 1} & c \leq d \end{cases}$	24.	$X_i = \begin{cases} \frac{8 * \lg(d - 1) - c}{a_i * 2 + d/c} & c > d \\ \frac{4 * \ln(a_i/d) + 1}{c - d + a_i} & c \leq d \end{cases}$
10.	$X_i = \begin{cases} \frac{42 * c - d/2 + 1}{a_i * a_i - \ln(d - 5)} & c > d \\ \frac{\arctg(2 * c)/d + 2}{d - a_i - 1} & c \leq d \end{cases}$	25.	$X_i = \begin{cases} \frac{4 * \ln(d)/c + 1}{2 * c + a_i - c} & c > d \\ \frac{\arctg(d - c)/d + a_i/4}{a_i + c - 1} & c \leq d \end{cases}$
11.	$X_i = \begin{cases} \frac{\arctg(12/c) + 73}{a_i * a_i - 1} & c > d \\ \frac{2 * c/a_i - d * d}{d + \lg(a_i - 1)} & c \leq d \end{cases}$	26.	$X_i = \begin{cases} \frac{\arctg(a_i - c) * d + 28}{4 * c/a_i + 1} & c > d \\ \frac{c * d - 24 + a_i}{d - \lg(3 * c)} & c \leq d \end{cases}$

12.	$X_i = \begin{cases} \frac{\sqrt{\frac{53}{a_i} + d - 4 * a_i}}{1 + a_i * c} & c > d \\ \frac{\sqrt{15 * a_i + b - \frac{a_i}{4}}}{c * d - 1} & c \leq d \end{cases}$	27.	$X_i = \begin{cases} \frac{\frac{a_i + d/c}{a_i - c/4 + 1}}{\sqrt{\frac{41 * d}{a_i}} + 1} & c > d \\ \frac{1}{a_i - c/d} & c \leq d \end{cases}$
13.	$X_i = \begin{cases} \frac{-25/a_i + c - tg(c)}{1 + c * d/2} & c > d \\ \frac{lg(4 * a_i - 1) + d/2}{d * c - 5} & c \leq d \end{cases}$	28.	$X_i = \begin{cases} \frac{\frac{a_i + tg(d/4 - 1)}{c/3 - a_i * d}}{c * a_i + c/2} & c > d \\ \frac{1}{c - tg(d + 1)} & c \leq d \end{cases}$
14.	$X_i = \begin{cases} \frac{8 * lg(d + 1) - c}{a_i/2 + d * c} & c > d \\ \frac{4 * c - ln(d - 1)}{a_i/d + 18} & c \leq d \end{cases}$	29.	$X_i = \begin{cases} \frac{lg(25 + 2 * a_i)/d}{c + a_i - 1} & c > d \\ \frac{c + 23 - d + 4}{a_i - ln(a_i + c/d)} & c \leq d \end{cases}$
15.	$X_i = \begin{cases} \frac{arctg(4 * d)/c - 1}{12 + a_i - c} & c > d \\ \frac{arctg(c) + c * d - a_i/4}{a_i * d - 1} & c \leq d \end{cases}$	30.	$X_i = \begin{cases} \frac{d/2 - 53/c}{arctg(d - a_i) * c + 1} & c > d \\ \frac{c * 4 + 28 * d/c}{5 - arctg(a_i * d)} & c \leq d \end{cases}$

Примітка: a_i – елементи масиву дійсних чисел подвійної точності; c та d – дійсні числа одинарної точності.

ПРИКЛАД ВИКОНАННЯ

Завдання: Написати програму, яка обчислює арифметичний вираз над знаковими даними, введеними в десятковій системі і результат виводить на екран в десятковій формі зі знаком.

Знайти результат обчислення виразу:

$$y = \begin{cases} \cos(3\pi \cdot x), & x > 3 \\ \sin(3\pi \cdot x), & x \leq 3. \end{cases}$$

Текст програми, яка реалізовує завдання, приведений нижче. Програма передбачає ввід/вивід дійсних чисел з використанням функції C. Далі, застосовуються можливості арифметичного співпроцесора для обчислень всіх арифметичних дій. Для цього в сегменті даних передбачаємо змінні, що будуть використовуватися для обчислень. Вони повинні бути 4 або 8 байтними (згідно форматів даних для співпроцесора).

Програма обчислення трансцендентних функцій з використанням математичного співпроцесора

main.cpp

calc.asm

```
#include <stdio.h>
```

```
.386
```

```

extern "C" float calc(float);

int main()
{
    float x=0;
    float res=0;
    printf("Enter numbers:\n");
    printf("X = ");
    scanf("%f",&x);
    res=calc(x);
    printf("Result is: %f\n",res);
    return 0;
}

.model flat, c
.data
    const_3 dd 3.0
.code
calc PROC
    push ebp
    mov ebp, esp
    finit
    fld dword ptr[ebp+8]
    fcomp const_3
    fstsw ax
    sahf
    fldpi
    fmul
    fmul const_3
    ja __cos
    fsin
    jmp __next
__cos:
    fcos
__next:
    pop ebp
    ret
calc ENDP
END

```