

Лабораторна робота № 7

ОБЧИСЛЕННЯ АРИФМЕТИЧНИХ ВИРАЗІВ І ТРАНСЦЕНДЕНТНИХ ФУНКЦІЙ (СПІВПРОЦЕСОР i87)

Мета заняття: ознайомитися з основними командами мови Assembler для обчислення арифметичних виразів і трансцендентних функцій; набути практичних навичок в написанні програм, які обчислюють заданий змішаний арифметичний вираз, використовуючи арифметичні операції співпроцесора на мові Assembler.

Теоретичні відомості

Математичний співпроцесор (FPU – Floating Point Unit) – спеціальний пристрій, який слугує для обробки дійсних чисел (чисел з плаваючою комою).

З моменту свого виникнення співпроцесор розширював обчислювальні можливості основного процесора i8086 (i80286, i82386, i80486) і спочатку був виконаний у вигляді окремої мікросхеми i8087 (i80287, i80387, i80487). Його присутність в перших моделях процесора була не обов'язковою. Починаючи з сімейства процесорів i486DX співпроцесор став складовою частиною основного процесора.

Сучасний співпроцесор забезпечує повну підтримку стандартів IEEE-754 і IEEE-854 по представленню і обробці чисел з плаваючою комою. Він може виконувати трансцендентні операції (обрахування тригонометричних функцій, логарифмів і т.д.) з більшою точністю.

Типи даних

Співпроцесор може виконувати операції з сімома звичайними типами даних, а також із спеціальними числами.

Звичайні дані можуть бути дійсними або цілими числами із знаком. Наявність цілочисельних даних дозволяє виконувати так звані змішані обрахування, коли в якості операндів використовуються і цілі, і дійсні числа.

Співпроцесор виконує всі обрахунки в 80-бітному розширеному дійсному форматі, а 16-, 32- і 64-бітні дані використовуються для обміну даними і можуть застосовуватись в командах співпроцесора як операнди.

Таблиця 8.1

Типи звичайних даних співпроцесора

Тип даних	Довжина (біт)	К-сть значимих цифр	Діапазон представлення
Ціле слово	16	4-5	-32768...32767
Коротке ціле	32	9-10	-2147483648 ... -2147483647
Довге ціле	64	18-19	-9223372036854775808 ... 9223372036854775807
Упаковане двійково-десятькове	80	18	-999999999999999999 ... 999999999999999999
Коротке дійсне	32	7-8	1.175494351e-38 ... 3.402823466e+38
Довге дійсне	64	15-16	2.2250738585072014e-308... 1.7976931348623158e+308
Розширене дійсне	80	19-20	3.3621031431120935063e-4932... 1.189731495357231765e+4932

Крім звичайних чисел, специфікація стандарту IEEE передбачає декілька спеціальних форматів, які можуть виникнути в результаті виконання математичних операцій співпроцесора і над якими можна виконувати окремі операції.

- **Додатний нуль** – всі біти числа скинуті до нуля.
- **Від’ємний нуль** – знаковий біт дорівнює 1, всі інші біти числа скинуті до нуля.
- **Додатня нескінченність** – знаковий біт дорівнює 0, всі біти експоненти установлені в 1, а біти мантиси скинуті до нуля.
- **Від’ємна нескінченність** - знаковий біт дорівнює 1, всі біти експоненти установлені в 1, а біти мантиси скинуті до 0.
- **Денормалізовані числа** – всі біти експоненти скинуті до 0. Ці числа дозволяють представляти дуже маленькі числа при обрахунках з розширеною точністю.

- **Невизначеність** – знаковий біт дорівнює 1, перший біт мантиси дорівнює 1 (для 80-розрядних чисел перші два біта дорівнюють 11), інші біти мантиси скинуті до нуля, всі біти експоненти встановлені в 1.

- **Не-число типу SNAN (сигнальне)** – перший біт мантиси дорівнює 0 (для 80-розрядних чисел перші два біти дорівнюють 10), інші біти мантиси можуть містити 0 або 1, всі біти експоненти встановлені в 1.

- **Не-число типу QNaN (тихе)** – перший біт мантиси дорівнює 1 (для 80-розрядних чисел перші два біти дорівнюють 11), інші біти мантиси можуть містити 0 або 1, всі біти експоненти встановлені в 1.

- **Непідтримуване число** – всі інші ситуації.

Регістри

У співпроцесора є 8 регістрів для зберігання даних R0-R7 і 5 допоміжних регістрів, які складають середовище співпроцесора.

Регістри даних співпроцесора R0-R7 мають довжину 80 біт (тобто 16-розрядних слів) і розглядаються як круговий стек, вершина якого (TOP) називається ST або ST(0) і є плаваючою.

Принцип роботи з круговим стеком співпроцесора аналогічний звичайному калькулятору. Будь-яка команда завантаження даних співпроцесора автоматично переміщує вершину стеку співпроцесора: $TOP = TOP + 1$.

В таблиці 8.2 показана гіпотетична ситуація, коли в результаті виконання якоїсь команди вершиною стека став регістр R6. Інші регістри розподіляються по колу: R7-ST(1), R0-ST(2), ..., R5-ST(7). Це і є їх поточні імена ST(i), $i=1, \dots, 7$ на момент виконання даної команди співпроцесора.

Якщо в цих регістрах є дані, то вони можуть слугувати операндами в командах співпроцесора. Звертатися ж напряму до регістрів R0-R7 не можна.

Таблиця 8.2

**Регістри співпроцесора
і поняття вершини стеку співпроцесора**

Назва регістрів	Біти															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWR	Регістр управління (Control Word Register)															
SWR	Регістр стану співпроцесора (Status Word Register)															
TWR	Слово ознак-тегів (Tags Word Register)															
FIP (32 біти)	Регістр вказівника останньої виконаної команди (Floating Instruction Pointer)															
FDP (32)	Регістр, де зберігається адреса операнда останньої вико- наної команди															
	(Floating Data Pointer)															
R0 – ST (2)																
(80 біт)	Розширене дійсне чи будь-яке інше доступне дане спів- процесора															
R1 – ST (3)																
(80 біт)	Розширене дійсне чи будь-яке інше доступне дане спів- процесора															
.....																
R6 – (TOP)																
(80 біт)	Розширене дійсне чи будь-яке інше доступне дане спів- процесора															
R7 – ST (1)																
(80 біт)	Розширене дійсне чи будь-яке інше доступне дане спів- процесора															

Регістр стану співпроцесора сигналізує співпроцесору про його стан і про те, як виконалась та чи інша арифметична команда. Таким чином, цей регістр виконує функції, аналогічні функціям регістра прапорців процесора FLAGS. За бітами регістра стану співпроцесора закріплеі відповідні імена, які полегшують розуміння їх призначення.

Таблиця 8.3

Регістр стану співпроцесора SWR

Регістр (слово) стану співпроцесора (SWR)																
Старший байт								Молодший байт								
1	14	1	1	1	10	9	8	7	6	5	4	3	2	1	0	
5		3	2	1												
B	C	TOP				C	C	C0	E	S	P	U	O	Z	D	I
	3					2	1		S	E	E	E	E	E	E	E
Аналогія з регістром прапорців процесора (Flags)																
	Z				PF		C									
	F						F									

Регістр управління визначає для співпроцесора варіанти обробки дійсних чисел (таблиця 8.4).

Таблиця 8.4

Регістр управління співпроцесора SWR

Регістр (слово) управління співпроцесора (CWR)															
Старший байт								Молодший байт							
1	14	1	1	1	10	9	8	7	6	5	4	3	2	1	0
5		3	2	1											
			I	RC		PC		IEM		PM	U	O	Z	D	I
			C								M	M	M	M	M

Молодші біти слова управління визначають маскування обрахованих виключень. Біти 0-5 містять індивідуальні маски для кожного із шести виключень, які розглядаються процесором при виконанні операцій з плаваючою комою. Старші біти (8-12) управляючого слова визначають параметри роботи співпроцесора.

Регістр тегів містить 8 пар бітів, які описують вміст кожного регістра даних R0-R7:

- 00 – число;
- 01 – істинний нуль;
- 10 – особливе число;
- 11 – регістр пустий.

Таблиця 8.5

Регістр (слово) тегів (TWR)															
Регістр (слово) тегів (TWR)															
Старший байт								Молодший байт							
1	14	1	1	1	10	9	8	7	6	5	4	3	2	1	0
5		3	2	1											
R7		R6		R5		R4		R3		R2		R1		R0	

Система команд

Система команд співпроцесора досить проста, якщо знати ключ і розуміти англійську. Для їх підключення необхідно зробити наступне:

1) Skorистatись однією із директив Асемблера: .8087, .287 (.286p), .387 (.386p), .487(.486p). Необхідно мати на увазі, що не всі команди процесора сумісні зверху вниз.

2) Зробити ініціалізацію співпроцесора за допомогою команди FINIT.

3) При компіляції використовувати додатковий ключ.

Базовий співпроцесор i8087 має 69 базових команд, які підкоряються наступним закономірностям:

- Всі вони починаються на букву F (Floating).
- Друга буква може бути пов'язана з форматом оброблюваних чисел:

I (Integer) – цілі числа;

B (Binary-coded decimal) – упаковане двійково-десятькове число (BCD).

Для дійсних чисел спеціальна буква не виділяється.

- Далі йде мнемонічний код операції (МКОП), наприклад:

LD(LoaD) – завантажити дане в вершину стека ST;

ST(STore) – вивантажити дане із вершини стека ST і уже відомі нам команди **ADD**, **MUL**, **DIV** і т.д.

- Закінчується команда може буквою **R** (Reserved – реверсна операція) і/або **P** (Pop – виштовхнути результат із стека і звільнити вершину стека ST).

Таблиця 8.6

Звичайні і реверсні операції

МКОП	Звичайна операція	Реверсна операція
SUB	Приймач=Приймач-Джерело	Приймач=Джерело-Приймач
DIV	Приймач=Приймач/Джерело	Приймач=Джерело/Приймач

Співпроцесори пізніших моделей, починаючи з i80387, порушили цю систему умовних позначень, зате збільшили функціональні можливості команд по обробці даних.

1.1 Команди пересилки даних

Всі команди цієї групи мають один операнд: або джерело, або приймач. Завантаження або вивантаження інформації відбувається в/з вершини стека ST(0).

Зазвичай вершину стека позначають через ST, і в командах цієї групи вона явно в операндах не є присутньою.

Таблиця 8.7

Команди передачі даних

Команда	Тип даних	Характеристика операнда	Приклад
Команди завантаження даних в стек			
FLD джерело	Дійсне	Змінна в ОЗП або ST(i), i=0,...7	FLD a FLD ST(5)
FBLD джерело	Типу BCD	Змінна в ОЗП	FBLD aBc
FILD джерело	Ціле	Змінна в ОЗП	FILD ka
Команди кооперування даних з стеку			
FST приймач	Дійсне	Змінна в ОЗП або ST(i), i=1,...7	FST ap FST ST(1)
FIST приймач	Ціле	Змінна в ОЗП	FIST kab

Команди передачі даних

Команди вивантаження даних з стеку із звільненням вершини стеку			
FSTP приймач	Дійсне	Змінна в ОЗП або ST(i), $i=1,\dots,7$	FSTP app FSTP ST(5)
FBSTP прий- мач	Типу BCD	Змінна в ОЗП	FBSTP ppp
FISTP приймач	Ціле	Змінна в ОЗП	FISTP kabP
Команда обміну вмісту джерела з ST(0)			
FXCH (джерело)	Якщо джерело не вказане, то вважається, що він не відповідає ST (1)	ST(i), $i=1,\dots,7$	FXCH FXCH ST(4)

1.2 Команди завантаження констант

Команди цієї групи поміщають в вершину стека ST(0) константи, які часто використовуються. Починаючи з співпроцесора i80987, ці константи зберігаються в більш точному форматі. Команди операндів не мають.

Таблиця 8.8

Команди завантаження констант

Команда	Призначення
FLDI	Помістити в ST (0) число 1.0
FLDZ	Помістити в ST (0) число +0.0
FLDPI	Помістити в ST (0) число π
FLDL2E	Помістити в ST (0) число, рівне $\log_2 e$
FLDL2T	Помістити в ST (0) число, рівне $\log_2 10$
FLDLN2	Помістити в ST (0) , рівне $\ln 2$
FLDLG2	Помістити в ST (0) , рівне $\lg 2$

1.3 Арифметичні команди

Арифметичні команди реалізують базову арифметику співпроцесора (чотири основні арифметичні операції: +, -, *, /) і декілька спеціальних арифметичних команд.

Формат команд базової арифметики для дійсних чисел досить складний (операнди можуть бути задати в явному вигляді, а можуть і бути відсутні, тобто неявні), тому, щоб уникнути непорозумінь, в таблиці 8.9 даний явний формат.

Спеціальні арифметичні команди в своєму форматі не містять операндів, тобто всі їх операнди задані неявно.

Таблиця 8.9

Команди базової арифметики

Команда	Тип даних	Алгоритм виконання
Команди додавання		
FADD приймач, джерело	Дійсне	Приймач=приймач+джерело
FADDP приймач, джерело	Дійсне	
FIADD приймач	Ціле	
Команди звичайного віднімання		
FSUB приймач, джерело	Дійсне	Приймач=приймач-джерело
FSUBP приймач, джерело	Дійсне	
FISUB джерело	Ціле	
Команди реверсного (зворотнього) віднімання		
FSUBR приймач, джерело	Дійсне	Приймач=джерело-приймач
FSUBPR приймач, джерело	Дійсне	
FISUBR джерело	Ціле	
Команди множення		
FMUL приймач, джерело	Дійсне	Приймач=приймач*джерело
FMULP приймач, джерело	Дійсне	
FIMUL джерело	Ціле	
Команди звичайного ділення		
FDIV приймач, джерело	Дійсне	Приймач=приймач/джерело
FDIVP приймач, джерело	Дійсне	
FIDIV джерело	Ціле	
Команди реверсного (зворотного) ділення		
FDIVR приймач, джерело	Дійсне	Приймач=джерело*приймач
FDIVRP приймач, джерело	Дійсне	
FIDIVR джерело	Ціле	

Таблиця 8.10

Спеціальні арифметичні команди

Команда	Призначення	Алгоритм виконання
FPREM	Знаходження часткової остачі від ділення шляхом послідовного віднімання 64 рази вмісту ST(1) з ST(0)	$ST(0) = \text{остача}[ST(0)/ST(1)]$ Формується прапорець C2 регістра управління CWR: C2=0 – отримана точна остача від ділення, тобто $\text{остача} < ST(1)$ C2=1 – точна остача не отримана
FPREM1 (i80387)	Знаходження часткової остачі від ділення в стандарті IEEE – округлення до найбільшого цілого	
FABS	Знаходження абсолютного значення	$ST(0) = \text{abs}(ST(0))$
FSQRT	Знаходження квадратного кореня	$ST(0) = \text{sqrt}(ST(0))$
FCHS	Зміна знака	$ST(0) = -ST(0)$
FRNDINT	Округлення до цілого	Вміст ST(0) округляється до цілого у відповідності з бітами RC регістра управління співпроцесором CWR
FXTRACT	Вивантажити експоненту і мантису числа	Число $====> ST(0)$ $ST(0)$ = мантиса числа $ST(1)$ = експонента числа
FSCALE	Команда, зворотна FXTRACT	$ST(0) <===$ мантиса числа $ST(1) <===$ експонента числа $ST(0) = ST(0) * 2^{ST(1)}$

1.4 Трансцендентні операції

Команди цієї групи в своєму форматі не містять операндів, тобто їх операнди задані неявно. Вони призначені для обчислення найбільш вживаних арифметичних функцій.

Таблиця 8.11

Команди трансцендентних операцій

Команда	Призначення	Алгоритм виконання
FSIN (i80387)	Обрахування синуса $\sin(x)$, де значення аргумента x задається в радіанах	$x \implies ST(0)$; $ST(0) = \sin(ST(0))$ $-2^{63} \leq x \leq 2^{63}$. Якщо значення x виходить за ці межі, можна скористатися командою FPREM з дільником 2П. Інакше прапорець C2=1 і значення $ST(0)$ не змінюється.
FCOS (i80387)	Обрахування синуса $\sin(x)$, де значення аргумента x задається в радіанах	$x \implies ST(0)$; $ST(0) = \cos(ST(0))$ $-2^{63} \leq x \leq 2^{63}$. Якщо значення x виходить за ці межі, можна скористатися командою FPREM з дільником 2П. Інакше прапорець C2=1 і значення $ST(0)$ не змінюється.
FSINCOS (i80387)	Обрахування синуса і косинуса	$x \implies ST(0)$; $ST(1) = \sin(ST(0))$; $ST(0) = \cos(ST(0))$ $-2^{63} \leq x \leq 2^{63}$. Якщо значення x виходить за ці межі, можна скористатися командою FPREM з дільником 2П. Інакше прапорець C2=1 і значення $ST(0)$ не змінюється.
FPTAN	Обрахування часткового тангенса $\operatorname{tg}(a)=y/x$, де значення аргумента a задається в радіанах	$a \implies ST(0)$; $ST(0) = x$; $ST(1) = y$; $0 < a < \pi/4$ (i8087, i80287). $-2^{63} \leq a \leq 2^{63}$. Якщо значення a виходить за ці межі, можна скористатися командою FPREM з дільником П. Інакше прапорець C2=1 і значення $ST(0)$ не змінюється.
FPATAN	Обрахування арктангенса $a=\operatorname{arctg}(y/x)$	$ST(0) = x$; $ST(1) = y$; $a \leq ST(0)$; знак співпадає із знаком $ST(1)$ $0 < y < x < \text{нескінченність}$. $ a < \pi$.
F2XM1	Обрахування $2^x - 1$	$X \implies ST(0)$; $ST(0) = 2^X - 1$ $-1 < X < 1$, інакше результат операції не визначений.
FYL2X	Обрахування $y \cdot \log_2(x)$	$x \implies ST(0)$; $y \implies ST(1)$; $0 < x < \text{нескінченність}$, -нескінченність $\leq y < +\text{нескінченність}$
FYL2XP1	Обрахування $y \cdot \log_2(c+1)$	$C \implies ST(0)$; $y \implies ST(1)$; $0 < c < 1 - 2^{0.5/2}$, -нескінченність $\leq y < +\text{нескінченність}$

1.5 Команди порівняння

Команди цієї групи виконують порівняння вмісту регістра ST(0) з джерелом і виставляють відповідні прапорці співпроцесора або процесора.

При порівнянні дійсних даних джерело може знаходитися або в регістрі ST(i), $i=1, \dots, 7$, або в оперативній пам'яті. Якщо джерело в форматі команди не вказане, припускається, що воно зберігається в ST(1).

При порівнянні цілочисельних даних джерело може знаходитися тільки в оперативній пам'яті.

Таблиця 8.12

Основні команди порівняння

Команда	Призначення
FCOM джерело	Порівняння дійсних даних
FCOMP джерело	Порівняння дійсних даних і звільнення вершини стека ST(0)
FCOMPP	Порівняння дійсних даних і звільнення вершини стека ST(0) і регістра ST(1)
FUCOM джерело (i80387)	Порівняння дійсних даних без врахування порядків
FUCOMP джерело (i80387)	Порівняння дійсних даних без врахування порядків і звільнення вершини стека ST(0)
FUCOMPP (i80387)	Порівняння дійсних даних без врахування порядків і звільнення вершини стека ST(0) і регістра ST(1)
FICOM джерело	Порівняння цілочисельних даних
FICOMP джерело	Порівняння цілочисельних даних і звільнення вершини стека ST(0)
FCOMI ST(0), ST(i)	Порівняння даних і встановлення EFLAGS
FCOMIP ST(0), ST(i)	Порівняння даних, встановлення EFLAGS і звільнення вершини стека ST(0)
FUCOMI ST(0), ST(i)	Порівняння даних без врахування порядків і встановлення EFLAGS
FUCOMIP ST(0), ST(i)	Порівняння даних без врахування порядків, встановлення EFLAGS і звільнення вершини стека ST(0)
FTST	Порівняння вмісту ST(0) з нулем
FXAM	Аналіз вмісту і визначення типу числа в ST(0)

1.6 Команди управління співпроцесором

В командах цієї групи неявних операндів нема. Приймач або джерело відповідають ділянці в оперативній пам'яті необхідної довжини.

Таблиця 8.13

Команди управління співпроцесором

Команда	Призначення
FLDCW джерело	Завантаження регістрів управління CWR з джерела
FSTCW приймач	Запис вмісту слова управління CWR в приймач. Дана команда повністю еквівалентна команді WAIT FNSTCW
FNTCW приймач	Запис вмісту слова управління CWR в приймач без очікування закінчення обробки виключних ситуацій
FSTSW приймач (i80287)	Запис вмісту слова стану SWR в приймач. Дана команда повністю еквівалентна команді WAIT FNSTSW
FNSTSW приймач (i80287)	Запис вмісту слова стану SWR в приймач без очікування закінчення обробки виключних ситуацій
FSAVE приймач	Збереження повного стану FPU в приймачі. Дана команда повністю еквівалентна команді WAIT FNSAVE
FNSAVE приймач	Збереження повного стану FPU без очікування обробки виключних ситуацій
FXSAVE приймач	Швидке збереження повного стану FPU в приймачі. Дана команда не сумісна з командами FSAVE/FRSTOR
FRSTOR джерело	Відновлення повного стану FPU. Дана команда є зворотною до команди FSAVE
FXSTOR джерело	Швидке відновлення повного стану FPU. Дана команда є зворотною до команди FXSAVE
FLDENV джерело	Завантаження з джерела 5 допоміжних регістрів оточення співпроцесора (CWR, SWR, TWR, FIP, FDP)
FSTENV приймач	Збереження 5-ти допоміжних регістрів. Ця команда повністю еквівалентна команді WAIT FNSTENV і є оберненою до команди FLDENV

Команди управління співпроцесором

FNSTENV приймач	Збереження 5-ти допоміжних регістрів без очікування закінчення обробки виключних ситуацій. Дана команда є оберненою до команди FLDENV
FWAIT WAIT	Очікування готовності співпроцесора
FINIT	Ініціалізація співпроцесора. Дана команда повністю еквівалентна команді WAIT FNINIT
FNINIT	Ініціалізація співпроцесора без очікування обробки виключних ситуацій
FENI (тільки в i8087)	Включення виключень. В старших версіях співпроцесора сприймається як команда FNOP.
FDISI (тільки в i8087)	Заборона виключень. В старших версіях співпроцесора сприймається як команда FNOP.
FNOP	Відсутність операції
FCLEX	Обнулення прапорців виключень в регістрі стану SWR (PE, UE, OE, ZE, DE, IE, ES, SE, B). Дана команда повністю еквівалентна команді WAIT FNCLEX
FNCLEX	Обнулення прапорців виключень без очікування
FINCSTP	Поле TOP регістра стану співпроцесора збільшується на 1
FDECSTP	Поле TOP регістра стану співпроцесора зменшується на 1
FFREE ST(i)	Звільнення регістра даних ST(i), $i=0, \dots, 7$. В регістрі тегів TWR даних регістр позначається як пустий.

Основні особливості програмування

До основних особливостей програмування співпроцесора, пов'язаних з операндами, належать:

1) При анонімних зверненнях до пам'яті бажано задавати тип операнда через директиву **PTR**. Наприклад: **FLD QWORD PTR[BX]**.

2) Консанти в операндах не допускаються. Їх потрібно задавати.

3) Наявність або відсутність операнда пов'язана не тільки з самою командою, але і з допустимими форматами команд співпроцесора. Наприклад, для арифметичних команд можливі декілька форматів.

Таблиця 8.14

Формати команд базової арифметики

Формат	Мнемокод	Допустимі операнди	Приклад
Стековий	F _{op}	ST(1), ST(0)	FSUB
Регістровий	F _{op}	ST(i), ST(0) ST(0), ST(i), i=0,...,7	FMUL ST(5), ST FADD ST,ST(2)
Регістровий з витягуванням із стека	F _{op} P	ST(i), ST(0), i=0,...,7	FMULP ST(3), ST(0)
Дійсні дані в пам'яті	F _{op}	ST(0), mem ₃₂ ST(0), mem ₆₄	FADD a
Цілі дані в пам'яті	FI _{op}	ST(0), mem ₁₆ ST(0), mem ₃₂ ST(0), mem ₆₄	FIDIV k

Приклад: Написати програму для обчислення заданого змішаного арифметичного виразу для даних у форматі float, використовуючи команди співпроцесора. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні. Відмітити нормальні та аномальні результати, зробити аналіз результатів.

Вираз:

$$x = \frac{\frac{c}{b} + a - \sqrt{24 + d}}{2ac - 1}$$

Лістинг програми:

```
#include <iostream>
#include <stdio.h>
#include <math.h>
#include <conio.h>

int main()
{
    float a, b, c, d, res_c, res_asm, z = 24, x = 2, k = 1, n = 1;
    do
    {
        printf("Enter the values:\n");
        printf("a = "); scanf_s("%d", &a);
        printf("b = "); scanf_s("%d", &b);
        printf("c = "); scanf_s("%d", &c);
        printf("d = "); scanf_s("%d", &d);
    } while (a > 2147483647.0 || a < -2147483648.0 || b > 2147483647.0
|| b < -2147483648.0 || c > 2147483647.0 || c < -2147483648.0 || d >
2147483647.0 || d < -2147483648.0);

    if (b == 0)
    {
        printf("Error! Division by 0.\n");
    }

    else
    {
        res_c = (c / b + a - sqrt(24 + d)) / (2 * a * c - 1);
        printf("Result_C = %f\n", res_c);
    }
}
```

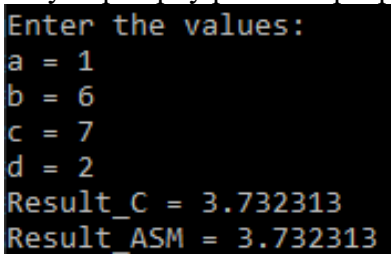


```

asm
{
    finit // ініціалізація співпроцесора
    fld c // завантаження в вершину стека
    fdiv b // ділення (c/b)
    fld d // завантаження в вершину стека
    fadd z // додавання (24+d)
    fsqrt // квадратний корінь sqrt(24+d)
    fsub // віднімання c/b-sqrt(24+d)
    fadd a // додавання c/b-sqrt(24+d)+a
    fld a // завантаження в вершину стека
    fmul x // множення (2*a)
    fmul c // множення (2*a*c)
    fsub k // віднімання (2*a*c-1)
    fdiv // ділення (c/b+a-sqrt(24+d))/(2*a*c-1)
    fstp res_asm // res_asm=(c/b+a-sqrt(24+d))/(2*a*c-1)
}
printf("Result_ASM = %f\n", res_asm);
_getch();
return 0;
}

```

Проведемо тестову перевірку роботи програми(рис.8.1).



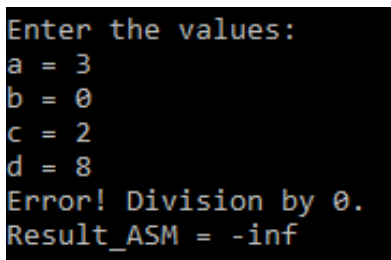
```

Enter the values:
a = 1
b = 6
c = 7
d = 2
Result_C = 3.732313
Result_ASM = 3.732313

```

Рисунок 7.1 – Перевірка роботи програми

В даному виразі може виникнути помилка ділення на 0(рис.7.2):



```

Enter the values:
a = 3
b = 0
c = 2
d = 8
Error! Division by 0.
Result_ASM = -inf

```

Рисунок 7.2 – Помилка при діленні на 0

Значення регістрів співпроцесора при пороковому виконанні

Команда	Опис	Стан регістрів	Коментар
finit	Ініціалізація співпроцесора	cwr=037Fh; swr=0; twr=FFFFh; dpr=0; ipr=0;	
fld c	Завантаження в вершину стека	src	c
fdiv b	Ділення	dst=dst/src	c/b
fld d	Завантаження в вершину стека	src	d
fadd z	Додавання	dst=dst+src	d+24
fsqrt	Квадратний корінь	$St(0) = \sqrt{St(0)}$	sqrt(24+d)
fsub	Віднімання	dst=dst-src	c/b-sqrt(24+d)
fadd a	Додавання	dst=dst+src	c/b-sqrt(24+d)+a
fld a	Завантаження в вершину стека	src	a
fmul x	Множення	dst=dst*src	2*a
fmul c	Множення	dst=dst*src	2*a*c
fsub k	Віднімання	dst=dst-src	2*a*c-1
fdiv	Ділення	dst=dst/src	(c/b-sqrt(24+d)+a)/ (2*a*c-1)
fstp res_asm	Збереження у вершині стеку	dst= St(0)	

Правила формування завдань

Якщо унікальний номер групи **G = 1**, то завдання згідно таблиці 8.15 будуть мати вигляд:

G – непарне – знак «+», **G** – парне – знак «-»

G – непарне – **f1** – ln, **G** – парне – **f1** lg

G – непарне – **f2** – tg, **G** – парне – **f2** arctg

G – непарне – **f3** – sin, **G** – парне – **f3** cos

№ Ва- ріанту	Вираз	Вирази якщо G=1 матимуть вигляд
1	$\frac{2 * c - d \pm \sqrt{G * a}}{f1 \left(\frac{a}{4} - 1 \right)}$	$\frac{2 * c - d + \sqrt{1 * a}}{\ln \left(\frac{a}{4} - 1 \right)}$
2	$\frac{\frac{b}{2} \pm \frac{G}{c}}{f2 (b - a) * c + 1}$	$\frac{\frac{b}{2} + \frac{1}{c}}{tg (b - a) * c + 1}$
3	$\frac{-2 * c \pm d * G}{f3 \left(\frac{q}{4} - 1 \right)}$	$\frac{-2 * c + d * 1}{\sin \left(\frac{q}{4} - 1 \right)}$

Всі інші варіанти формуються аналогічно!!!

Вимоги до оформлення звіту

Звіт оформлюється у текстовому документі і має містити:

1. Правильно оформлену рамку.
2. Тему, мету, номер варіанту.
3. Виконані **всі** завдання з ходом обрахунків.
4. Висновки.

Звіт оформлений неналежним чином і без ходу обрахунків не оцінюється і має бути перероблений!

Завдання на лабораторну роботу

1. Написати програму для обчислення заданих змішаних арифметичних виразів(табл.8.15) для даних у форматі float, використовуючи команди співпроцесора. Виконати покрокове виконання асемблерного коду та навести стан регістрів при їх виконанні. Відмітити нормальні та аномальні результати, зробити аналіз результатів.

Таблиця 8.15

Дані для розрахунку

N – номер варіанта зазначений у рейтинговій таблиці,

G – унікальний номер групи зазначений на сторінці курсу.

№ Варіанту	ЗНАК $\pm$ та $f1, f2, f3$	Вираз 1	Вираз 2
1	G – парне – знак «-» G – парне – $f1$ – ln, G – парне – $f1$ lg G – непарне – $f2$ – tg, G – парне – $f2$ arctg G – непарне – $f3$ – sin, G – парне – $f3$ cos	$\frac{2 * c - d \pm \sqrt{G * a}}{f1\left(\frac{a}{4} - 1\right)}$	$\frac{G * b \pm 8 * c}{f2(b + a) + 1}$
2		$\frac{c + 4 * d \pm \sqrt{G * c}}{f1\left(1 - \frac{a}{2}\right)}$	$\frac{G + f2\left(\frac{b}{4} - 1\right)}{\frac{c}{3} \pm a * d}$
3		$\frac{-2 * c \pm d * G}{f3\left(\frac{q}{4} - 1\right)}$	$\frac{f1\left(G + 2 * \frac{a}{d}\right)}{c + a \pm b - 1}$
4		$\frac{f1(2 * c - a) + d - G}{\frac{a}{4} \pm c}$	$\frac{\frac{b}{2} \pm \frac{G}{c}}{f2(b - a) * c + 1}$
5		$\frac{f2\left(a + \frac{c}{4}\right) - G * d}{a * b \pm 1}$	$\frac{4 * f1\left(\frac{a}{b}\right) + G}{c + b \pm d + a}$

6	G – непарне – знак «+», G – парне – знак «-» G – непарне – f1 – ln, G – парне – f1 lg G – непарне – f2 – tg, G – парне – f2 arctg G – непарне – f3 – sin, G – парне – f3 cos	$\frac{-2*c - f3\left(\frac{a}{d}\right) + G}{\frac{a}{4} \pm b}$	$\frac{\frac{G}{c} + f2(3*a)}{\frac{c}{a} \pm b - 1}$
7		$\frac{2*c \pm f2\left(\frac{d}{4}\right)}{a*a - G}$	$\frac{c + G \pm b + 4}{a - f1\left(a + \frac{c}{b} - 1\right)}$
8		$\frac{f2(c + G) - d*2}{\sqrt{2*a \pm 1}}$	$\frac{f2\left(\frac{c}{4} + G\right)*d}{\frac{a}{d} \pm c - 1}$
9		$\frac{2*c \pm \frac{G}{d}}{f1\left(1 - \frac{a}{4}\right)}$	$\frac{\sqrt{2*b} \pm G + \frac{b}{c}}{f3\left(a - \frac{c}{4} + 1\right)}$
10		$\frac{4*c \pm d - 1}{G - f3\frac{a}{2}}$	$\frac{G* f1(b - 1) - c}{a*2 \pm \frac{b}{c}}$
11		$\frac{2*c \pm d*\sqrt{\frac{G}{d}}}{f3(c + a - 1)}$	$\frac{\frac{f2(b - c)}{a} + \sqrt{G}}{a \pm b - 1}$
12		$\frac{\sqrt{\frac{G}{c} \pm d + 2}}{f3(d + a - 1)}$	$\frac{c*b \pm G + a}{b - f1(2*c - 1) + a}$

13	G – непарне – знак «+», G – парне – знак «-» G – непарне – f1 – ln, G – парне – f1 lg G – непарне – f2 – tg, G – парне – f2 arctg G – непарне – f3 – sin, G – парне – f3 cos	$\frac{f2\left(c \pm \frac{d}{2}\right)}{2 * a - G}$	$\frac{4 * f1(b) \pm 1}{G}$
14		$\frac{4 * f1(c) - \frac{d}{2} + G}{a^2 \pm 1}$	$\frac{\sqrt{\frac{G * d}{a}} \pm 1}{f3\left(a - \frac{c}{b}\right) + d}$
15		$\frac{c * f2(b + G)}{\frac{a}{2} \pm 4 * d - 1}$	$\frac{f1(a * b \pm 2) * c}{G - \frac{b}{c} + 1}$
16		$\frac{\frac{c}{d} \pm f1\left(3 * \frac{a}{2}\right)}{G - a + 1}$	$\frac{f2(a - c) * b + G}{4 * \frac{b}{a} \pm 1}$
17		$\frac{2 * c + f1(d) * \sqrt{G}}{d \pm a - 1}$	$\frac{G * a \pm \frac{c}{2}}{c - f2(b + 1)}$
18		$\frac{c + f1\left(\frac{c}{a} - G\right)}{a^2 \pm d + 1}$	$\frac{2 * c + f2(a - G)}{\frac{c}{a} \pm d + 1}$
19		$\frac{G * c \pm \frac{d}{2} + 1}{a^2 - f1(b - 5)}$	$\frac{f1(4 * b - c) * a}{G \pm \frac{c}{d} - 1}$
20		$\frac{f2(2 * c) \pm 2}{d}$	$\frac{1 + a \pm f3\left(\frac{b}{2}\right)}{\sqrt{G + d - c} + \frac{a}{b}}$
21		$\frac{f2\left(\frac{12}{c}\right) + G}{a^2 \pm 1}$	$\frac{f3\left(a * \frac{b}{4} - 1\right)}{\sqrt{G - d} \pm b * a + c}$

22	G – непарне – знак «+», G – парне – знак «-» G – непарне – $f1 - \ln$, G – парне – $f1 \lg$ G – непарне – $f2 - \operatorname{tg}$, G – парне – $f2 \operatorname{arctg}$ G – непарне – $f3 - \sin$, G – парне – $f3 \cos$	$\frac{2 * \frac{c}{a} \pm d^2}{G + f2(a-1)}$	$\frac{c - \frac{f1(G+b)}{4}}{a \pm \frac{c}{b} - 1}$
23		$\frac{\sqrt{\frac{G}{a}} + d - f2(4 * a)}{1 \pm a * c}$	$\frac{f1(G-a) * \frac{c}{4}}{1 + \frac{c}{a} \pm b}$
24		$\frac{\sqrt{G * a} \pm b - \frac{a}{4}}{f3(c * d) - 1}$	$\frac{2 * b - f1(a+b) * c}{\frac{G}{4} \pm 1}$
25		$\frac{-\frac{G}{a} + c - f2(b)}{1 \pm c * \frac{d}{2}}$	$\frac{f1\left(\frac{b}{a} + G\right) + d}{c \pm b + 1}$
26		$\frac{f1(4 * a - 1) \pm \frac{b}{2}}{d * c - G}$	$\frac{a \pm b * 4 - 1}{\frac{c}{G} + f2(a * b)}$
27		$\frac{8 * f1(b+1) - G}{\frac{a}{2} \pm d * c}$	$\frac{c * 4 \pm 28 * \frac{d}{e}}{G - f2(a * b - c - 1)}$
28		$\frac{4 * a \pm f1(b-1)}{\frac{c}{d} + G}$	$\frac{G \pm \frac{d}{4} - 1}{\frac{c}{f2(b+a)} - d}$

29		$\frac{\frac{f2(4*b)}{c} - 1}{G \pm a - b}$	$\frac{\frac{c}{b} \pm \sqrt{G+d} + a}{f3(2*a*c - 1)}$
30		$\frac{f3\left(b - \frac{a}{4}\right) \pm c*b}{a*d - G}$	$\frac{f1\left(a + \frac{c}{b}\right) - \sqrt{28*d}}{G*b \pm a + 1}$
31	G – непарне – знак «+», G – парне – знак «-» G – непарне – f1 – ln, G – парне – f1 lg G – непарне – f2 – tg, G – парне – f2 arctg G – непарне – f3 – sin, G – парне – f3 cos	$\frac{\frac{d \pm \sqrt{\frac{G}{2}}}{a}}{f1(b+c-d)}$	$\frac{\sqrt{\frac{f3(4*a)}{a}} + d}{G \pm a + c}$
32		$\frac{b*c \pm \frac{a^2}{2} + 1}{d + f1(G-5)}$	$\frac{f1\left(G*\frac{a}{2}\right)}{\frac{c}{d} \pm \sqrt{a+1}}$
33		$\frac{f3(2*d - \sqrt{G})}{e \pm b*a + c}$	$\frac{5*c \pm f1\left(\frac{d}{4}\right) + G}{a^2 - 1}$
34		$\frac{f1(a+d) + G}{\sqrt{\frac{b}{a}} \pm c*2}$	$\frac{G*c \pm d + \sqrt{\frac{b}{d}}}{f2(2*a - 1)}$
35		$\frac{c*b \pm \sqrt{2+a}}{f3(d+G) + 1}$	$\frac{G + \frac{c}{b} \pm \sqrt{28*d}}{f1(b*a + 1)}$

Контрольні питання

1. Що таке математичний співпроцесор?
2. Типи звичайних даних співпроцесора.
3. Спеціальних формати даних співпроцесора.
4. Назвіть регістри співпроцесора.
5. Звичайні і реверсні операції.
6. Команди пересилки даних.
7. Команди завантаження констант.
8. Арифметичні команди
9. Трансцендентні операції
10. Команди порівняння та управління співпроцесором