

Зміст лекції

1. Математичний співпроцесор (FPU)
2. Типи даних FPU
3. Регістри FPU
4. Система команд FPU

Математичний співпроцесор (також відомий як FPU - Floating Point Unit) - це частина процесора, яка спеціалізується на виконанні операцій з рухомою комою (операції, які включають десяткові або дробові числа). У контексті архітектури x86 та IA-32, математичний співпроцесор входить в склад процесора і виконує операції, які були спеціально розроблені для обробки чисел з плаваючою комою.

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

Типи звичайних даних співпроцесора

Тип даних	Довжина (біт)	Кількість 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 двійководесяткове	80	18	-9999999999999999 ... 9999999999999999
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	3.3621031431120935063e4932... 1.189731495357231765e+4932

Крім звичайних чисел, специфікація стандарту IEEE 754 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.
- Додатна 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 даних.

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.

Назва регістрів	Біти															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWR (16 біт)	Регістр управління (Control Word Register)															
SWR (16 біт)	Регістр стану співпроцесора (Status Word Register)															
TWR (16 біт)	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 виконаної команди (Floating Data Pointer)															
R0 – ST (5) (80 біт)	Розширене дійсне число, упаковане двійково-десятькове число або будь-яке інше доступне значення співпроцесора															
R1 – ST (6) (80 біт)	Розширене дійсне число, упаковане двійково-десятькове число або будь-яке інше доступне значення співпроцесора															
R2 – ST (7) (80 біт)	Розширене дійсне число, упаковане двійково-десятькове число або будь-яке інше доступне значення співпроцесора															
R3 – (TOP) вершина стека ST або ST (0) (80 біт)	Розширене дійсне число, упаковане двійково-десятькове число або будь-яке інше доступне значення співпроцесора															

R4 – ST (1) (80 біт)	Розширене дійсне число,
	упаковане двійково-десятькове число
	або будь-яке інше доступне значення співпроцесора
R5 – ST (2) (80 біт)	Розширене дійсне число,
	упаковане двійково-десятькове число
	або будь-яке інше доступне значення співпроцесора
R6 – ST (3) (80 біт)	Розширене дійсне число,
	упаковане двійково-десятькове число
	або будь-яке інше доступне значення співпроцесора
R7 – ST (4) (80 біт)	Розширене дійсне число,
	упаковане двійково-десятькове число
	або будь-яке інше доступне значення співпроцесора

Принцип роботи з круговим стеком співпроцесора полягає в наступному:

Будь-яка команда завантаження даних співпроцесора автоматично переміщує вершину стеку співпроцесора:

$TOP = TOP + 1$.

В таблиці показана гіпотетична ситуація, коли в результаті виконання якоїсь команди вершиною стека став регістр R3. Інші регістри розподіляються по колу: R4-ST(1), R5-ST(2), ..., R2-ST(7). Це ї є їх поточні імена ST(i), $i=1, \dots, 7$ на момент виконання даної команди співпроцесора. Звертатися ж напряду до регістрів R0-R7 не можна.

Якщо в цих регістрах є дані, то вони можуть слугувати операндами в командах співпроцесора.

Регістр стану співпроцесора

Регістр стану 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 закріплені відповідні імена, які полегшують розуміння їх призначення.

Регістр (слово) стану співпроцесора (SWR)																
Старший байт								Молодший байт								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
B	C3	TOP				C2	C1	C0	ES	SE	PE	UE	OE	ZE	DE	IE
Аналогія з регістром прапорців процесора (Flags)																
	ZF				PF		CF									

Розглянемо наявні прапорці:

Прапорці умов (станів). Conditions

C0 – Аналог прапорця переносу CF процесора. Містить ознаки переносу інформації зі старшого біта.

C1 – Прапорець умови співпроцесора. Не є аналогічним прапорцю умови процесора, а пов'язаний з прапорцем SE.

C2 – Аналог прапорці парності PF процесора.

C3 – Аналог прапорця нуля ZF процесора. Встановлюється в 1, якщо результат виконання команди дорівнює 0.

Прапорці виключень. Exceptions

ES – Загальний прапорець помилки. Встановлюється в 1, якщо виникла хоча б одна немаскована помилка.

SE – Помилка стеку. Якщо C1=1, то виникло переповнення стеку (команда намагалась записати інформацію в не порожню комірку стеку). Якщо C1=0, то виникло антипереповнення стеку (команда намагалась зчитати інформацію з пустої комірки стеку).

PE (precision) – Прапорець втрати точності. Встановлюється коли результат не може бути представлений точно, наприклад $1/3$.

UE (Underflow) – Прапорець антипереповнення – результат занадто малий.

OE (Overflow) – Прапорець переповнення – результат занадто великий.

ZE (Zero divide) – Прапорець ділення на нуль.

DE (Denormal) – Прапорець денормалізованого операнда – виконано операцію над денормалізованим числом.

IE (Invalid operation) – Прапорець недопустимої операції.

Інші прапорці станів

V (Busy) – Прапорець зайнятості FPU. Використовується для забезпечення сумісності з i8087 та i80287. В сучасних співпроцесорах його значення співпадає із значенням прапорця ES.

TOP – Двійкове число від 000 до 111, котре вказує, який з регістрів R0-R7 в даний момент є вершиною стеку ST.

Слід звернути увагу, що у співпроцесора немає аналога прапорця SF (sign flag).

Регістр управління

Цей регістр визначає для співпроцесора варіанти обробки дійсних чисел. Розглянемо формат та кодування полів в регістрі управління.

Регістр (слово) управління співпроцесора (CWR)															
Старший байт								Молодший байт							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			IC	RC		PC		IEM		PM	UM	OM	ZM	DM	IM

Молодші біти слова управління визначають маскування (заборону) обчислювальних виключень. Біти 0-5 містять індивідуальні маски для кожного із шести виключень, які визначаються співпроцесором при виконанні операцій з плаваючою комою.

Старші біти (8-12) управляючого слова визначають параметри роботи співпроцесора.

Регістр тегів

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

00 – число;

01 – істинний нуль;

10 – особливе число (нескінченність, денормалізоване число і т.д);

11 – регістр пустий.

Регістр (слово) тегів (TWR)															
Старший байт								Молодший байт							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R7		R6		R5		R4		R3		R2		R1		R0	

Інформація регістра тегів може використовуватись програмами, які оброблюють виключення, що виникають при обчисленнях у співпроцесорі

Система команд

- Потрібно зробити ініціалізацію співпроцесора за допомогою команди FINIT.
- При компіляції використовувати додатковий ключ /r або /e (Real або Emulated floating-point instruction)
- Базовий співпроцесор i8087 має 69 базових команд, що мають наступні закономірності:
 - Перша буква команди – F (Floating)
 - Друга буква може бути пов'язана за форматом оброблюваних чисел:
 - I (Integer) – ціле число.
 - B (Binary-coded decimal) – упаковане двійково-десятькове число BCD.
 - Далі – мнемонічний код операції (МКОП):
 - LD (LoaD) – завантажити дані на вершину стеку ST
 - ST (STore) – вивантажити дані з вершини стека ST
 - Інші команди ADD, MUL, DIV
 - Остання буква може бути:
 - R (Reversed – реверсивна операція) (коли результат отримується навпаки, операнд 2 ділиться на операнд 1, або від операнда 2 віднімається операнд 1, хоча результат все одно зберігається в операнді 1 $Op1=Op2/Op1$, або $Op1=Op2-Op1$)
 - P (Pop – виштовхнути результат зі стека та звільнити вершину стека ST)

Команди пересилання даних

Всі ці команди мають лише один операнд – або джерело, або приймач. Завантаження та вивантаження даних виконується з/у вершини стека ST(0). Зазвичай вершина стека позначається просто ST, та в командах явно не вказується.

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 вивантаження даних з стеку із звільненням вершини стеку			
FSTP приймач	Дійсне	Змінна в ОЗП або ST(i), i=1,...7	FSTP app FSTP ST(5)
FBSTP приймач	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,...7	FXCH FXCH ST(4)

Команди завантаження констант.

Команди завантаження констант поміщають в вершину стека ST(0) константи, які часто використовуються. В сучасних процесорах ці константи зберігаються в більш точному форматі.

Відповідно, ці команди не мають явних операндів.

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 $\lg 2$

Арифметичні команди

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

Формат команд базової арифметики для дійсних чисел досить складний (операнди можуть бути задати в явному вигляді, та у не явному, тобто бути прихованими/відсутніми).

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 (зворотного) 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	

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

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 знаку	$ST(0) = -ST(0)$
FRNDINT	Округлення до цілого	Вміст $ST(0)$ округляється до цілого у відповідності з бітми RC регістра управління співпроцесором CWR
FXTRACT	Вивантажити експоненту і мантису числа	Число $\Rightarrow ST(0)$ $ST(0)$ = мантиса числа $ST(1)$ = експонента числа
FSCALE	Команда, зворотна FXTRACT	$ST(0) \Leftarrow$ мантиса числа $ST(1) \Leftarrow$ експонента числа $ST(0) = ST(0) * 2^{ST(1)}$

Трансцендентні операції.

Трансцендентні операції також не містять операндів, і призначені для обчислення найбільш вживаних арифметичних функцій.

FSINCOS (i80387)	Обрахування синуса і косинуса	$x \Rightarrow ST(0)$; $ST(1) = \sin(ST(0)); ST(0) = \cos(ST(0))$ $-2^{63} \leq x \leq 2^{63}$. 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 $tg(a)=y/x$, де значення аргумента a задається в радіанах	$a \Rightarrow ST(0)$; $ST(0) = x; ST(1) = y$; $0 < a < \pi/4$ (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 $a = \arctg(y/x)$	$ST(0) = x; ST(1) = y$; $a \Leftarrow ST(0)$; знак співпадає із знаком $ST(1)$ $0 < y < x < \infty$. $ a < \pi$.
F2XM1	Обрахування $2^x - 1$	$x \Rightarrow ST(0)$; $ST(0) = 2^x - 1$ $-1 < x < 1$, інакше результат операції не визначений.
FYL2X	Обрахування $y * \log_2(x)$	$x \Rightarrow ST(0); y \Rightarrow ST(1)$; $0 < x < \infty$, $-\infty < y < +\infty$
FYL2XP 1	Обрахування $y * \log_2(c+1)$	$c \Rightarrow ST(0); y \Rightarrow ST(1)$; $0 < c < 1 - 2^{-0.5}/2$, $-\infty < y < +\infty$

Команди порівняння

Команди порівняння 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).

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

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)

Результати порівняння в прапорці C0, C2, C3 у процесорах Intel (Pentium Pro) одразу формують відповідні прапорці CF, PF та ZF регістру EFLAGS). Але слід враховувати, що не всі співпроцесори сторонніх від Intel виробників це підтримують.

Умова	Прапорець C3 / ZF	Прапорець C2 / PF	Прапорець C0 / CF
ST (0) > джерело	0	0	0
ST (0) < джерело	0	0	1
ST (0) = джерело	1	0	0
Операнди не можуть бути порівняні	1	1	1

Окремо слід розглянути Таблицю результатів виконання команди FXAM

Тип числа в ST (0)	Прапорець C3	Прапорець C2	Прапорець C0
Не підтримує число	0	0	0
НЕ-число	0	0	1
Нормальне скінчене число	0	1	0
Нескінченість	0	1	1
Нуль	1	0	0
Регістр порожній	1	0	1
Денормалізоване число	1	1	0

При цьому прапорець C1 встановлюється рівним знаку числа в ST (0)

Команди управління співпроцесором

Команди управління співпроцесором не містять явних операндів. Приймач або джерело відповідають ділянці в оперативній пам'яті необхідної довжини.

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

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.