

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний авіаційний університет

АРХІТЕКТУРА КОМП'ЮТЕРІВ

Методичні рекомендації
до виконання домашнього завдання
для студентів спеціальності
123 «Комп'ютерна інженерія»

Київ 2019

УДК 004.27(076.5)
ББК 3973.20 p

Укладачі: Єфимець В.М., Жуков І.А., Ковальов М.О.

Рецензент Зіатдінов Ю.К.

Затверджено методично-редакційною радою Національного авіаційного університету (протокол №4/18 від 13.12.2018 р.)

А 875 **Архітектура комп'ютерів:** методичні рекомендації до виконання домашнього завдання / уклад.: В.М. Єфимець, І.А. Жуков, М.О. Ковальов – К. НАУ, 2019. - 32 с.

Містить рекомендації та навчальний матеріал розробки окремих питань архітектури комп'ютерів. Наведено варіанти завдань, приклади їх реалізації та вимоги до оформлення пояснювальної записки.

Для студентів спеціальності 123 «Комп'ютерна інженерія».

ВСТУП

Проектування комп'ютерів складається з послідовності етапів, перший з яких – архітектурний. На цьому етапі визначають систему команд, формати даних, структурну організацію комп'ютера.

Важливою частиною вивчення дисципліни «Архітектура комп'ютерів» є виконання домашнього завдання (ДЗ), що дає змогу теоретичний матеріал перевести у практичну площину.

У методичних рекомендаціях розкриваються теми ДЗ, що вивчаються у відповідних розділах навчальної дисципліни:

- формати (стандарти) даних з плаваючою комою (ПК);
- арифметика чисел з плаваючою комою (додавання, віднімання, множення, ділення);
- комп'ютер з ефективним набором команд.

Така структура ДЗ дозволяє охопити значний обсяг навчального матеріалу, його різноманітність у часі. Крім того, збільшується індивідуалізація завдань.

Домашнє завдання 1 Формати чисел з ПК

Мета: дослідити формати даних (класичний, IBM 360/370, DEC, IEEE-754) з плаваючою комою.

Визначити:

- особливості з кожного з форматів (стандартів) чисел з ПК;
- діапазони, точність чисел з ПК;
- двійкові та десяткові еквіваленти чисел з ПК.

Теоретичні відомості

Існує мало задач, у яких використовуються дані з фіксованою комою. Тому для комп'ютерних обчислень розроблена структура (формат) з плаваючою комою.

$$X = \pm M \cdot q^{\pm Px}$$

Складовими чисел з ПК є: M – мантиса; P_x – порядок; q – основа системи числення.

Мантиса M може бути цілим, дробовим, змішаним числом. Для її однозначної інтерпретації в ЕОМ мантиса представляється правильним нормалізованим дробом. Порядок P_x є цілим знаковим числом. При $q = 2$ нормалізована мантиса має такий діапазон значень:

$$0,5 \leq M \leq 1 - 2^{-m}$$

Відповідно до поколінь ЕОМ змінювались структури чисел з ПК:

- класична структура використовувалась в ЕОМ 1-го та 2-ого поколінь;
- структура IBM 360/370 реалізована в ЕОМ 3-го покоління фірми IBM;
- структура DEC запропонована в міні-ЕОМ PDP-11 фірми DEC;
- структура IEEE-754 є стандартизованою і використовується в сучасних ЕОМ.

1.1 Класична структура чисел з ПК

Особливості:

- система числення – двійкова;
- число (кількість) полів – 4 (рис.1);

- порядок числа – дійсний;
- розрядність мантиси M , порядку Π відповідно m та p .



Рис.1.1 Класична структура чисел з ПК

Визначимо діапазон нормалізованих чисел у форматі « m, p ». Мінімальне число X має мінімальну мантису M_{\min} та максимальний від'ємний порядок Π_{\min} :

$$X_{\min}: 0 \ 10\dots0 \ 1 \ 1\dots1,$$

$$M_{\min} = 2^{-1}, \ \Pi_{\min} = -(2^p - 1),$$

$$X_{\min} = 2^{-1} \cdot 2^{-(2^p-1)} = 2^{-2^p}.$$

Максимальному числу X відповідають максимальні мантиси та порядок:

$$X_{\max}: 0 \ 1\dots1 \ 0 \ 1\dots1,$$

$$M_{\max}: (1-2^{-m})=1; \quad \Pi_{\max}=2^p - 1,$$

$$X_{\max}: 1 \cdot 2^{(2^p-1)} = 2^{(2^p-1)}$$

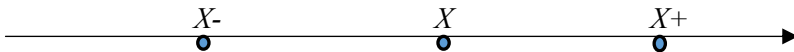
Приклад:

$$p=6, \ X_{\max} = 2^{(2^6-1)} = 2^{63} \cong 10^{+19};$$

$$X_{\min} = 2^{-2^6} = 2^{-64} \cong 10^{-20};$$

Діапазон чисел з ПК залежить від розрядності порядку « p ».

Визначимо залежність точності числа X . На числовій осі розмістимо найближчі значення чисел до числа X :



$$X = M \cdot 2^\Pi; \ X_- = (M - 2^{-m}) \cdot 2^\Pi; \ X_+ = (M + 2^{-m}) \cdot 2^\Pi.$$

Різниця між сусідніми значеннями відповідає абсолютній похибці і буде абсолютною:

$$\Delta_- = X_- - X = (M - 2^{-m}) \cdot 2^\Pi - M \cdot 2^\Pi = -2^{-m+\pi},$$

$$\Delta_+ = X_+ - X = (M + 2^{-m}) \cdot 2^\Pi - M \cdot 2^\Pi = 2^{-m+\pi}.$$

Тоді відносна похибка буде:

$$\delta = \frac{\Delta}{X} = \frac{2^{-m+\Pi}}{M \cdot 2^{\Pi}} = \frac{2^{-m}}{M}$$

Максимальна відносна похибка при $M_{min} = 2^{-1}$

$$\delta_{max} = \frac{2^{-m}}{M_{min}} = \frac{2^{-m}}{2^{-1}} = 2^{-m+1}.$$

Приклад:

$$m = 24, \delta_{max} = 2^{-24+1} = 2^{-23} \cong 10^{-\frac{23}{3,3}} \cong 10^{-7},$$

$$m = 64, \delta_{max} = 2^{-64+1} = 2^{-63} \cong 10^{-\frac{63}{3,3}} \cong 10^{-20}.$$

Таким чином, точність чисел з ПК визначається розрядністю мантиси m .

Відносна похибка $\delta = 2^{-m+1} \cong 10^{-\frac{m+1}{3,3}} \cong 10^{-7}$ визначає точність числа X до 7-го десяткового розряду.

1.2. Структура чисел з ПК IBM-360/370

Формат чисел з ПК IBM-360/370 (ЕС ЕОМ) де-факто використовується в ЕОМ 3-го покоління.

Особливості:

- система числення – шістнадцяткова;
- два формати даних: одинарний – 32 біт; подвоєний – 64 біт;
- змінений порядок.

Формати даних:

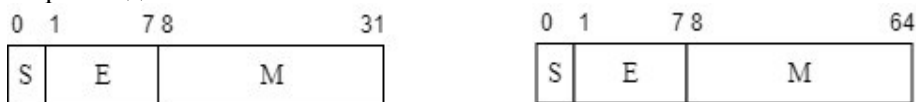


Рис. 1.2. Формати даних IBM-360/370

S – знак числа; E – зміщений порядок; M – мантиса.

Порядок E – ціле беззнакове число. Це дозволяє порівнювати числа з ПК як цілі беззнакові числа. $E = N + \Pi$, де $N = 2^6 = 64$ – зміщення; Π – дійсний порядок. Граничні значення порядку Π будуть такими:

$$E_{min} = 0 = 64 + \Pi_{min}; \Pi_{min} = -64,$$

$$E_{max} = 128 = 64 + P_{max}; P_{max} = 63, \\ -64 < P < 63.$$

Використання шіснадцяткової системи збільшує діапазон чисел і пришвидшує виконання арифметичних операцій. Нормалізоване число - це ненульове значення чотирьох старших розрядів мантиси, що відповідає значенню шіснадцяткового числа. Операція нормалізації виконується швидше, але точність відображення мантиси зменшується.

Загальна форма числа в цьому форматі така:

$$X = (-1)^s \cdot M \cdot 16^{E-N}$$

Визначимо діапазон чисел одинарної довжини:

$$X_{min}: \underbrace{0}_{S} \underbrace{0000000}_{E_{min}} \underbrace{00010 \dots 0}_{M_{min}},$$

$$X_{min} = 16^{-1} \cdot 16^{-64} = 16^{-65} = 2^{-260} \cong 10^{-78},$$

$$X_{max}: \underbrace{0}_{S} \underbrace{1111111}_{E_{max}} \underbrace{11 \dots 1}_{M_{max}},$$

$$X_{max} = 16^{+63} = 2^{-252} \cong 10^{+76},$$

$$10^{-78} < X < 10^{+76}.$$

Максимальна відносна похибка для одинарного та подвійного форматів буде такою. Позначимо через m число 16-розрядних цифр мантиси:

$$X = M \cdot 16^{\Pi},$$

$$X = (M \cdot 16^{-m}) \cdot 16^{\Pi},$$

$$X_+ = (M - 16^{-m}) \cdot 16^{\Pi},$$

$$\Delta_- = -16^{-m+\Pi}, \Delta_+ = 16^{-m+\Pi};$$

$$\delta = \frac{|\Delta|}{X} = \frac{16^{-m+\Pi}}{M \cdot 16^{-\Pi}} = \frac{16^{-m}}{M};$$

$$\delta_{max} = \frac{16^{-m}}{M_{min}} = 16^{-m+1} = 2^{4(-m+1)}$$

Одинарний формат: $m=6$

$$\delta_{max} = 2^{4(-6+1)} = 2^{-20} \approx 10^{-6},$$

Подвійний формат: $m = 16$

$$\delta_{max} = 2^{4(-16+1)} = 2^{-52} \approx 10^{-17}.$$

1.3. Структура чисел з ПК DEC

Цей формат запропонований фірмою DEC і використаний в міні-ЕОМ PDP-11.

Особливості формату:

- система числення – двійкова;
 - зміщений порядок;
 - два формати даних: короткий (32 біт) та довгий (64 біт)
- (рис. 1.3);

- Наявність схованого біта мантиси.



Рис.1.3. Формати даних DEC.

Зміщений порядок є 8-розрядним

$$E=N+П, N=2^{p-1} = 2^{8-1} = 128.$$

$$E_{min}=128+П_{min}=0; \quad П_{min}=-128; \quad E_{max}=128+П_{max}=255; \quad П_{max}=127; \\ -128 \leq П \leq 127.$$

Мантиса M – це правильний нормалізований дріб, в старшому розряді якого завжди знаходиться «1» ($M=1 \times \dots \times$). В форматі DEC це враховано – вона наявна умовно. Позиція цього розряду мантиси M використовується для розміщення молодшого біта, зміщеного порядку E . Таким чином порядок E подається у 8-розрядному полі, що збільшує вдвічі діапазон чисел з ПК. У такому вигляді число зберігається в пам'яті. Під час завантаження в процесор схований біт стає явним і в операції бере участь 24- або 56-розрядна мантиса. При записі числа з ПК у пам'ять старший біт мантиси стає віртуальним і мантиса відображається 23-а або 55-а бітами.

Число з ПК у форматі DEC:

$$X = (-1)^S \cdot 1(M) \cdot 2^{E-N}.$$

Діапазон і точність чисел з ПК визначимо для короткого формату:

$$X_{min}: \underbrace{0}_S \underbrace{00000000}_{E_{min}} \underbrace{00000000 \ 00000000 \ 00000000 \ 00000001}_{M_{min}},$$

$$X_{min} = +(2^{-1} + 2^{-24}) * 2^{-128} \approx 2^{-129} \approx 10^{-39},$$

$$X_{max}: \underbrace{0}_S \underbrace{11111111}_{E_{max}} \underbrace{11111111 \ 11111111 \ 11111111 \ 11111111}_{M_{max}},$$

$$X_{max} = (1 - 2^{-24}) * 2^{+127} \approx 2^{-127} \approx 10^{-38},$$

$$10^{-39} \leq X \leq 10^{+38}.$$

У довгому форматі діапазон чисел такий самий. Короткий та довгий формати відрізняються тільки точністю чисел з ПК, наприклад:

$$m=24, \quad \delta_{max} = 2^{-m+1} = 2^{-23} \approx 10^{-7};$$

$$m=56, \quad \delta_{max} = 2^{-m+1} = 2^{-55} \approx 10^{-17}.$$

1.4 Стандарт IEEE-754

Розглянуті формати чисел унеможливають перенесення програм, які розроблені для одного компілятора на інші. Стандарт IEEE-754 уніфікує формат чисел з ПК. Стандарт розроблений у 1985р. В. Каганом – професором математики Каліфорнійського університету в Берклі.

Особливості стандарту:

- система числення – двійкова;
- зміщений порядок;
- нормалізована мантиса має 1 у цілій частині;
- наявність схованого біта мантиси;
- три формати чисел: короткий – 32 біт; довгий – 64 біт;

тимчасовий ≥ 80 біт.

Короткий та довгий біти – стандартизовані, тимчасовий – визначається розробником архітектури комп'ютера. Формати чисел показано на рис. 1.4.

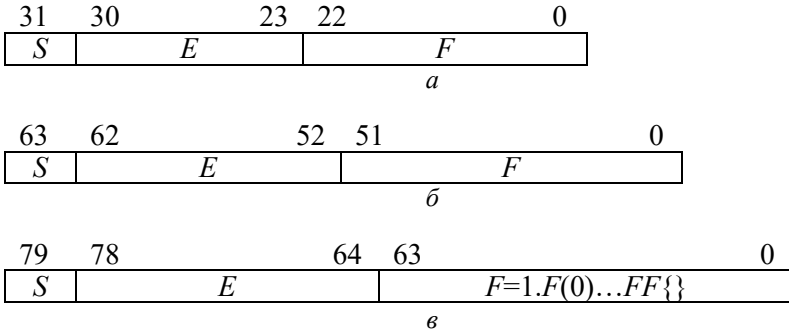


Рис. 1.4. Формати чисел IEEE-754: *a* – короткий; *б* – довгий; *в* – тимчасовий.

Кожний із форматів включає три поля: знак мантиси *S*, зміщений порядок *E*, мантису *F*. Мантиса *F* має цілу та дробову частини:

$$F = F(0).F(1) \dots F(m).$$

Ціла частина $F(0)$ має вагу 1 і в нормалізованому вигляді $F(0)=1$. Умова нормалізації мантиси:

$$1 \leq F < 2.$$

Число з ПК:

$$X = (-1)^S \cdot 2^{E-N} \cdot \underbrace{F(0)}_{\text{Ціла частина}} . \underbrace{F(1) \dots F(m)}_{\text{Дріб правильний}}.$$

Під час зберігання числа в пам'яті біт $F(0)$ – відсутній (схований біт).

Особливості форматів:

1. Короткий базовий – КБ.

Формат (рис. 1.4, *a*) містить знаковий біт *S*, 8-розрядний зміщений порядок *E* та 23-розрядний дріб мантиси *F*. Біт $F(0)$ цілої частини мантиси – схований.

Зміщений порядок:

$$E = N + \Pi = (2^{(p-1)} - 1) + \Pi = 127 + \Pi$$

Значення чисел *X* цього формату мають такі інтерпретації:

- 1). $E=255, F \neq 0$, то $X = \text{NaN}$;
- 2). $E=255, F = 0$, то $X = (-1)^S \cdot \infty$ – знакова нескінченність;

- 3). $0 < E < 255$, то $X = (-1)^{E-N} \cdot 1.F$ – нормалізоване число;
 4). $E=0, F \neq 0$, то $X = (-1)^S \cdot 2^{-126} \cdot (0.F)$ – ненормалізоване число;
 5). $E=0, F = 0$, то $X = (-1)^S \cdot 0$ – знаковий нуль.

Приклад: $X=100,2_{10}=1100100,001100_2$
 $F=1.1001000011$; $\Pi=6$;
 $E = N + \Pi = 127 + 6 = 133_{10} = 10000101_2$;
 $X: \underbrace{0}_S \underbrace{10000101}_E \underbrace{10000110 \dots 0}_{F\text{-дробова частина}}$

Діапазони:

$$X_{min} = 0 \ 00000001 \ 0\dots 01,$$

$$X_{max}: \underbrace{0}_S \underbrace{1111111}_E \underbrace{1 \dots 10}_F,$$

$$E=127+\Pi; \Pi_{min} = 1 - 127 = -126,$$

$$X_{min} = 2^{-126} \cdot (2^0 + 2^{-23}) \approx 2^{-126} \approx 10^{-38},$$

$$\Pi_{max} = 254 - 127 = 127,$$

$$X_{max} = 2^{127} \cdot (2^0 + (1 - 2^{-23})) = 2^{127}(1 + 1) \approx 2^{128} \approx 10^{38}.$$

2. Довгий подвійної точності (ДПТ).

Формат (рис. 1.4,б) містить знаковий біт S , 11-розрядний зміщений порядок E та 52-розрядний дріб мантиси F . Біт $F(0)$ цілої частини мантиси – схований. Зміщений порядок:

$$E = N + \Pi = (2^{(p-1)} - 1) + \Pi = 1023 + \Pi$$

Інтерпретації числа X у цьому форматі:

- 1). $E=2047, F \neq 0$, то $X=\text{NaN}$;
- 2). $E=2047, F = 0$, то $F = (-1)^S \cdot \infty$ – знакова нескінченність;
- 3). $0 < E < 2047$, то $X = (-1)^{E-N} \cdot 1.F$ – нормалізоване число;
- 4). $E=0, F \neq 0$, то $X = (-1)^S \cdot 2^{-1022} \cdot (0.F)$ – ненормалізоване число;
- 5). $E=0, F = 0$, то $X = (-1)^S \cdot 0$ – знаковий нуль.

Діапазони:

$$X_{min}: 0 \ 00 \dots 01 \ 0 \dots 01,$$

$$\begin{aligned}
 X_{max} &: \underbrace{0}_{1} \underbrace{11 \dots 10}_{11 \text{ розрядів}} \underbrace{1 \dots 11}_{52 \text{ розряди}}, \\
 X_{min} &= 2^{E-N} \cdot (2^0 + 2^{-52}) \approx 2^{1-1023} \approx 10^{-310}, \\
 X_{max} &= 2^{(2046-1023)} \cdot (2^0 + 2 - 2^{-52}) = 2^{1023} \cdot 2 \approx 2^{1024} \approx 10^{310}
 \end{aligned}$$

3. Тимчасовий з розширеною точністю (ТРТ).

У цьому форматі розряд мантиси $F(0)$ є явним. Незалежно від початкового формату числа з ПК під час його завантаження з пам'яті на регістрі процесора він перетворюється у формат ТРТ. Під час записування числа з ПК у пам'ять відбувається перетворення у початковий попередній стан. Формат ТРТ не є стандартизованим і визначається виробником. На рис. 1.4 зображений формат ТРТ реалізований у блоці FPU фірми Intel.

Значення параметрів формату: машинне слово – 80 біт, зміщений порядок – 15 біт, мантиса – 64 біт.

$$E = N + \Pi; N = 2^{p-1} = 16383.$$

$$\Pi_{min} = E_{min} - N = 0 - 16383 = -16383,$$

$$\begin{aligned} \Pi_{max} &= E_{max} - N = (2^{15} - 1) - N = (32768 - 1) - 16383 = \\ &= 16384 \end{aligned}$$

Діапазон:

$$X_{min}: 0\ 00 \dots 01\ 1 \dots 00,$$

$$X_{max}: \underbrace{0}_{1} \underbrace{11 \dots 110}_{115 \text{ розрядів}} \underbrace{1 \dots 11}_{64 \text{ розряди}},$$

$$X_{min} = 2^{1-16383} \cdot 2^0 = 2^{-16382} = 10^{-4965},$$

$$X_{max} = 2^{16383} \cdot (2^0 + (2^0 - 2^{-64})) = 2^{16384} = 10^{+4965},$$

Приклад: число X перевести у формати КБ, ДПТ, ТРТ.

$$X = -247,345 = -11110111,011; F = -1,1110111; \Pi = 7;$$

$$\text{КБ} \quad X: 1 \quad \underbrace{10000110}_8 \quad 11101110110 \dots 0;$$

$$E_{\text{КБ}} = 127 + 7 = 134$$

$$\text{ДПТ} \quad X: 1 \quad \underbrace{10000000110}_{11} \quad 11101110110 \dots 0;$$

$$E_{\text{ДВ}} = 1023 + 7 = 1030$$

$$\text{ТРТ} \quad X: 1 \quad \underbrace{10 \dots 0110}_{15} \quad 111101110110 \dots 0;$$

$$E_{\text{ВВ}} = 16383 + 7 = 16390.$$

Особливості використання форматів КБ, ДПТ, ТРТ.

У більшості випадків для початкових даних та результатів обчислень рекомендується використовувати формат ДПТ.

Формат КБ використовують у випадку обмежень на пам'ять, а також при налаштуванні програм. Точність та діапазон незмінні.

Формат ТРТ використовується для проміжних результатів. Великий діапазон та точність гарантують захист результатів від помилок округлення та суттєво зменшують вірогідність появи переповнень та антипереповнень.

Методичні рекомендації

Домашнє завдання є індивідуальним і складається з двох частин.

В першій частині ДЗ потрібно виконати перетворення двійкових форматів даних, що відповідають розглянутим вище форматам у десяткові еквіваленти даних.

Варіанти завдань першої частини ДЗ наведено у табл.1.1. Число варіантів – 15.

У табл. 1.1 використанні такі позначення: p , m – розрядності порядку і мантиси відповідно (класичний формат); S – слово, DS – подвійне слово; ДПТ – довгий, КБ – короткий та ТРТ – тимчасовий формати.

У другій частині ДЗ слід реалізувати перетворення десяткових числових даних у відповідні машинні формати. Варіанти завдань наведено у табл. 1.2. Використані скорочення аналогічні наведеним у табл.1.1.

Джерела: [2].

Таблиця 1.1

Варіанти ДЗ 1 (перша частина)

№ варіанта	Числа з ПК	Формат
1	1 110...0101..1001 0 10111	$p=5, m=16$
	0 1001101 0101...100...011100011000	IBM-360, С
	1 11001010 10...01...101...10...01010	DEC, С, ДС
	1 11111111 1...10...0101	IEEE-754, КБ
2	0 10101...101...10 1 100100	$p=6, m=18$
	1 1001111 01...10...01101	IBM-360, С, ДС
	0 1000100 1...100...01010...101	DEC, С, ДС
	0 11111111 0...01	IEEE-754, ДПТ
3	0 110010101...10 1 001110	$p=6, m=16$
	1 0101011 0101...00...11011...101	IBM-360, С, ДС
	0 0000100 1001...100...011	DEC, С
	1 10001101 1000...01...11	IEEE-754, ТРТ
4	0 1...100101...1010 1 10101	$p=5, m=20$
	1 1111111 000101...10...0101	IBM-360, ДС
	0 10000001 1000...1001	DEC, С
	1 100101010101 0010110...0101	IEEE-754, ТРТ
5	0 10...01...1...10101 1 010101	$p=6, m=12$
	1 00...0 0001010...0101	IBM-360, С
	0 00...0 000...01...1011	DEC, ДС
	1 11111111111 0101...1	IEEE-754, ДПТ
6	0 10...01...101101 11001101	$p=7, m=18$
	1 1111111 00010...011...101	IBM-360, С
	0 11000001 0001...10101	DEC, С
	1 01011101010 00...0101	IEEE-754, ДПТ
7	0 1010...010...01 1 001111	$p=6, m=18$
	1 1010101 1000...01...1011	IBM-360, С
	1 10101000 0000...01...1000	DEC, С
	0 11000011011 00...011...11001	IEEE-754, ТРТ
8	1 10100...01101 1 0011011	$p=7, m=16$
	0 1001010 000100...01...1001	IBM-360, С
	0 10010000 0001000...010001	DEC, С
	0 000101...0 0010...1011...101	IEEE-754, ТРТ
9	1 100011...1 1 0101	$p=4, m=16$
	0 0010001 0101...100...0101	IBM-360, С
	1 10001010 0000...0111...101	DEC, С
	1 000...0000 000...0110	IEEE-754, КБ

№ варіанта	Числа з ПК	Формат
10	1 11...011...100...01 1 001110	$p=6, m=18$
	0 1110000 000101...101	IBM-360, С
	0 11101110 0010...0...0101	DEC, С
	0 0...0 0...0	IEEE-754, KB
11	0 1...10...001101 1 0001110	$p=7, m=16$
	1 0000011 00010...01...101	IBM-360, С
	1 0000001 00100...01...101	DEC, С
	1 01100100111 01001	IEEE-754, BB
12	1 01...10...0 0 101011	$p=6, m=12$
	0 1001111 00111...10...0	IBM-360, ДС
	1 10011100 00...011...10101	DEC, ДС
	1 00001111 0...01...101	IEEE-754, ДВ
13	1 10...01...101 1 110000	$p=6, m=20$
	0 1011001 010011...1011	IBM-360, С
	1 10000001 01001...10001	DEC, С
	1 00...010101 10...11...10...0101	IEEE-754, ТРТ
14	1 10...01...101 0 010110	$p=6, m=24$
	1 0001111 000110...0...0111	IBM-360, ДС
	1 00011110 000110...0...0111	DEC, С
	0 1...1001...10 000000...01...1000	IEEE-754, ТРТ
15	1 1...1000...0111 1 11010	$p=5, m=24$
	1 0011100 100011...011	IBM-360, С
	1 00100000 000...0...01	DEC, С
	1 00000001 00...0...0111	IEEE-754, KB

Таблиця 1.2

Варіанти ДЗ 1 (друга частина)

№ варіанта	Дані	Класичний		IBM-360/370	DEC	IEEE-754
		m	p			
1	-27/64	12	4	С	ДС	КБ
	+375,75	18	5	ДС	С	ДПТ
	-0,0037	24	6	С	ДС	ТРТ
	+10 ⁶	32	7	ДС	С	КБ
	+725/10 ⁴	64	8	С	ДС	ДПТ
2	-50	10	5	С	ДС	КБ

	$0,3 \cdot 10^{-3}$	12	7	ДС	С	ДПТ
	-35000	18	6	С	ДС	ТРТ
	51,738	32	7	ДС	С	КБ
	$-6 \cdot 10^3$	24	6	С	ДС	ДПТ
3	21.713	24	5	С	ДС	КБ
	$0,5 \cdot 10^5$	80	6	ДС	С	ДПТ
	$-0,6 \cdot 10^{-4}$	50	6	С	ДС	ТРТ
	$+700 \cdot 10^5$	24	6	ДС	С	КБ
	$-9/10^5$	50	4	С	ДС	ДПТ
4	+51/704	80	6	С	ДС	КБ
	-375/20	50	4	ДС	С	ДПТ
	$3 \cdot 10^5$	24	6	С	ДС	ТРТ
	$-50 \cdot 10^{-2}$	32	7	ДС	С	КБ
	0,6254	36	6	С	ДС	ДПТ
5	25/37	60	5	С	ДС	КБ
	$-20 \cdot 10^3$	50	6	ДС	С	ДПТ
	30576/41	40	5	С	ДС	ТРТ
	0,0006	12	3	ДС	С	КБ
	$-16 \cdot 10^{-4}$	24	5	С	ДС	ДПТ
6	50/785	12	4	С	ДС	КБ
	5,7853	24	6	ДС	С	ДПТ
	$62 \cdot 10^{-3}$	18	5	С	ДС	ТРТ
	$-378 \cdot 10^3$	28	6	ДС	С	КБ
	6/7954	80	7	С	ДС	ДПТ
7	-415/56	28	6	С	ДС	КБ
	$+415 \cdot 10^2$	51	5	ДС	С	ДПТ
	0,00751	48	6	С	ДС	ТРТ
	$-0,7 \cdot 10^3$	24	5	ДС	С	КБ
	75312/6	18	6	С	ДС	ДПТ

Продовження табл. 1.2

№	Дані	Класичний		ІВМ-360/370	DEC	ІЕЕЕ-754
		<i>m</i>	<i>p</i>			
8	$50 \cdot 10^4$	80	7	С	ДС	КБ
	$45 \cdot 10^{-2}$	60	6	ДС	С	ДПТ
	$0,7 \cdot 10^3$	50	6	С	ДС	ТРТ
	$-87 \cdot 10^{-2}$	48	5	ДС	С	КБ
	0,9865	24	6	С	ДС	ДПТ
9	-27/645	14	5	С	ДС	КБ
	$51 \cdot 10^{-2}$	24	6	ДС	С	ДПТ

	378·10 ²	28	5	С	ДС	ТРТ
	-415/158	60	6	ДС	С	КБ
	45·10 ³	80	5	С	ДС	ДПТ
10	-20·10 ⁻³	60	5	С	ДС	КБ
	378·10 ³	50	6	ДС	С	ДПТ
	5/100	40	6	С	ДС	ТРТ
	5016052	32	6	ДС	С	КБ
	415000	24	7	С	ДС	ДПТ
11	50/1785	18	7	С	ДС	КБ
	30576/27	24	6	ДС	С	ДПТ
	10 ⁵	40	12	С	ДС	ТРТ
	0,00006	24	5	ДС	С	КБ
	-16·10 ⁻⁴	32	6	С	ДС	ДПТ
12	415/628	28	6	С	ДС	КБ
	0,000075	30	5	ДС	С	ДПТ
	-700/10 ⁻³	24	6	С	ДС	ТРТ
	382·10 ⁴	28	6	ДС	С	КБ
	-87,125	30	7	С	ДС	ДПТ
13	3/315	24	5	С	ДС	КБ
	315·10 ²	30	6	ДС	С	ДПТ
	726·10 ⁻³	40	5	С	ДС	ТРТ
	500000	30	6	ДС	С	КБ
	0,625/3	75	8	С	ДС	ДПТ
14	0,95·10 ⁻⁵	32	5	С	ДС	КБ
	50,606	24	4	ДС	С	ДПТ
	0,6267	32	4	С	ДС	ТРТ
	51/151	32	6	ДС	С	КБ
	-0,0006	24	7	С	ДС	ДПТ

Закінчення табл. 1.2

№	Дані	Класичний		ІВМ-360/370	DEC	ІБЕЕ-754
		<i>t</i>	<i>p</i>			
15	62·10 ⁻³	20	5	С	ДС	КБ
	51·10 ³	18	6	ДС	С	ДПТ
	-0,7·10	28	5	С	ДС	ТРТ
	62/375	30	6	ДС	С	КБ
	0,6256	32	4	С	ДС	ДПТ

Домашнє завдання 2

Арифметика чисел з ПК

Мета: дослідити особливості виконання алгоритмів арифметичних операцій (АО) над числами з ПК.

Операції над числами з ПК є значно складнішими від операцій над числами з фіксованою комою. Тому арифметичний пристрій для цих операцій є, як його називають, відокремленим самостійним пристроєм – співпроцесором, що забезпечує реалізацію операцій додавання, віднімання, множення та ділення.

Арифметична операція над числами з ПК в загальному вигляді записується так:

$$Z = X \cdot Y, \quad \underbrace{F_z 2^{E_z}}_Z = \underbrace{F_x 2^{E_x}}_X \cdot \underbrace{F_y 2^{E_y}}_Y,$$

де F_z, F_x, F_y – нормалізовані мантиси; E_z, E_x, E_y – зміщені порядки результату та операндів.

Числа з ПК складаються зі знаку числа, зміщеного порядку E та мантиси F . Операції над їх складовими виконуються окремими апаратними блоками.

Для зберігання операндів та результату в арифметичному пристрої є блок регістрів з плаваючою комою.

У випадку стандарту IEEE-754 при завантаженні регістрів відновлюється старший біт мантиси, який був «схований» під час запису у пам'ять.

Послідовність кроків реалізації АО:

1. На початковому етапі АО виконується перевірка на рівність нулю операндів. Це пришвидшує АО.

2. При завершенні АО необхідно:

- виявити нульове значення мантиси;
- нормалізувати мантису;
- визначити від'ємне переповнення порядку (антипереповнення).

Нульове значення мантиси можливе при додаванні (відніманні) мантис, а також зсуві мантиси вправо у випадку переповнення.

Нормалізація мантиси виконується зсувом її вліво до появи у старшому розряді 1. Із кожним зсувом мантиси на 1 зменшується значення порядку результату.

2.1. Додавання та віднімання

Операції додавання та віднімання складніші за аналогічні операції над числами з фіксованою комою і виконуються приблизно.

Алгоритм додавання (віднімання) складається з такої послідовності кроків:

- перевірка операндів на рівність нулю;
- вирівнювання порядків;
- додавання (віднімання мантис);
- нормалізація результату;

Операції над мантисами F_x, F_y та порядками E_x, E_y складових X, Y виконуються роздільно (окремо). Значення суми:

$$Z = X + Y = F_z 2^{E_z}$$

Перевірка операндів на рівність нулю

Цей етап виконується тривіально.

Вирівнювання порядків

Порівняння порядків виконується шляхом віднімання другого порядку з першого. При цьому можливі такі випадки:

- $(E_x - E_y) > m$ (m – розрядність мантиси). У цьому випадку результатом операції є перший операнд, оскільки при вирівнюванні порядків усі розряди мантиси другого операнду становляться нульовими.

- $(E_y - E_x) > m$. Результатом операції є другий операнд.

- $(E_x - E_y) = 0$. Порядки операндів однакові. Виконується додавання (віднімання) мантис. Після отримання результату суми (різниці) аналізуємо стан розряду переповнення суматора мантиси та стан старшого розряду. Якщо в розряді переповнення стоїть одиниця, то виконується нормалізація вправо; якщо в старшому розряді – нуль, то виконується нормалізація вліво;

- $(E_x - E_y) = k < m$. Мантиса другого операнду зсувається на « k » розрядів вправо, і виконується додавання мантис. Результату присвоюється порядок першого операнду, якщо необхідно, слідує нормалізація;

- $(E_x - E_y) = -k$ ($|k| < m$). Мантиса першого операнду зсувається на « k » розрядів вправо. Потім мантиси додаються і результату присвоюють порядок другого операнда.

Додавання (віднімання) мантис і нормалізація результату

Після вирівнювання порядків мантиси операндів додаються (віднімаються) як числа з фіксованою комою. У разі переповнення результату він нормалізується вправо. При цьому можливе додатне переповнення порядку, формується ознака переповнення і операція припиняється.

У випадку, коли мантиса суми (різниці) має значення

$$0 < F_z < 1,$$

то вона перевіряється на необхідність нормалізації. У разі додатної ознаки виконується нормалізація зсувом вліво.

Якщо $F_z = 0$, нормалізація неможлива, значимість мантиси загубиться. Незалежно від значення порядку E_z результатом операції буде $Z = 0$.

Якщо $F_z \neq 0$ і $F_z(0) = 1$, мантиса є нормалізованою, якщо

$F_z(0) = 0$, мантиса нормалізується зсувом вліво, порядок суми E_z зменшується на одиницю. Після зсуву мантиси необхідно перевірити порядок результату на від'ємне переповнення. Якщо це має місце, то результат не нормалізується. Мантиса зсувається на один розряд вправо, порядок збільшується на одиницю і набуває мінімального від'ємного значення.

Збільшення точності операції додавання (віднімання)

На відміну від додавання чисел з фіксованою комою операція додавання чисел з ПК є приблизною. Похибка виникає при вирівнюванні порядків операндів. Зменшити похибку можна шляхом арифметичного округлення суми мантис. Для цього мантиси операндів мають додатковий розряд. Після додавання мантис операндів сума округляється за значенням додаткового розряду результату.

2.2. Множення

Ця операція значно простіше від операції додавання (віднімання).

Послідовність кроків:

- співмножники перевіряються на рівність нулю. Якщо умова виконується, результат дорівнює нулю і видається у відповідному форматі чисел з ПК;

- додавання порядків. Якщо використовується зміщений порядок, то отримана сума буде мати подвоєне зміщення і її необхідно зменшити на величину зміщення. За цих дій з порядками можливе переповнення порядку, а також і губиться значущість. В обох випадках виникає аварійна ситуація;

- перемноження мантис як чисел з фіксованою комою (правильний дріб) з округленням добутку до довжини мантиси;

- нормалізація результату.

2.3. Ділення

Послідовність кроків:

- ділене та дільник аналізуються на рівність нулю. Якщо нулю дорівнює дільник, то залежно від реалізації архітектури комп'ютера видається повідомлення про ділення на нуль або результатом визначається нескінченність. Коли ділене нуль, то результатом також буде нуль;

- віднімання порядку дільника із порядку діленого, що приводить до зникнення зміщення. Тому для отримання зміщення порядку необхідно до різниці додати зміщення. При цьому також можливі переповнення порядку та зникнення значущості;

- ділення мантис як чисел з фіксованою комою;

- нормалізація та округлення результату.

Методичні вказівки

Домашнє завдання є індивідуальними, у яких виконують обчислення (операції) над числами з ПК. Варіанти завдань наведено в табл. 2.1. Число варіантів – 15.

Джерела: [1].

Таблиця 2.1

Варіанти ДЗ

№ варіанта	Функція $F:=$	Z_1	Z_2	Z_3	Класичний		DE C	IB M	IEEE - 754
					й				
					m	p			
1	$Z_1 \cdot Z_2 / Z_3$	0,5	500	0,06	10	5	C	ДС	KB

2	$(Z_1 - Z_3) \cdot Z_1$	10^{-1}	0,75	750	—	—	ДС	—	КВ
3	$Z_1 / Z_3 + Z_2$	25	750	100	12	4	С	—	—
4	$(Z_1 + Z_2) / Z_3$	2,5	7	0,04	—	—	ДС	С	—
5	$(Z_1 \cdot Z_2) + Z_3$	10	80	$\frac{100}{0}$	10	6	—	—	ВВ
6	$Z_2 \cdot Z_3 - Z_1$	10^3	0,05	60	—	—	С	—	ДВ
7	$Z_2 + Z_3 - Z_1$	0,05	$2 \cdot 10^2$	0,03	—	—	С	—	КВ
8	$Z_3 / Z_2 - Z_3$	5	0,05	500	20	6	—	С	—
9	$Z_3 \cdot Z_2 + Z_1$	$5 \cdot 10^{-1}$	60	200	—	—	С	—	КВ
10	$Z_1 \cdot Z_3 - Z_2$	20	0,85	150	12	6	ДС	—	—
11	$Z_2 \cdot Z_3 + Z_1$	25	0,05	10^{-1}	—	—	—	С	КВ
12	$Z_1 - Z_2 \cdot Z_3$	0,05	32	10^{-2}	16	6	—	С	—
13	$Z_2 \cdot Z_3 - Z_1$	50	3,25	0,06	8	6	С	—	—
14	$Z_1 / Z_3 \cdot Z_2$	10^{-2}	32	10^2	—	—	С	—	ДВ
15	$Z_2 + Z_3 / Z_1$	100	1000	0,52	16	6	—	С	ВВ

Домашнє завдання 3 Комп'ютер з ефективним набором команд

Мета: Визначення ефективного набору команд комп'ютера при мінімальному часі прогону (рішенні) НТЗ.

Комп'ютерне рішення задачі можливе при виконанні різних наборів операцій. Ефективність набору операцій в загальному вигляді визначається витратами устаткування та часу на рішення задачі.

У домашньому завданні реалізується фрагмент архітектурного етапу проектування комп'ютера, який полягає в орієнтуванні архітектури (системи операцій) на ефективне розв'язання задачі. Множину наборів визначають два набори операцій, склад яких залежить від класу (типу) задач та методу (алгоритму) її розв'язання.

3.1. Час прогону задачі

Будемо вважати, що комп'ютер має синхронне керування і постійну тривалість тактів процесора та пам'яті. Час розв'язання (прогону) задачі визначається за формулою:

$$t_{np} = \tau \sum_{i=1}^k p_i m_i + 1,5\tau m_{\Pi} + \tau(m_{jz} + m_{jmp}) + \tau_{оп} m_{оп},$$

де τ – тривалість процесорного такту; k – кількість типів операцій обробки (наприклад, АО); p_i – середня кількість процесорних тактів в i -й операції; m_i – кількість звернень до i -ї операції; m_{Π} – кількість операцій пересилань (ST, LD); m_j, m_{jmp} – кількість операцій умовного та безумовного переходів; $m_{оп}$ – кількість звернень до основної пам'яті; $\tau_{оп}$ – тривалість циклу основної пам'яті.

Припустімо, що тривалість операції пересилань дорівнює $1,5\tau$, а операцій переходу JZ, JMP – 1τ .

Час прогону задачі виконується в такій послідовності.

1. Задаються набори операцій, які потрібні для розв'язання задачі.

2. Розробляються схеми алгоритмів розв'язання задач. За схемою алгоритму визначаються величини m_i . При цьому потрібні враховувати циклічні ділянки в схемі алгоритму. Якщо число повторень не визначено, то їхня кількість береться довільно, наприклад, 5.

3. Для кожної операції розробляється мікропрограма у вигляді графа мікропрограми (ГМП) для визначення кількості процесорних тактів « p_i ». Якщо в ГМП є розгалуження, то у формулу (3.1) входить середнє значення

$$p_i = \frac{p_{imax} + p_{imin}}{2},$$

4. За схемою алгоритму задачі складається програма комп'ютера в змістовній формі з урахуванням її адресності та обчислюються параметри $m_z, m_{оп}, m_z, m_{jnp}$ та ін.

5. За формулою (3.1) обчислюється час прогону науково-технічної задачі..

3.2. Ефективний набір операцій

Ефективний набір операцій визначимо на прикладі рішення задачі $F=Sh(z_1/z_2)$. Адресність комп'ютера АК=2.

Схеми алгоритмів задачі F

Кількість наборів операцій визначається типом задачі та методом (алгоритмом) її розв'язання. Кількість можливих наборів операцій буде два. Формування наборів операцій розглянемо на прикладі обчислення функції $F = Sh(z_1/z_2)$. Для цього використаємо розкладання її у степеневий ряд:

$$F = ShZ = Z - \frac{Z^3}{3!} + \frac{Z^5}{5!} + \dots = U_0 + \sum_{k=1}^n U_k, (k = 1, 2, \dots, n).$$
$$U_0 = Z, U_k = U_{k-1} \cdot \left(\frac{Z^2}{2k \cdot (2k + 1)} \right).$$

На основі аналізу задачі та методу її розв'язання задаємося двома наборами операцій:

$\{/, +, -, \cdot\}; \{/, Sh\}$.

Наборам операцій відповідають схеми алгоритмів (рис. 3.1).

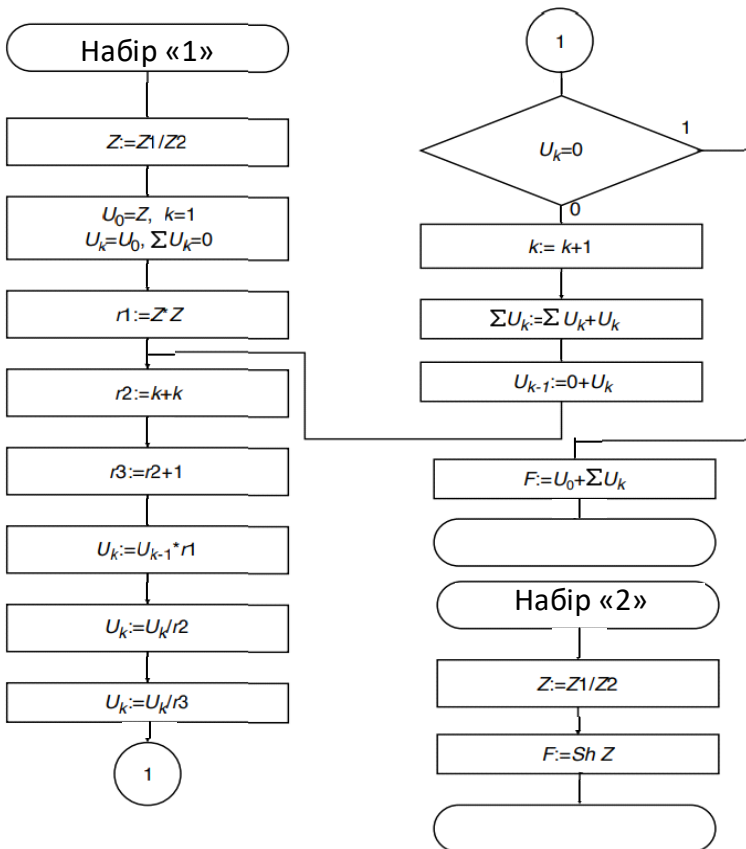


Рис. 3.1. Схеми алгоритма розв’язання задачі $F=Sh(Z_1/Z_2)$: Набір «1» – набір операцій «1»; Набір «2» – набір операцій «2».

Програми задачі F у змістовній формі

У табл.3.1 представлені програми задачі F у змістовній формі, що відповідають двом наборам операцій: набір «1» та набір «2». При цьому враховується адресність команд (2-адресні). У програмі набір «1» є циклічна ділянка (команди 5 – 22). Вважаємо, що умова $U_k = 0$ виконується при п’яти ітераціях. Програма набір «2» – лінійна. Цим програмам відповідають дві системи команд (набір «1» та набір «2»).

Таблиця 3.1

Програми задачі F у змістовній формі						
набір "1"				набір "2"		
№	OP	A_1	A_2	OP	A_1	A_2
1	/	$Z1$	$Z2$	/	$Z1$	$Z2$
2	ST	Z	-	ST	Z	-
3	*	Z	Z	Sh	Z	-
4	ST	$r1$	-	ST	F	-
5	+	k	k	HLT	-	-
6	ST	$r2$	-			
7	+	$r2$	$\langle 1 \rangle$			
8	ST	$r3$	-			
9	*	U_{k-1}	$r1$			
10	ST	U_k	-			
11	/	U_k	$r2$			
12	ST	U_k	-			
13	/	U_k	$r3$			
14	ST	U_k	-			
15	JZ	-	23			
16	+	k	$\langle 1 \rangle$			
17	ST	k	-			
18	+	ΣU_k	ΣU_k			
19	ST	ΣU_k	-			
20	+	$\langle 0 \rangle$	U_k			
21	ST	U_{k-1}	-			
22	JMP	-	5			
23	+	U_0	ΣU_k			
24	ST	F	-			
25	HLT	-	-			

Набір "1" = {/, *, +, ST , JZ , JMP , HLT }.

Набір "2" = {/, Sh , ST , HLT }.

З цих командних наборів необхідно вибрати ефективний.


Об'єднаємо набори операцій, щоб отримати список операцій, для яких необхідно розробляти мікропрограми {/, *, +, ST , JZ , JMP , Sh , HLT } \cup {/, Sh , ST , HLT } = {/, *, +, ST , JZ , JMP , Sh , HLT }.

Мікропрограми машинних операцій

Мікропрограми відповідають такту виконання операції визначають число процесорних тактів " p_i ".

1. Операція ділення " $/$ ".

Цикл команди " $/$ " $T_{BK} \quad T_d \quad T_o \quad T_l \quad t$.



Для визначення числа тактів " p " необхідно знати розрядність операндів " n ", алгоритм ділення. Будемо вважати, що $p_i = n \cdot S_i$, де S_i – число процесорних тактів при одному прогоні операції ділення.

$$p = 16 \cdot 3 = 48 \text{ тактів.}$$

2. Операція додавання " $+$ ".

Цикл команди " $+$ " $T_{BK} \quad T_d \quad T_o \quad T_+ \quad t$.



Вважаємо, що $p_+ = 1$ такт.

3. Операція множення " $*$ ".

Цикл команди " $*$ " $T_{BK} \quad T_d \quad T_o \quad T_* \quad t$.



$$P_* = 16 \cdot 3 = 48 \text{ тактів.}$$

4. Операції JZ, JMP, ST, HLT .

Вважаємо, що ці операції виконуються за один процесорний такт.

$$p \equiv 1 \text{ такт.}$$

5. Операція Sh

Цикл команди " Sh " $T_{BK} \quad T_d \quad T_o \quad T_{Sh} \quad t$.



Для визначення числа процесорних тактів p_{Sh} операції Sh необхідно по схемі алгоритму (рис. 1, набір "1") розробити ГМП функції Sh (рис. 3.2).

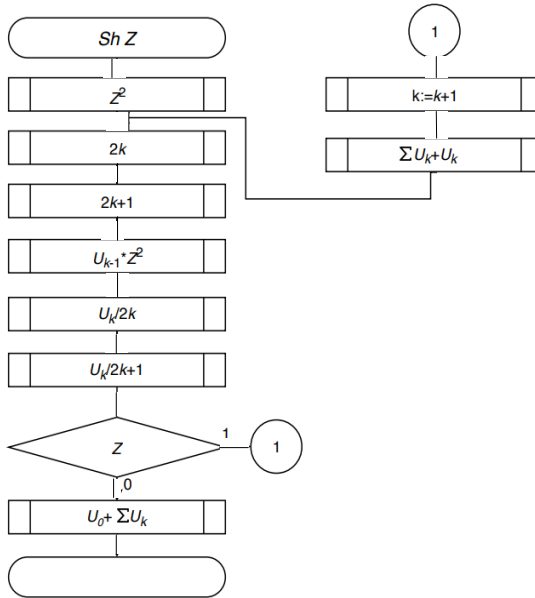


Рис. 3.2. ГМП “Sh”

Орієнтовано кількість процесорних тактів для реалізації функції $Sh \epsilon$:

$$p_{Sh} = p_* + (p_{2k} + p_{2k+1} + p_* + p_l) \cdot 5 + p_+ .$$

ГМП “Sh” є послідовністю мікропрограм операцій (*, +, /) з внутрішньою циклічною ділянкою (число циклів 5).

Ефективний набір операцій

Обчислимо час рішення задачі для наборів 1 та 2. Для цього визначимо значення параметрів $m_i, m_{\Pi}, m_{jz}, m_{jmp}, m_{оп}$ для кожного з наборів.

Час рішення задачі для обох наборів:

$$t_1 = \tau \sum_{i=1}^k p_i \cdot m_i + 1,5\tau m_{\Pi} + \tau(m_{jz} + m_{jmp}) + \tau_{оп} m_{оп} ,$$

$$t_2 = \tau \sum_{i=1}^k p_i \cdot m_i + 1,5\tau m_{\Pi} + \tau(m_{jz} + m_{jmp}) + \tau_{оп} m_{оп} .$$

Порівняння значень t_1, t_2 визначає ефективний набір операцій.

Методичні вказівки

Домашнє завдання є самостійною роботою студента, в якій визначається ефективний набір команд. Для цього виконується така послідовність кроків.

1. На основі аналізу задачі та алгоритму її рішення визначаються можливі базові набори операцій перетворення. В ДЗ число цих наборів – 2.

2. Розробляють схеми алгоритмів для кожного з цих наборів операцій.

3. По схемах алгоритмів рішення задачі написати програми у змістовній формі та отримати можливі набори машинних операцій.

4. Об'єднати отримані набори машинних операцій.

5. Для машинних операцій розробляються мікропрограми і визначають середню кількість процесорних тактів.

6. Для кожного набору операцій обчислюють час прогону задачі t_{n1}, t_{n2}, \dots . Мінімальний час визначає ефективний набір команд комп'ютера.

7. Варіанти завдань наведено в табл. 3.2. В ДЗ прийняті такі скорочення: ФК, ПК – фіксована, плаваюча кома; АК – адресність команди, τ – такт процесора, $t_{ОП}$ – цикл пам'яті.

Джерела: [3]

Варианти ДЗ

№	$F : +$	АК	Дані	τ , нс	$t_{оп}$	z_1	Z_2
1	chz_{1+z_2}	3	ФК	5	10	0,5	0,8
2	shz_{1/z_2}	2	ПК	10	15	5	8
3	$cos z_{2+z_1/z_3}$	1	ФК	5	10	5	2
4	$sin z_{1+1}$	3	ПК	6	20	6	-
5	shz_{2-z_1}	2	ФК	10	10	-	2
6	shz_{1/z_2}	1	ПК	8	10	-	0,4
7	$cos z_2 - z_1$	1	ФК	10	15	3	0,5
8	$z_2 - chz_2$	1	ПК	5	10	-	0,7
9	shz_{1/z_2}	2	ФК	10	15	0,5	0,8
10	chz_{2/z_1}	3	ПК	5	10	2	3
11	z_{2+1-1}	2	ФК	10	15	6	-
12	shz_{2-1}	1	ПК	15	20	-	5
13	chz_{1/z_2}	2	ФК	10	20	-	8
14	$exp(z_{2+z_1})$	3	ПК	5	10	6	0,8
15	$exp(z_2/z_1)$	-	ФК	10	15	0,8	0,3

Список літератури

1. Орлов С.А. Организация ЭВМ и систем: учебник / С.А. Орлов, Б.Я. Цилькер. – СПб.: Питер, 2011. – 688 с.
2. Таненбаум С., Архитектура компьютера. / С. Таненбаум, Т. Остин. — СПб.: Питер, 2013 — 816 с.
3. Архітектура обчислювальних машин. Методичні рекомендації до виконання домашнього завдання. – К.: НАУ, 2011. – 36 с.