Полтавський національний технічний університет імені Юрія Кондратюка

Лекція № 4 з навчальної дисципліни “Архітектура обчислювальних систем”. Розділ 1. Організація апаратної частини комп'ютерів

Модуль 1. Організація ядра обчислювальної системи (центрального процесора, пам’яті та системного інтерфейса). Методи розміщення інформації у пам’яті комп’ютерів.

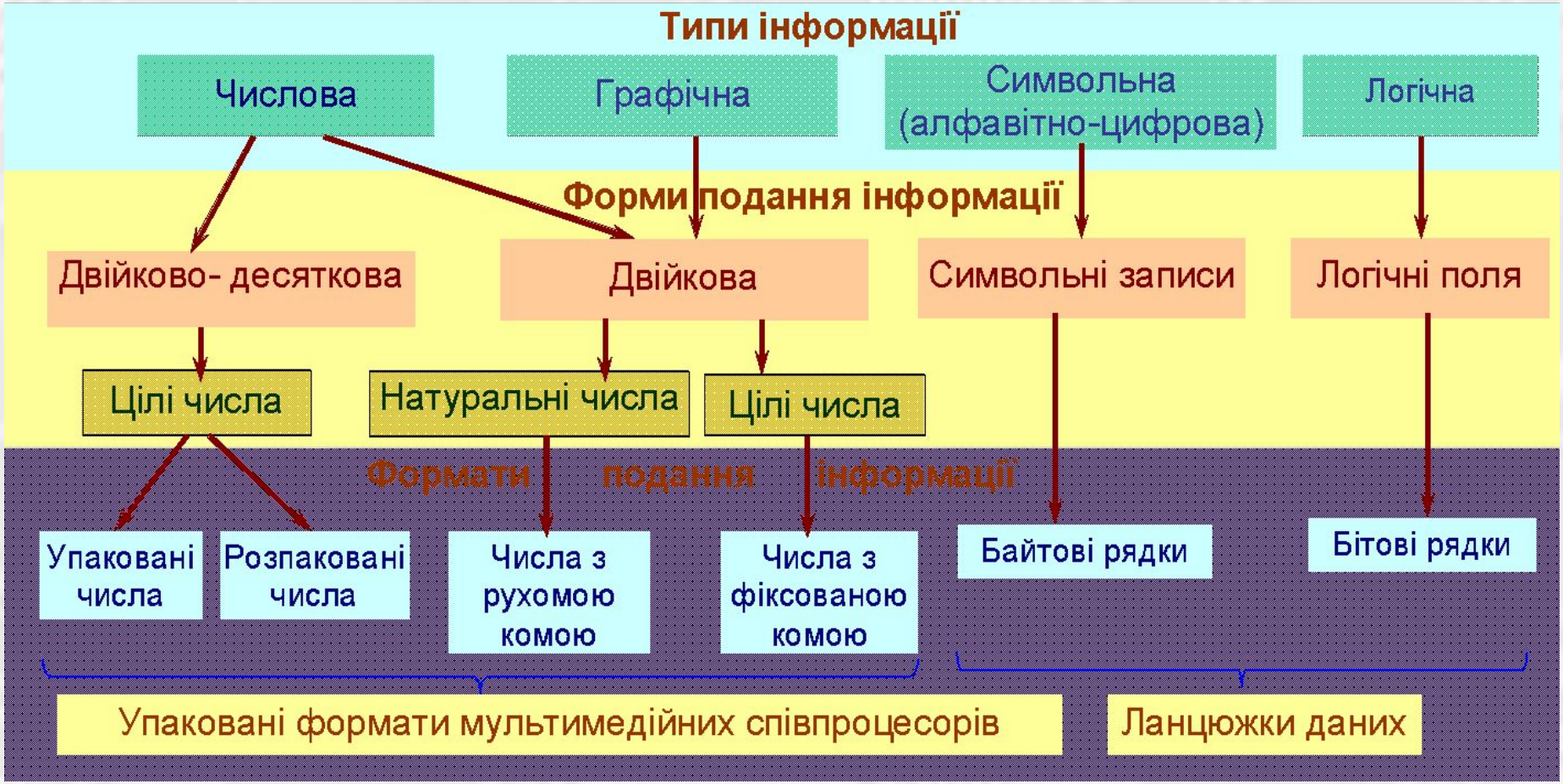
Тема лекції:
Представлення даних та команд у комп’ютері

План лекції

1. *Типи, форми та формати подання інформації у комп'ютері.*
2. *Формати представлення двійкових чисел.*
3. *Кодування символьної та логічної інформації.*
4. *Кодування команд в комп'ютері.*
5. *Класифікація архітектур комп'ютера за складом системи команд*

Кафедра комп'ютерної інженерії
К.т.н., доцент Тиртишніков О.І.

1. Типи, форми та формати подання інформації у комп'ютері



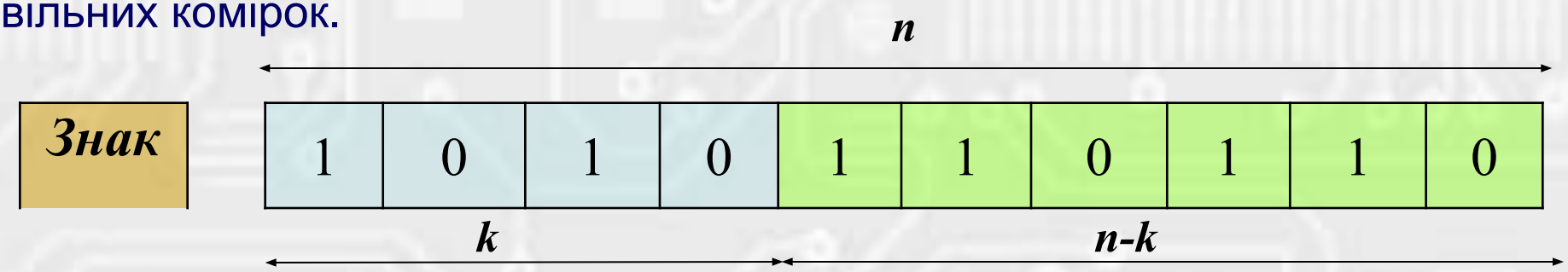
Сукупність комірок пам'яті, призначених для розміщення одного двійкового числа, називають ***розрядною сіткою***. Довжина розрядної сітки обмежена та складає 2^n байтів. ***Формати подання даних*** - це правила розміщення інформації в розрядній сітці машини.

2. Формати представлення двійкових чисел

Існують два способи представлення двійкових чисел: із *фіксованою комою* і з *рухомою (або «плаваючою») комою* (відповідно *природна* і *напівлогарифмічна* форми представлення двійкових чисел).

Числа з фіксованою комою

Якщо для розміщення цілої частини числа виділяється *k* комірок *n+1* розрядної сітки, то для розміщення дробової частини залишиться *n - k* вільних комірок.



Якщо кількість розрядів у дробовій частині числа перевищує *n - k,* то деякі молодші розряди виходять за межі розрядної сітки і не будуть сприйматися обчислювальним пристроєм. Отже, будь-яке двійкове число, менше, ніж одиниця молодшого розряду розрядної сітки, сприймається як нуль і називається ***машинним нулем***. У результаті відкидання молодших розрядів дробової частини числа, розташованої за межами розрядної сітки, виникає похибка подання. Максимальне значення абсолютної похибки подання не перевищує одиниці молодшого розряду сітки.

2. Формати представлення двійкових чисел

Найчастіше вважається, що кома знаходиться після молодшого розряду числа (цілі числа) або перед старшим розрядом числа (дробові числа). В сучасних комп'ютерах формат використовується для подання цілих чисел.

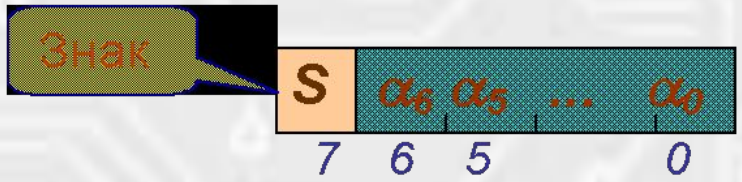
Формат цілого числа без знаку

$0 <= X <= 2^n - 1$

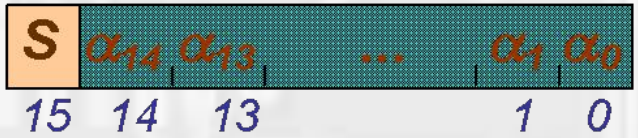
Формат цілого числа зі знаком

$0 <= abs(X) <= 2^{n-1} - 1$

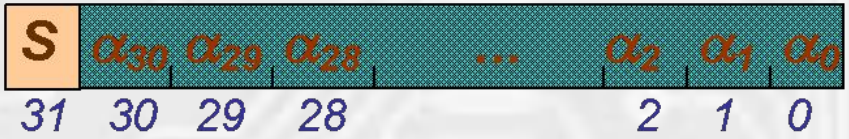
Типові формати цілих чисел, що використовуються в ПК



Короткий однобайтовий формат (байт)



Короткий двобайтовий формат (слово)



Нормальний формат (подвійне слово)

Довгий формат

2. Формати представлення двійкових чисел

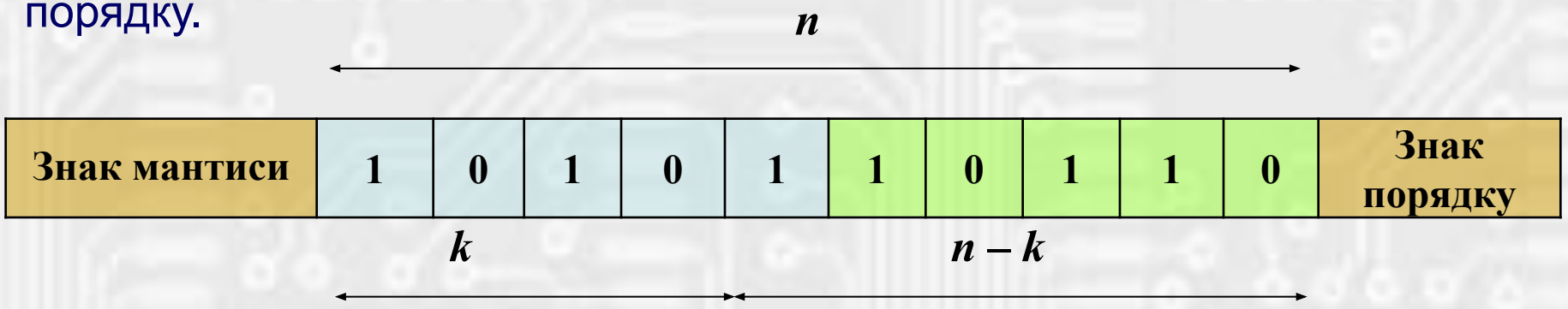
Числа із рухомою комою

Основною є форма представлення чисел із рухомою («плаваючою») комою. Її використання дозволяє суттєво розширити діапазон і зменшити відносну похибку подання чисел. У цій формі числа подаються у вигляді добутку деякого ступеня основи системи числення ***q*** і частини, що має вигляд правильного дробу:

$$N = \pm aq^{\pm p}$$

де ***p*** – порядок числа, ***a*** – його мантиса. Мантиса і порядок є знаковими числами. Тому для позначення знаків у розрядній сітці відводяться два додаткові розряди. Знак усього числа співпадає із знаком мантиси.

Під час запису двійкового числа у показовій формі в розрядній сітці використовуються дві групи розрядів (без урахування знакових розрядів мантиси і порядку). Перша група (*k* розрядів) призначена для розміщення коду мантиси, друга (*n* - *k* розрядів) – для розміщення коду порядку.



2. Формати представлення двійкових чисел

Мантиса числа може приймати необмежену кількість різних значень, менших за одиницю, при відповідних значеннях порядку (тобто кома може рухатися або «плавати»). З усіх можливих варіантів представлення числа у показовій формі той, що не має в старшому розряді мантиси нуля, називають ***нормалізованим.*** Всі інші варіанти є ненормалізованими***. У нормалізованій формі значення мантиси завжди більші або рівні 1/2, але не перевищують одиниці.***

У обчислювальних пристроях із рухомою комою усі числа зберігаються у нормалізованому вигляді, при цьому не втрачаються молодші розряди мантиси і підвищується точність обчислень. Якщо після виконання якої-небудь арифметичної операції результат виявляється ненормалізованим, то перед записом числа в пам'ять виконується його нормалізація, тобто зсув мантиси ліворуч на відповідну кількість розрядів, і зменшення порядку числа на відповідну кількість одиниць.

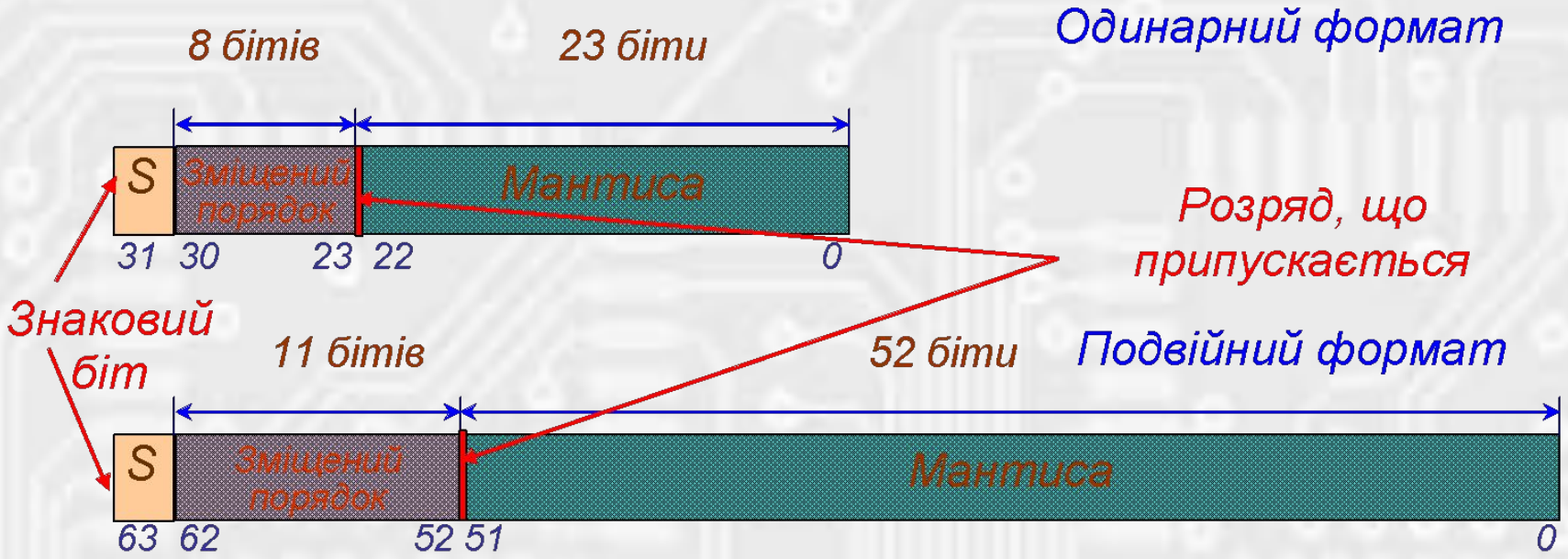
Показова форма має і свої вади, основною з яких є порівняно висока складність виконання арифметичних операцій та, відповідно, більша вимогливість до ресурсів обчислювального пристрою. Це обмежує її застосування, наприклад, у спеціалізованих радіотехнічних обчислювальних пристроях, у системах управління технологічними процесами та обробки вимірювальної інформації у реальному часі.

2. Формати представлення двійкових чисел

Якщо мантиса є нормалізованою, то її старший розряд завжди в форматі IEEE 754 дорівнює 1 (1/2≤M<1). Ця одиниця не відображається у форматі числа, тобто є уявною (розряд, що припускається).

Щоб порядок завжди був додатним числом (при цьому непотрібно зберігати знаковий розряд порядку), використовують зміщений порядок $P_{зм} = P + P_{зсуву}$, де $P_{зсуву} = 2^{k-1}-1$ визначається числом двійкових розрядів ***k***, що відводяться під збереження порядку (-1 – для урахування старшого розряду мантиси, що припускається).

Основні формати чисел з рухомою комою в стандарті IEEE 754



2. Формати представлення двійкових чисел

Основні характеристики форматів представлення двійкових чисел (стандарт IEEE 754)

Параметр	Формат	
	Одинарний	Подвійний
Довжина слова, біт	32	64
Довжина порядку, біт	8	11
Зміщення порядку	127	1023
Діапазон чисел	10^{-38}, 10^{+38}	10^{-308}, 10^{+308}
Довжина мантиси, біт	23	52
Найменше нормалізоване число	$2^{-126}=10^{-38}$	$2^{-1022}=10^{-308}$
Наближена точність представлення чисел	$2^{-23}=10^{-7}$	$2^{-52}=10^{-15}$

Одинарний формат містить знаковий біт *S*, 8-бітовий зміщений порядок *E* та 23-бітову дріб *M*. Зсув дорівнює 127, використовується прихований біт цілої частини мантиси, а *Emin=0* та *Emax=255* зарезервовані для подання спеціальних чисел.

Приклад: подати число 45,75 в базовому одинарному форматі. 45,75=101101,110; *S*=0 (число додатне); *M*=**1**01101110000000000000000; Мантиса без прихованого біту: M=01101110000000000000000; порядок *p*=6; *E*=6+127-1=132=10000100.

***X*= 0100001000110111000000000000000 = 42370000h**

S ***E*** ***M***

2. Формати представлення двійкових чисел

Значення числа, записаного в одинарному форматі, обчислюється за формулою:

$$N = (-1)^S \cdot 2^{E-127} \cdot (1, M).$$

Приклад: Маємо число X= **0**10000100011011100000000000000000, подане в одинарному форматі. Записати його в десятковому вигляді.

- Знаковий розряд - 0, тому число додатне;
- Порядок: E=**$10000100_2 = 132_{10}$; 132-126=6**;
- Мантиса з урахуванням прихованого біту: M=**1**01101110000000000000000;
- Зсуваємо кому мантиси на **6** (фактично виконуємо операцію множення на $\mathbf{2^5}$) розрядів праворуч та відкидаємо всі нулі після останній одиниці: 101101,11;
- **$101101{,}11_2 = 45{,}75_{10}$.**

Для формату із рухомою комою характерно протиріччя:

Для збільшення діапазону подання чисел потрібно збільшувати розрядність порядку, але при цьому (при постійної довжині формату) буде зменшуватися точність надання чисел, так як скорочується розрядність мантиси. Для збільшення і діапазону і точності подання чисел потрібно збільшувати довжину всього формату.

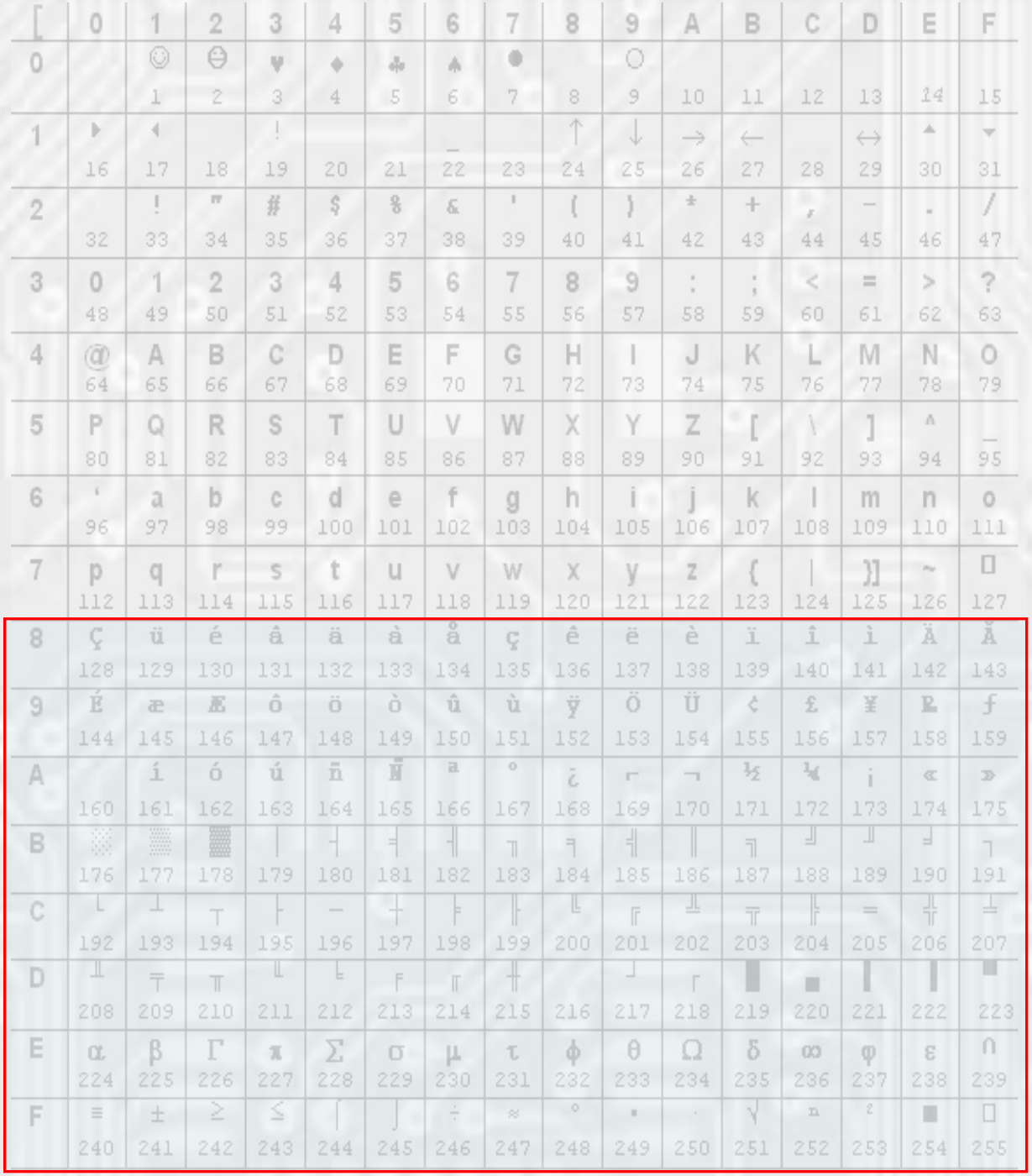
3. Кодування символьної та логічної інформації

Американський стандартний код інформаційного обміну ASCII

До символьної інформації відносяться букви, цифри, спеціальні знаки. Кожному символу ставиться в відповідність двійковий однобайтовий код. Він визначає коди для 32 символів керування, 10 цифр, 52 букв (великі та маленькі літери англійського алфавіту), 32 спеціальних символів, а також для символу інтервалу. Старший 8-й біт було введено для забезпечення перевірки на парність. Цій біт дозволяє виявляти однократні помилки при передачі даних.

Приклад: Визначити символ, який має код 01011010_2. Старший біт – контрольний, не впливає на значення коду. $101_2 = 5_{16}$; $1010_2 = A_{16}$. Символ з кодом $5A_{16}$ – ***Z***.

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2.		!	"	#	$	%	&	'	(	)	*	+	,	—	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL



3. Кодування символьної та логічної інформації

З підвищенням надійності комп'ютерів важливість біта парності знизилась, тому на початку 80-х років його стали використовувати для розширення набору кодованих символів в межах від 128_{10} до 255_{10} (наприклад, для кодування математичних символів або символів інших мов, а також символів псевдографіки (для оформлення рамок та таблиць).

Кодова таблиця, запропонована IBM.

3. Кодування символьної та логічної інформації

Варіанти кодування кирилиці:

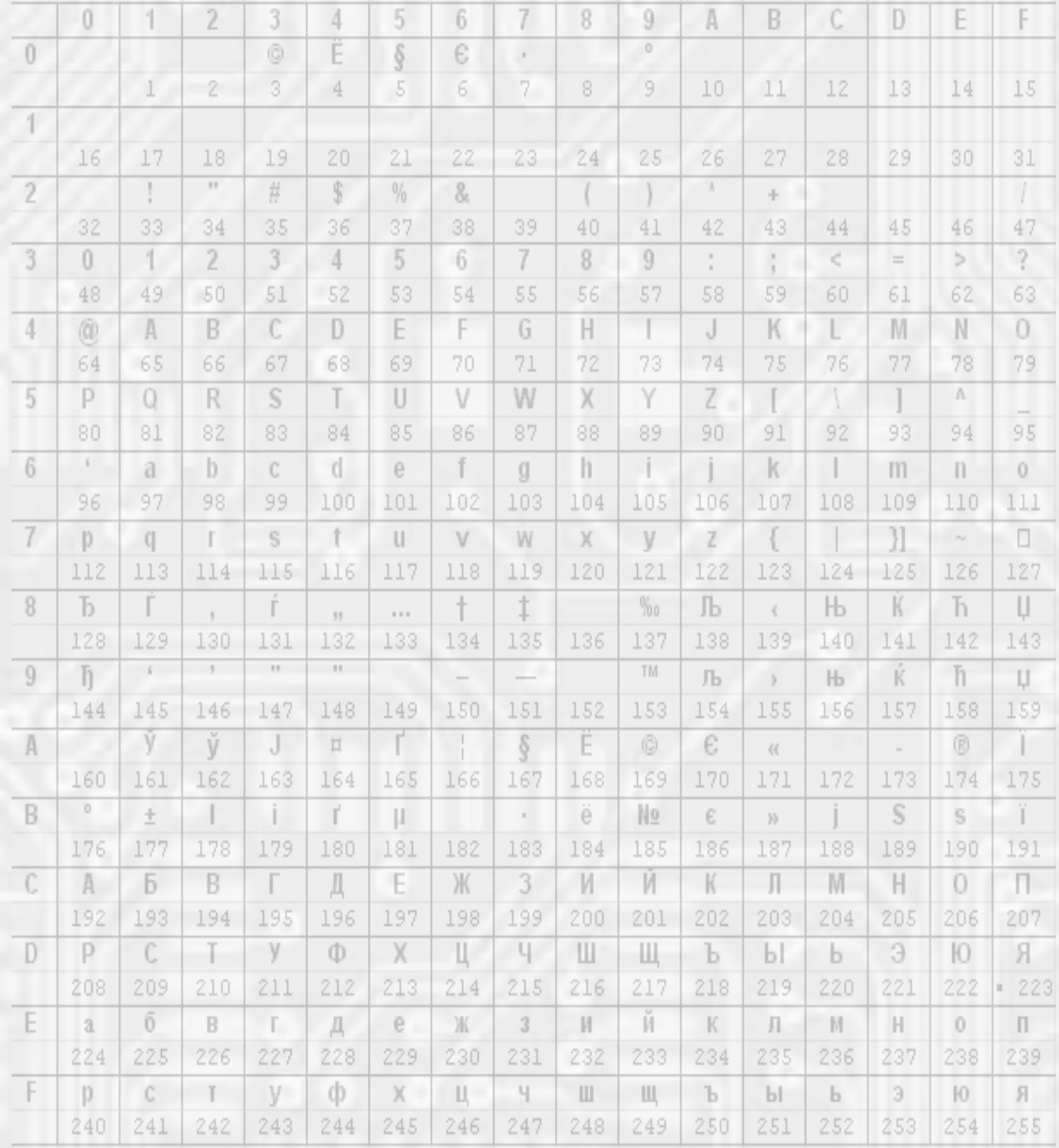
- альтернативна кодова таблиця CP-866;
- міжнародний стандарт ISO 8859;
- кодова таблиця фірми Microsoft CP-1251 (кодування Windows);
- кодова таблиця для в ОС Unix KOI 8-r.

8	А 128	Б 129	В 130	Г 131	Д 132	Е 133	Ж 134	З 135	И 136	Й 137	К 138	Л 139	М 140	Н 141	О 142	П 143
9	Р 144	С 145	Т 146	У 147	Ф 148	Х 149	Ц 150	Ч 151	Ш 152	Щ 153	Ъ 154	Ы 155	Ь 156	Э 157	Ю 158	Я 159
A	а 160	б 161	в 162	г 163	д 164	е 165	ж 166	з 167	и 168	й 169	к 170	л 171	м 172	н 173	о 174	п 175
B	▒ 176	▒ 177	▓ 178	│ 179	┤ 180	╡ 181	╢ 182	╖ 183	╕ 184	╣ 185	║ 186	╗ 187	╝ 188	╜ 189	╛ 190	┐ 191
C	└ 192	┴ 193	┬ 194	├ 195	─ 196	┼ 197	╞ 198	╟ 199	╚ 200	╔ 201	╩ 202	╦ 203	╠ 204	═ 205	╬ 206	╧ 207
D	╨ 208	╤ 209	╥ 210	╙ 211	╘ 212	╒ 213	╓ 214	╫ 215	╪ 216	┘ 217	┌ 218	▉ 219	▄ 220	▌ 221	▐ 222	▀ 223
E	р 224	с 225	т 226	у 227	ф 228	х 229	ц 230	ч 231	ш 232	щ 233	ъ 234	ы 235	ь 236	э 237	ю 238	я 239
F	Ё 240	ё 241	≥ 242	≤ 243	⌠ 244	⌡ 245	÷ 246	≈ 247	° 248	∙ 249	· 250	√ 251	ⁿ 252	² 253	■ 254	□ 255

Альтернативна кодова таблиця CP-866

3. Кодування символьної та логічної інформації

Кодова таблиця фірми Microsoft CP-1251



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				©	Ё	§	Є	·		°						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2		!	"	#	$	%	&		(	)	'	+				/
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}]	~	□
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	Ђ	Г	,	ѓ	„	…	†	‡		‰	Љ	‹	Њ	Ќ	Ћ	Џ
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	ђ	'	'	"	"		–	—		™	љ	›	њ	ќ	ћ	џ
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A		Ў	ў	J	¤	Ґ	¦	§	Ё	©	Є	«		-	®	Ї
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	°	±	І	і	ґ	µ		·	ё	№	є	»	j	S	s	ї
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Наприклад, в альтернативній таблиці CP-866 велика літера А має код 128, а в кодуванні Windows - 192

3. Кодування символьної та логічної інформації

Стандарт кодування символів Unicode

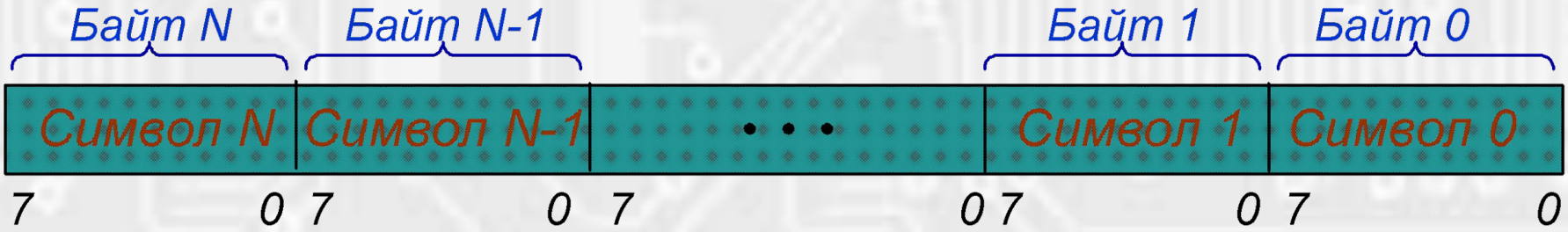
Unicode – це 16-розрядний алфавіт, сумісний з ASCII та погоджений з міжнародним алфавітом ISO/IEC 10646-1. Оскільки 16-ма розрядами можна закодувати 64К символів, цього достатньо для кодування всіх букв алфавітів народів світу.

Тип символу	Опис набору символів	Кількість символів	Шістнадцяткові значення символів
Алфавіти	Латинський, кирилиця, грецький і т.д.	8192	Від 0000 до 1FFF
Символи	Графічні мітки, математичні символи і т.д.	4096	Від 2000 до 2FFF
CJK	Китайські, японські і корейські символи	4096	Від 3000 до 3FFF
Han	Уніфіковані китайські, японські і корейські символи	40960	Від 4000 до 4FFF
	Розширення чи надлишок від Han	4096	Від E000 до EFFF
Вказані користувачем		4096	Від F000 до FFFF

3. Кодування символьної та логічної інформації

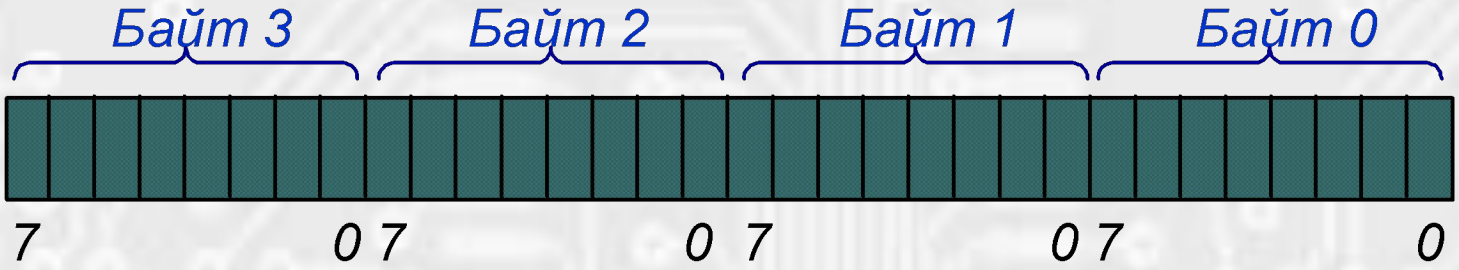
Кодування символьних рядків

Символьна інформація об'єднується в рядки (символьні або текстові). Максимальна довжина рядку складає 2^{32} байт = 4 Гбайти.



Кодування бітових полів

Бітові поля призначені для запису логічних змінних. Вони мають довжину від 1 до 32 бітів. Кожний розряд формату призначений для подання однієї логічної змінної.



4. Кодування команд в комп'ютері

Програма – це алгоритм, записаний у вигляді послідовності команд.

Команда – це двійковій код, який вказує:

- операцію, яку треба виконати;
- дані, що використовуються в цій операції;
- інформацію про адресу, за якою необхідно зберегти результат операції;
- адресу наступної команди.

Код команди складається з двох частини (полів): ***коду операції*** та ***адресної (операндної) частини***. ***Структура команди*** визначає склад, призначення і розміщення полів в команді.

Загальна структура команди

Операційна частина містить код операції ***(КОп),*** який визначає: тип операції (запис, читання, додавання, множення, пересилання інформації та інші); тип операндів; формат операндів; довжину коду команди; спосіб адресації операндів та інколи іншу інформацію.

Адресна (операндна) частина команди містить інформацію про адреси операндів (або самі операнди) і результату операції, а в деяких випадках, інформацію про адресу наступної команди.

4. Кодування команд в комп'ютері

Формат команди - це порядок розміщення її структури в розрядної сітці комп'ютера з розміткою номерів розрядів, що визначають межу окремих полів команди, або з вказівкою кількості бітів в полях команди.

Структура і формат команди визначають її довжину. Довжина команди має бути узгодженою з довжиною машинного слова, що прийнята для даної архітектури комп'ютера. Це дозволяє використовувати одні і ті ж самі апаратні засоби (у першу чергу пам'ять) для передачі, зберігання і перетворення як операндів, так і команд. Крім того, вона має бути найменшою (чим коротше команди, тим менше потрібно місця в пам'яті для зберігання програм).

Довжина операційної частини команди при двійковому позиційному кодуванні КОп: $n_{о.ч.} \geq \log_2 M$. Як правило, для більшості сучасних процесорів кількість операцій $M > 200$, тому довжина операційної частини команд складає не менше одного байта.

Структура команд

Код операції	*Перша адреса* A_1	*Друга адреса* A_2	*Третя адреса* A_3	*Четверта адреса* A_4

4. Кодування команд в комп'ютері

Більшу частину формата команди займають адресні поля, тому що у загальному випадку команда повинна мати чотири адреси:

- двох адрес вхідних операндів;
- адреси розміщення результату;
- адреси наступної команди.

Спосіб кодування команд – це правило опису двійковим кодом операційної та адресної частин команд таким чином, щоб цей код дозволяв однозначно визначати операцію, яку потрібно виконати, та операнди, над якими цю операцію потрібно виконувати.

Для скорочення кількості адресних полів використовується ***адресація, що припускається.***

1. Припускається, що після виконання команди, розміщеної за адресою A, що має довжину *L* байтів, виконується команда з комірки за адресою (A + *L*). ***Такий порядок вибірки команд називається натуральним.***

2. Припускається, що результат операції завжди розміщується на місці одного з операндів, що приймають участь в операції, частіше − першого. Тоді адреса розміщення результату задається неявно.

4. Кодування команд в комп'ютері

В зв’язку з вищесказаним в комп’ютерах найчастіше використовуються двоадресні, одноадресні та, інколи, безадресні команди:

В ***двоадресній команді*** вказуються адреси двох вхідних операндів, над якими виконується операція, а результат записується на місце одного з вхідних операндів, звичайно − першого.

В ***одноадресній команді*** не тільки адресу результату, але і адресу одного з вхідних операндів припускають і в явному вигляді в команді не зазначають. При цьому адреса одного операнда зазначається в адресному полі команди, а другий операнд розміщується в спеціальному регістрі процесору, який називається акумулятором. Результат операції записується в цей же регістр (***акумуляторна архітектура процесора***).

В ***безадресних командах*** припускаються адреси обох операндів а також адреса результату операції. Приклад – робота зі стековою пам’яттю (***стекова архітектура процесора***).

5. Класифікація архітектур комп'ютера за складом системи команд

Одна з основних задач розвитку архітектури процесорів – усунення проблеми семантичного розриву між складними операторами мов програмування високого рівня та достатньо простими машинними операціями, що виконуються в процесорі, при постійному рості вимог до продуктивності.

За ознакою архітектури системи команд розрізняють:

- система з повним набором команд: CISC (Complex Instruction Set Computer) – велика кількість порівняно складних команд, що мають різні довжину та час виконання. Забезпечує зручність та гнучкість програмування, але неефективно використовує ресурси комп'ютера;
- система з скороченим набором команд: RISC (Reduced Instruction Set Computer) – невелика кількість простих команд, що мають переважно однакову довжину та час виконання; програмування ускладнено, але ресурси комп'ютера використовуються більш ефективно.
- система з командними словами надвеликої довжини: VLIW (Very Long Instruction Word) – RISC- команди об'єднуються у «зв'язки» фіксованої довжини з метою їх паралельної передачі та виконання.

5. Класифікація архітектур комп'ютера за складом системи команд

Характеристика	CISC	RISC	VLIW
Довжина команд	Варіюється	Однакова	
Розташування полів в команді	Варіюється	Незмінне	
Кількість регістрів	Декілька (часто спеціалізованих)	Багато регістрів загального призначення (переважно рівноправних)	
Доступ до пам'яті	Може виконуватися як частина команд різних типів	Виконується тільки спеціальними командами	

Література

1. Мельник А.О. Архітектура комп'ютера. Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – с.56 – 118.