

Specyfikacja argumentów (danych)

Reprezentacja **zewnętrzna** (języki programowania):

- format łatwo rozpoznawalny przez człowieka:
 - zapis pozycyjny dziesiętny ze znakiem
 - zapis naukowy/inżynierski
 - zapis znakowy (tekst)

Reprezentacja **wewnętrzna** (kod maszynowy):

- uporządkowany ciąg bitów (ciąg zero-jedynkowy)
- **słowo maszynowe** – uporządkowany ciąg bitów o **długości** 2^k bitów (8,16,32,...)

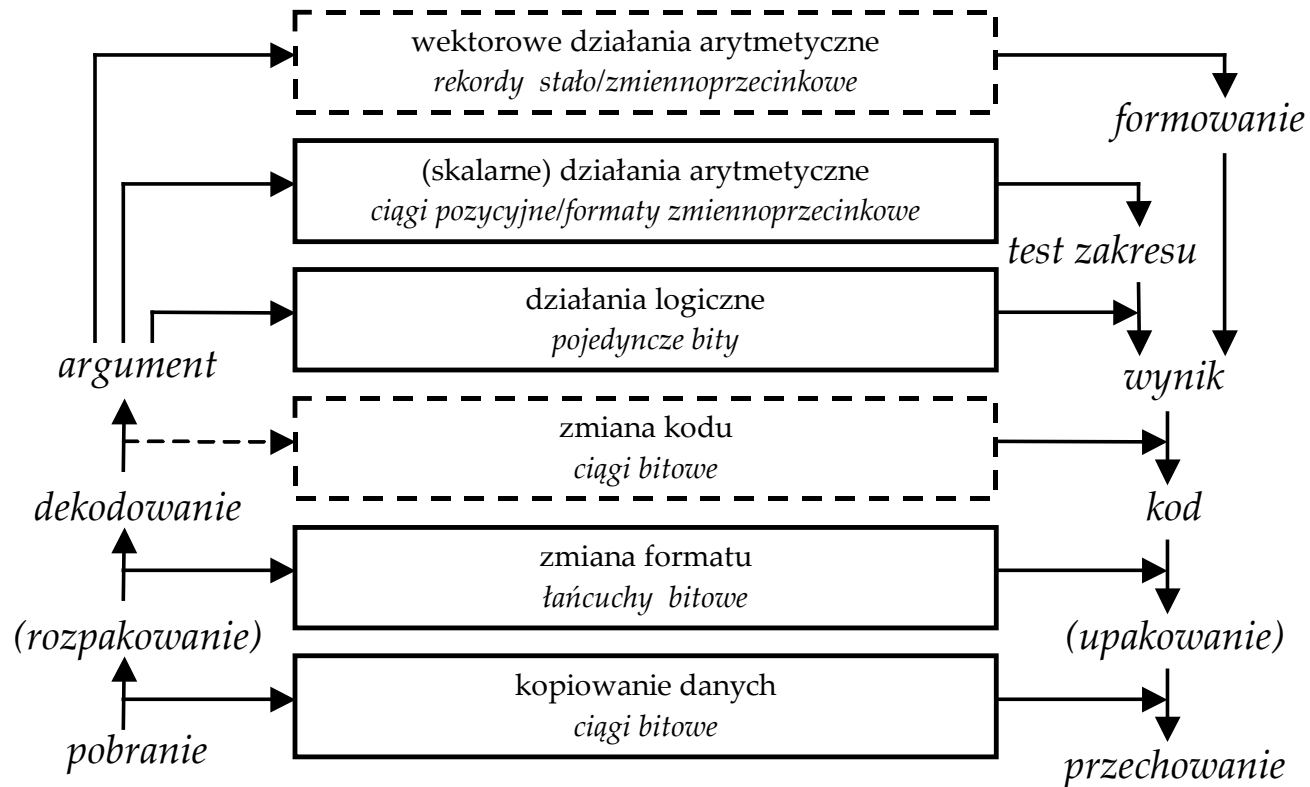
Interpretacja ciągu bitów zależy od:

- specyfikacji (sposobu wykonania) działania elementarnego
- definicji funkcji

Deklaracja zmiennej:

- nie narzuca interpretacji
- określa rozmiar zmiennej w przestrzeni logicznej/ pamięci ope
- ułatwia kontrolę poprawności wyników działań

Argumenty działań



Argumenty działań

Cechy działań (1)

kopiowanie

- niszczące, nieodwracalne
- kopiowanie bloków – ryzyko zniszczenia źródła (kolejność przesłań)
- wymiana (*exchange*) (także przestawianie (*swap*)) – odwracalne

zmiana formatu

- przemieszczenie pól (rekordów) – przestawienie (*swap*)
- systematyczne przemieszczenie bitów
 - zwykłe – przesunięcie arytmetyczne lub logiczne
 - cykliczne – rotacja prosta i rozszerzona
- rozszerzanie kodów liczb
- rozpakowanie i upakowanie kodu
- konwersje formatów zmiennoprzecinkowych

(zmiana kodu)

- konwersje typów liczb (zmiennoprzecinkowy \leftrightarrow stałoprzecinkowy)
- przekodowanie przez tablicę

Cechy działań (2)

logiczne

- równoległe na parach bitów
- nieodwracalne – AND / OR albo ANDN (iloczyn z zanegowanym operandem)
- odwracalne – XOR

arytmetyczne

- syndromy wyniku (kody warunkowe)
- stałoprzecinkowe
 - odwracalne
 - mnożenie pełne lub częściowe
 - dzielenie pełne lub skrócone
- zmiennoprzecinkowe
 - nieodwracalne (zaokrąglanie, brak łączności dodawania)
 - obsługa kodów specjalnych (nie-liczby i nieskończoności)

wektorowe

- wyniki formowane w rekordach, zawsze w zakresie (nasycanie lub obcięcie)

Typy i formaty danych

– kody rozkazów (*op-codes*)

- regularny format i stały rozmiar (architektura RISC)
- nieregularny format i zmienny rozmiar (architektura CISC)

– dane użytkowe (*user data*)

- ciągi bajtów o **znaczeniu interpretowanym** przez procesor
- **umowy i standardy**
 - kody znaków (ISO-5 → ISO-7 → ASCII)
 - liczby naturalne (kod NB) i całkowite (kod U2)
 - liczby pseudorzeczywiste (standard IEEE 754-2008)

– dane systemowe (*system data*)

- dynamiczne – tworzone podczas wywołania funkcji lub procedury
- statyczne tworzone podczas kreacji procesów (zadań)

Kody rozkazów (RISC)

kody rozkazów – dane użytkowe – dane systemowe

- a)

OP	RD	RA	DISP / IMM		
----	----	----	------------	--	--
- b)

OP	RD	RA	RB	OP-EXT	
----	----	----	----	--------	--
- c)

OP	RD	RA	RB	RC	OP-EXT	
----	----	----	----	----	--------	--
- d)

OP	ADDR				
----	------	--	--	--	--
- e)

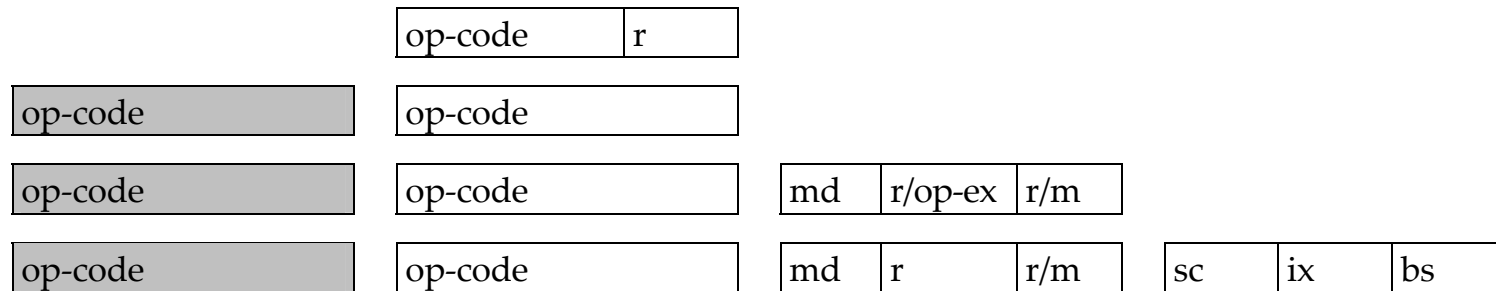
OP	BO	BI	ADDR		
----	----	----	------	--	--
- f)

OP	BO	BI	OP-EXT		
----	----	----	--------	--	--

Jednolita struktura słów kodu maszynowego procesora RISC (architektura L/S)

- a) transfery (*load/store*), b,c) działania arytmetyczne i logiczne,
d) skok bezwarunkowy, e,f) rozgałęzienia warunkowe

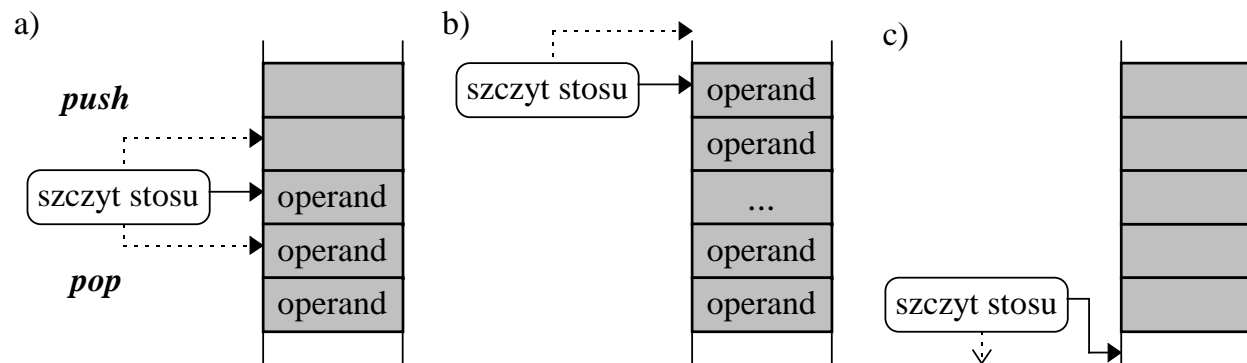
Zmienna struktura kodu rozkazów (CISC)



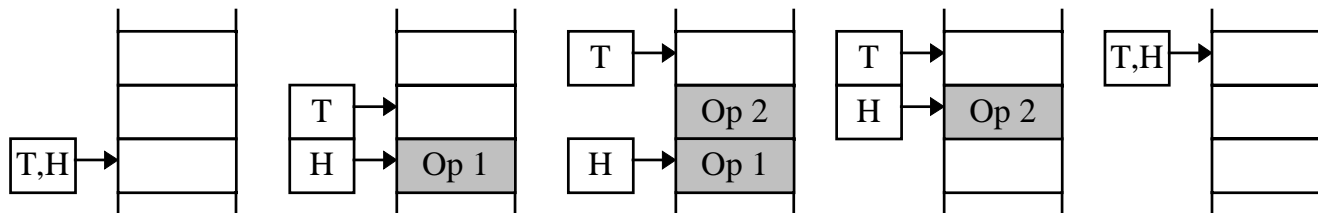
Struktura kodu procesorów Intel x86

- (prefiks blokady magistrali **lock**)
- przedrostek rozmiaru adresu (*default address size*) [80386+]
- przedrostek rozmiaru operandu (*default operand size*) [80386+]
- przedrostek zmiany segmentu (*segment override prefix*) | (lub **rep**)
- **kod rozkazu** (*opcode*)
- **rozszerzenie kodu** (*opcode extension*)
- **bajt trybu adresowania** (*addressing mode byte*)
- **bajt rozszerzenia adresu** (*address extension*) [80386+]
- bajty przemieszczenia (*displacement*) – 0, 1, 2 lub 4 [80386+]
- bajty argumentu bezpośredniego (*immediate data*) – 0, 1, 2 lub 4 [80386+]

Dane systemowe

kody rozkazów – dane użytkowe – dane systemowe

Organizacja stosu: a) działania, b) przepełnienie, c) wyczerpanie



Organizacja kolejki (T – wskaźnik końca, H – wskaźnik czoła)

Dane użytkowe - interpretacje

kody rozkazów – dane użytkowe – dane systemowe

Typy *skalarne*:

- identyfikatory (*enumeration*) – kody informacji nieliczbowej, cechy (*attribute*)
 - logiczne (*boolean*), znakowe (*character*), opisowe (*descriptive*);
- dyskretne (*discrete*), w szczególności:
 - całkowite (*integer*) i porządkowe (*cardinals*), inaczej naturalne (*natural*),
- pseudo-rzeczywiste (*non-discrete*), w szczególności:
 - stałoprzecinkowe (*fixed-point*) – interpretowane całkowite,
 - zmiennoprzecinkowe (*floating-point*).

Typy *strukturalne* – zbiory danych skalarnych lub strukturalnych i obejmują:

- zestawy (*sets*) – nieuporządkowane zbiory danych,
- wektory (*vectors*) i tablice (*arrays*) – uporządkowane zbiory danych,
 - łańcuchy (*strings*), – uporządkowane ciągi (wektory) znaków,
- rekordy (*records*), – regularne struktury danych dowolnych typów.

Typy *wskaźnikowe* (*access*) – identyfikują lokalizację (adres) obiektu

Kod ASCII (American Standard Code for Information Interchange)

Kod ASCII (część międzynarodowa) = 0 || ISO-7 (CCITT No 5)

H L	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0010	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	—
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

NUL <i>nullify</i>	SOH – <i>start of header</i>	STX – <i>start of text</i>	ETX – <i>end of text</i>
EOT – <i>end of transfer</i>	ENQ – <i>enquire</i>	ACK – <i>acknowledge</i>	BEL – <i>bell</i>
BS – <i>backspace</i>	HT – <i>horizontal tab</i>	LF – <i>line feed</i>	VT – <i>vertical tab</i>
FF – <i>form feed</i>	CR – <i>carriage return</i>	SO / SI – <i>shift out i in</i>	DLE – <i>data link ESC</i>
DC1,...4 – <i>data control</i>	NAK – <i>negative ACK</i>	SYN – <i>synchronize</i>	ETB – <i>end of text block</i>
CAN – <i>cancel</i>	EM – <i>end of medium</i>	SUB – <i>substitute</i>	ESC – <i>escape</i>
FS – <i>file separator</i>	GS – <i>group separator</i>	RS – <i>record separator</i>	US – <i>unit separator</i>

UNICODE – kod 16/32-bitowy, obejmujący znaki diaktryczne większości języków

Kod ASCII – regularności

Konwencje assemblerowe:

'znak' – kod ASCII (1bajt) znaku alfanumerycznego (litery, cyfry, +, -, =,...)

"tekst" – ciąg kodów ASCII kolejnych znaków tekstu (konwencja BE)

Zapis znaków specjalnych w tekście ciągłym – $\backslash \spadesuit$ # znak specjalny \spadesuit (np. $\backslash \backslash$, $\backslash "$, $\backslash ?$), w szczególności

$\backslash ddd$ # kod ósemkowy ddd	$\backslash xDD$ # kod szesnastkowy DD
$\backslash n$ # ($\backslash x0A = \backslash 012$) LF, NL, new line	$\backslash 0$ # ($\backslash x00 = \backslash 000$) NUL, koniec rekordu
$\backslash b$ # ($\backslash x08 = \backslash 010$) BS, backspace	$\backslash t$ # ($\backslash x09 = \backslash 011$) HT, TAB, tabulation
$\backslash f$ # ($\backslash x0C = \backslash 014$) FF, form feed	$\backslash r$ # ($\backslash x0D = \backslash 015$) CR, carriage return

kody cyfr dziesiętnych ($Z_{16} = bbbb_2$ – wartość cyfry):

$3\#_{16}$ – w notacji szesnastkowej ('7' = $0x37$)

0011 bbbb – w notacji dwójkowej ('3' = $0b0011\ 0011$)

wartość cyfry X: 'X' – '0' lub 'X' – $0x30_{16}$ lub 'X' AND $0x0F_{16}$

kody liter (bbbbbb – 5-bitowy nr litery w porządku alfabetu łacińskiego):

dużych: 010 bbbbb ('A' = $0b010\ 00001 = 0x41 = 41_{16}$),

małych: 011 bbbbb ('z' = $0b011\ 11010 = 0x7A = 7A_{16}$)

nr litery w porządku alfabetu: – 'X' AND $0x1F_{16}$

zamiana „duża” \leftrightarrow „mała”: – 'X' XOR $0x20_{16}$

Reprezentacje liczb

Jednostka stałoprzecinkowa (*Integer Unit*)

łańcuch bitów odzwierciedlający zapis pozycyjny lub pokrewne

- dwoista interpretacja (IA-32, Motorola 68K): naturalna lub uzupełnieniowa
 - weryfikacja wyniku na podstawie tworzonych kodów warunkowych
- interpretacja przypisana działaniu (PowerPC)
 - weryfikacja wyniku przypisana interpretacji (\geq / \leq /nadmiar)

Jednostka zmiennoprzecinkowa (*Floating-Point Unit*)

złożenie łańcuchów bitów reprezentujących liczbę

- reprezentacje standardowe (IEEE 754-2008)
 - format wykładniczy liczby: znak – wykładnik – mnożnik (mantysa)
 - kody specjalne: nie-liczby (*NaN*) i nieskończoności
 - zaokrąglenia
 - wyjątki
- reprezentacje niestandardowe
 - specyficzne dla architektury (DSP)

System dwójkowy

kod naturalny dwójkowy (NB, *natural binary*)

...	$\times 2^{m-1}$	$\times 2^{m-2}$...	$\times 2^0$	$\times 2^{-1}$...	$\times 2^{-p-1}$	$\times 2^{-p}$...
...	b_{m-1}	b_{m-2}	...	b_0	b_{-1}	...	b_{-p-1}	b_{-p}	...

$$X = \sum b_i 2^i$$

notacja szesnastkowa (b_j – bity, d_i – cyfry reprezentacji szesnastkowej)

$$\sum d_i \cdot 16^i = \sum (b_{4i+3} 2^3 + b_{4i+2} 2^2 + b_{4i+1} 2^1 + b_{4i} 2^0) \cdot 2^{4i}$$

...	d_1				d_0				d_{-1}				...
...	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	b_{-1}	b_{-2}	b_{-3}	b_{-4}	...

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Kodowanie liczb dziesiętnych

- zapis pozycyjny – $N = \sum d_i \beta^i$
- binarne zakodowanie jednej cyfry wymaga $\lceil \log_2 \beta \rceil$ bitów
 → potrzebne 4 bity (tetradą) na każdą cyfrę dziesiętną
 → nadmiar w przestrzeni kodowej → wiele sposobów kodowania

Kod BCD (*Binary Coded Decimal*)

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	1	2	3	4	5	6	7	8	9	–	–	–	–	–	–

Kod BCD+3 i jego dopełnienie ($9-d$)

–	–	–	0	1	2	3	4	5	6	7	8	9	–	–	–
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000
–	–	–	9–0	9–1	9–2	9–3	9–4	9–5	9–6	9–7	9–8	9–9	–	–	–

Liczby stałoprzecinkowe

zapis pozycyjny / uzupełnieniowy

skalowanie: liczba stałoprzecinkowa = liczba całkowita $\times b^s$

kodowanie arytmetyczne (następna: +1, poprzednia: -1):

- uzupełnianie – liczba ujemna = 0 – liczba przeciwna (dodatnia)
- polaryzacja – wartość = wartość naturalna – stała (tylko liczby całkowite)

kodowanie umowne (wyjątkowo):

- znak-moduł – „znak” | wartość bezwzględna liczby
- dopełnianie – liczba ujemna = dopełnienie cyfr liczby przeciwnej dodatniej

właściwości

- **uzupełnianie** – łatwa arytmetyka (pozycyjna), porównanie i skalowanie
- **polaryzacja** – łatwe porównanie, dodawanie i odejmowanie, bez skalowania
- **znak-moduł** – skomplikowane dodawanie, odejmowanie i skalowanie
- **dopełnianie** – skomplikowana arytmetyka, porównanie i skalowanie

Reprezentacja uzupełnieniowa

Reprezentacją liczby przeciwnej do danej jest

wynik jej pozycyjnego odejmowania od 0: $-X \approx 0-X$

Dwójkowy kod uzupełnieniowy

wartość liczby reprezentowanej przez $\{x_{k-1}x_{k-2}...x_1x_0\}$ to

$$-x_{k-1}2^{k-1} + \sum_{i=0}^{k-2} x_i 2^i = -2^{k-1} + (1-x_{k-1})2^{k-1} + \sum_{i=0}^{k-2} x_i 2^i$$

Reprezentacja spolaryzowana dodatnio

$$X = \sum_{i=0}^{k-1} b_i 2^i - (2^{k-1} - 1)$$

- *porządek liczb zgodny z porządkiem kodów*
- łatwa konwersja na kod U2 i odwrotnie:

$$\left| \{x_{k-1}, x_{k-2}, \dots, x_0\}_{2^{k-1}-1} \right| = - \left| \{x_{k-1}, (1-x_{k-2}), \dots, (1-x_0)\}_{U2} \right|$$

Reprezentacje zmiennoprzecinkowe

notacja wykładnicza (naukowa / inżynierska):

$$\text{wartość} = \pm \text{mnożnik} \times \text{podstawa}^{\text{wykładnik}}$$

w notacji odręcznej najczęściej:

- podstawa = 10
- mnożnik ma 1 cyfrę części całkowitej i dowolną liczbę pozycji ułamka

w komputerach

- podstawa = 2
- spełnienie postulatu:
 - porządek kodów zgodny z porządkiem liczb naturalnych*
 - kodowanie wykładnika w kodzie z obciążeniem „+N”
 - mnożnik ze znakiem w kodzie „znak-moduł”
- mnożnik (mantysa) postaci $\pm(1+f)$, gdzie f jest dodatnim ułamkiem dwójkowym, zapewnia unikatowe kodowanie każdej reprezentowalnej liczby

Standard zmiennoprzecinkowy IEEE 754-2008 (1)

Standard IEEE754-2008

– kod wykładnika k -bitowy „ $+2^{k-1}-1$ ”, zakres wykładnika:

$$E_{\min} = -2^{k-1} + 2 \text{ (kod } 0 \dots 001), \quad E_{\max} = 2^{k-1} - 1 \text{ (kod } 1 \dots 110)$$

– normalizacja – unikatowe kodowanie liczby

liczba znormalizowana (ukryty bit „1”)

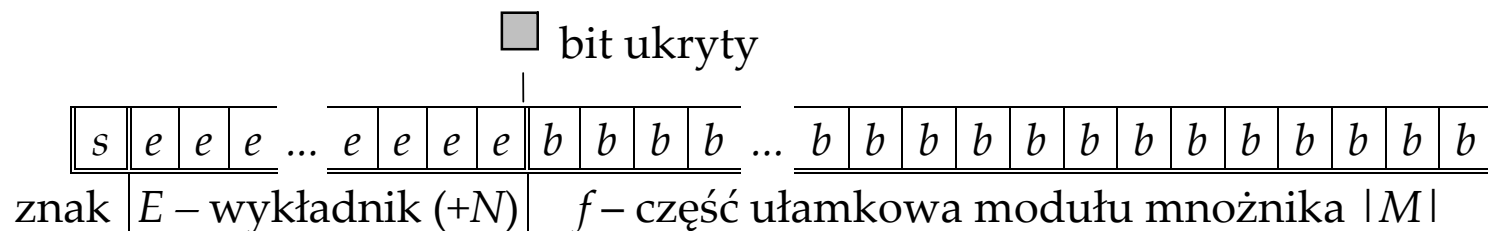
$$F = (-1)^s 2^E (1 + f), \quad 0 \leq f < 1$$

gdzie $f = 0, b_1 b_2 b_3 \dots b_m$ jest m -bitowym ułamkiem dwójkowym

nie istnieje znormalizowana reprezentacja zera !!

liczba zdenormalizowana (ukryty bit „0”) – także zero

$$F = (-1)^s 2^{E_{\min}} (0 + f), \quad 0 \leq f < 1$$



Standard zmiennoprzecinkowy IEEE 754-2008 (2)

Wzorce kodów obiektów binarnych standardu IEEE 754-2008

Wykładnik	Ułamek	Kod binarny	Wielkość
$E_{\min} \leq E \leq E_{\max}$	f	$s\ e\dots ee\ b\dots bb$	$F = (-1)^s 2^{E+1} (1 + f)$
$1\ 1\dots 1$	$f = 0$	$s\ 1\dots 11\ 0\dots 00$	$\pm \infty$
$1\ 1\dots 1$	$f \neq 0$	$s\ 1\dots 11\ b\dots bb$	NaN
$0\ 0\dots 0$	$f \neq 0$	$s\ 0\dots 00\ b\dots bb$	$\pm F_{\min} = (-1)^s 2^{E_{\min}} (0 + f)$
$0\ 0\dots 0$	$f = 0\dots 01$	$s\ 0\dots 00\ 0\dots 01$	$\pm F_{\min} = (-1)^s 2^{E_{\min}} 2^{-m}$
$E = E_{\max}$	$f = 1\dots 11$	$s\ 1\dots 10\ 1\dots 11$	$\pm F_{\max} = (-1)^s 2^{E_{\max}+1} (1 - 2^{-m-1})$

formaty:

16b (SEMI) – $[s_{15} \mid \mid E_{14:10} \mid \mid f_{9:0}]$

32b (SINGLE) – $[s_{31} \mid \mid E_{30:23} \mid \mid f_{22:0}]$

64b (DOUBLE) – $[s_{63} \mid \mid E_{62:52} \mid \mid f_{51:0}]$

128b (QUAD) – $[s_{127} \mid \mid E_{126:112} \mid \mid f_{111:0}]$

format rozszerzony

$n \times 32$ bity ($n \geq 8$), w tym co najmniej 16 bitów wykładnika

Wyjątki zmiennoprzecinkowe (IEEE 754-2008)

- dzielenie przez 0
- nadmiar wykładnika
- niedomiar wykładnika
- utrata dokładności (katastroficzna)
- niedozwolona operacja
- nie-liczba

Arytmetyka i zaokrąglanie

- do najbliższej (parzystej)
- do zera
- do nieskończoności

Bity ochronne: G (ang. *guard*) – "strażnik",
R (ang. *round*) – "zaokrąglanie",
S (ang. *sticky*) – "lepki"

Wektory i rekordy

wektor – uporządkowany zestaw argumentów jednakowego typu interpretacji:

wektory stałoprzecinkowe

wektory zmiennoprzecinkowe

działania specyficzne

rekord – uporządkowany zestaw argumentów różnego typu / o różnej interpretacji

działania zdefiniowane określone osobno w poszczególnych polach