

Organizacja i architektura komputerów ¹

Wykład 5

Piotr Patronik

5 kwietnia 2016

¹(Prawie) dokładna kopia slajdów dr hab inż. J. Biernata

Typy danych

kody rozkazów – dane użytkowe – dane systemowe

- ▶ Dane w pamięci maszyny
- ▶ Interpretacja zależna od stanu maszyny (rozkażu)

Typy danych

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
 - a transfery (load/store)
 - b,c działania arytmetyczne i logiczne
 - d skok bezwarunkowy
 - e,f rozgałęzienia warunkowe
- ▶ (architektura rejestrowa L/S)

Zmienna struktura kodu rozkazów (CISC)

kody rozkazów – dane użytkowe – dane systemowe



- ▶ 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+]

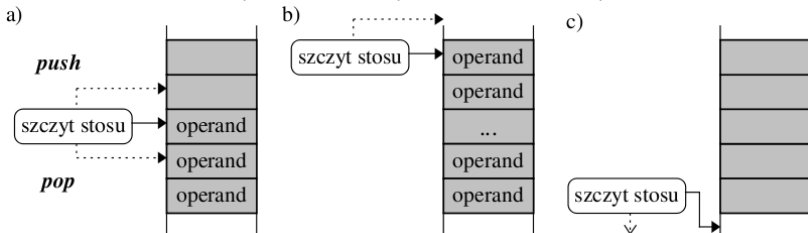
Mnemoniczny opis architektury listy rozkazów (ISA)

kody rozkazów – dane użytkowe – dane systemowe

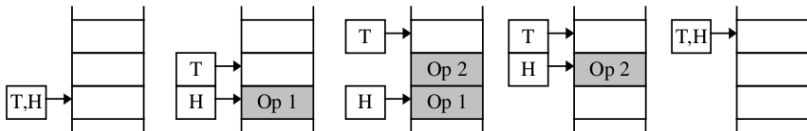
Mnemonic	Pełna nazwa	Rodzaj operacji	Typ
add	<i>add</i>	dodaj	A
sub	<i>subtract</i>	odejmij	A
mul, mpy	<i>multiply</i>	pomnóż	A
div	<i>divide</i>	podziel	A
cmp, cp	<i>compare</i>	porównaj (określ relację)	A
test	<i>test</i>	porównaj (sprawdź zgodność)	L
and	<i>and</i>	iloczyn logiczny	L
or	<i>or</i>	suma logiczna	L
xor	<i>exclusive-or</i>	suma wyłączająca (modulo 2)	L
inc / dec	<i>increment/decrement</i>	zwiększ / zmniejsz	K
shr / shl	<i>shift right/left</i>	przesuń w prawo / lewo	K
rr / rl	<i>rotate right/left</i>	przesuń cyklicznie w prawo / lewo	K
mov(e)	<i>move</i>	kopiuj (przenieść)	T
ld, load	<i>load</i>	pobierz (z pamięci) do rejestru	T
st, store	<i>store</i>	zapisz do pamięci (z rejestru)	T
bcc, jcc	<i>conditional branch</i>	rozgałęziaj (wybierz ścieżkę)	T
call, jsr	<i>call procedure</i>	wywołaj procedurę	T

Dane systemowe

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



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



- Bufory pierścieniowe
- Rekordy: deskryptory segmentów, stany procesora
- Tablice: wektorów przerwań, stron
- Tablice z haszowaniem: odwrócona tablica stron

Dane użytkowe

Typy skalarne

- ▶ jakościowe (*enumeration*) – kody informacji nieliczbowej, cechy (*attribute*)
- ▶ logiczne (*boolean*), znakowe (*character*)
- ▶ nieciągłe (*discrete*), w szczególności:
 - ▶ całkowite (*integer*) i porządkowe (*cardinals*), inaczej naturalne (*natural*),
- ▶ pseudo-ciągłe (*non-discrete*), w szczególności:
 - ▶ stałoprzecinkowe (*fixed-point*),
 - ▶ zmiennoprzecinkowe (*floating-point*).

Typy strukturalne – zbiory danych skalarnych lub strukturalnych

- ▶ 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 nullify

EOT – end of transfer

BS – backspace

FF – form feed

DC1,...4 – data control

CAN – cancel

FS – file separator

SOH – start of header

ENQ – enquire

HT – horizontal tab

CR – carriage return

NAK – negative ACK

EM – end of medium

GS – group separator

STX – start of text

ACK – acknowledge

LF – line feed

SO / SI – shift out i in

SYN – synchronize

SUB – substitute

RS – record separator

ETX – end of text

BEL – bell

VT – vertical tab

DLE – data link ESC

ETB – end of text block

ESC – escape

US – unit separator

- ▶ UNICODE – kod 16-bitowy, znaki diakrytyczne większości języków

Kod UTF-8

- ▶ 1 bajt: 0xxx xxxx
- ▶ 2 bajty: 110x xxxx 10xx xxxx
- ▶ 3 bajty: 110x xxxx 10xx xxxx 10xx xxxx
- ▶ 4 bajty: 110x xxxx 10xx xxxx 10xx xxxx 10xx xxxx
- ▶ Bity kodują wartość znaku
- ▶ Dolne 128 znaków: ASCII

Reprezentacje liczb

Jednostka stałoprzecinkowa (*Integer Unit*)

- ▶ łańcuch bitów odzwierciedlający zapis pozycyjny lub pokrewne
 - ▶ dwoista interpretacja (x86/Pentium, Motorola 68K)
 - ▶ naturalna lub uzupełnieniowa (jednakowy schemat działania)
 - ▶ weryfikacja poprawności po wykonaniu działania
 - ▶ interpretacja przypisana działaniu (PowerPC)
 - ▶ weryfikacja poprawności przypisana interpretacji

Jednostka zmiennoprzecinkowa (*Floating-Point Unit*)

- ▶ złożenie łańcuchów bitów reprezentujących liczbę
- ▶ reprezentacje standardowe (IEEE 854)
- ▶ znak – wykładnik – mantysa (znacznik)
- ▶ nie-liczby i wyjątki
- ▶ zaokrąglania
- ▶ reprezentacje niestandardowe – wykraczające poza standard IEEE 854
- ▶ specyficzne dla architektury (DSP, IBM)

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_i – bity, d_i – cyfry reprezentacji szesnastkowej)

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

...	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 (tetrada) 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 całkowite i wymierne (pozycyjne)

skalowanie: liczba mieszana $\times b^S =$ liczba całkowita

kodowanie umowne

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

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

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

właściwości

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

Dwójkowa reprezentacja uzupełnieniowa

...rozszerzenie

[illegible]

Dziesiętna reprezentacja uzupełnieniowa

...rozszerzenie

0	...	0	0	0	0	5	0	0	0	0	0	0	0	
0	...	0	0	0	0	4	9	9	9	9	9	9	9	$+5 \cdot 10^7 - 1$
...
0	...	0	0	0	0	0	5	0	0	0	0	0	9	$+5 \cdot 10^6 + 2$
0	...	0	0	0	0	0	5	0	0	0	0	0	0	$+5 \cdot 10^6 + 1$
0	...	0	0	0	0	0	4	9	9	9	9	9	9	$+5 \cdot 10^6 - 1$
0	...	0	0	0	0	0	4	9	9	9	9	9	0	$+5 \cdot 10^6 - 2$
...
0	...	0	0	0	0	0	0	0	0	0	0	0	2	+2
0	...	0	0	0	0	0	0	0	0	0	0	0	1	+1
0	...	0	0	0	0	0	0	0	0	0	0	0	0	0
9	...	9	9	9	9	9	9	9	9	9	9	9	9	-1
9	...	9	9	9	9	9	9	9	9	9	9	9	8	-2
...
9	...	9	9	9	9	9	5	0	0	0	0	0	2	$-5 \cdot 10^6 + 2$
9	...	9	9	9	9	9	5	0	0	0	0	0	1	$-5 \cdot 10^6 + 1$
9	...	9	9	9	9	9	5	0	0	0	0	0	0	$-5 \cdot 10^6$
9	...	9	9	9	9	9	4	9	9	9	9	9	9	$-5 \cdot 10^6 - 1$
9	...	9	9	9	9	9	4	9	9	9	9	9	8	$-5 \cdot 10^6 - 2$
...
9	...	9	9	9	9	5	0	0	0	0	0	0	0	$-5 \cdot 10^7$

Modularna reprezentacja uzupełnieniowa

- ▶ $|-1|_{2^n} = 2^n - 1$

- ▶ $|-1|_A = A - 1$

Dwójkowy kod uzupełnieniowy pełny (U2)

$+2^{m-1}-1$	0	1	1	...	1	1	1	$\downarrow -1$
$+2^{m-1}-2$	0	1	1	...	1	1	0	
...	
$+2$	0	0	0	...	0	1	0	
$+1$	0	0	0	...	0	0	1	
0	0	0	0	...	0	0	0	
-1	1	1	1	...	1	1	1	$\uparrow +1$
-2	1	1	1	...	1	1	0	
...	
$-2^{m-1}+2$	1	0	0	...	0	1	0	
$-2^{m-1}+1$	1	0	0	...	0	0	1	
-2^{m-1}	1	0	0	...	0	0	0	

$$-x_{m-1}2^{m-1} + \sum_{i=0}^{m-2} x_i 2^i = -x_{m-1}2^{n+m-1} + \sum_{i=m-1}^{n+m-2} x_{m-1} 2^i + \sum_{i=0}^{m-2} x_i 2^i$$

\downarrow

$$|\{x_{m-1}, x_{m-2}, \dots, x_1, x_0, \dots, x_{-p}\}| = |\{x_{m-1}, x_{m-2}, \dots, x_1, x_0, \dots, x_{-p}, 0, 0, \dots\}|$$

Kod spolaryzowany

$N = 2^{m-1}$								$N = 2^{m-1} - 1$
$2^{m-1} - 1$	1	1	1	...	1	1	1	2^{m-1}
$2^{m-1} - 2$	1	1	1	...	1	1	0	$2^{m-1} - 1$
...
0	1	0	0	...	0	0	0	1
-1	0	1	1	...	1	1	1	0
...
$-2^{m-1} + 1$	0	0	0	...	0	0	1	$-2^{m-1} + 2$
-2^{m-1}	0	0	0	...	0	0	0	$-2^{m-1} + 1$

- ▶ *porządek liczb zgodny z porządkiem kodów*
- ▶ *dodawanie i odejmowanie wymaga korekcji*
- ▶ *łatwa konwersja na kod U2 i odwrotnie*

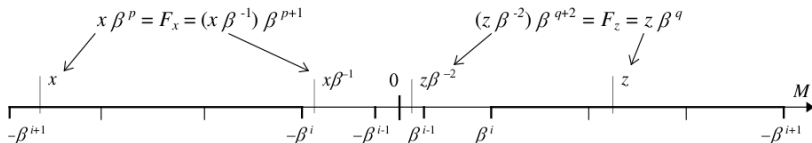
$$\begin{aligned}
 (p = 1) - & \left| \{x_{k-1}, x_{k-2}, \dots, x_0\}_{2^{k-1}-1} \right| = \left| \{x_{k-1}, (1 - x_{k-2}), \dots, (1 - x_0)\}_{U_2} \right| \\
 (p = 0) - & \left| \{x_{k-1}, x_{k-2}, \dots, x_0\}_{2^{k-1}-1} \right| = \left| \{(1 - x_{k-1}), x_{k-2}, \dots, x_0\}_{U_2} \right|
 \end{aligned}$$

Format stałoprzecinkowy

- ▶ $X = \sum_{i=-k}^m x_i 2^i$
- ▶ Przeskalowana liczba całkowita
- ▶ Reprezentacja zawsze przybliżona
 - ▶ kolejność i przemienność działań
 - ▶ dokładność
 - ▶ zaokrąglenia
- ▶ Implementacja w procesorach sygnałowych (np. TMS320)
- ▶ Ograniczony zakres części całkowitej i ułamka
 - ▶ specjalizowane zastosowania (DSP)
- ▶ Notacja naukowa: $X = A \cdot 10^B$

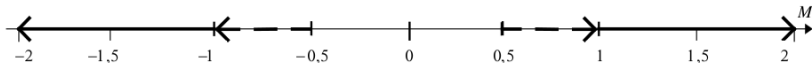
Formaty zmiennoprzecinkowe

- ▶ Notacja naukowa binarna $F = M\beta^E = M_i\beta^{E+i}$,
 $i = -m, \dots, -1, 0, 1, \dots, m$,
- ▶ Niejednoznaczna reprezentacja liczb zmiennoprzecinkowych



Propozycja: $\beta^{p-1} \leq |M| < \beta^p$

- ▶ Przedziały znormalizowanych wartości znacznika przy $\beta = 2$, $p = 1$ (—) i $p = 0$ (- - -)



- ▶ Gęstość reprezentacji

Formaty zmiennoprzecinkowe (2)

Jak zakodować znacznik?

► W kodzie U2

- warunek $2^{p-1} \leq |(-1)^s M| < 2^p$: 2 rozłączne warunki:
 $01, 00 \dots 00 \leq M \cdot 2^{-(p-1)} \leq 01, 11 \dots 11$, gdy $M > 0$
 $10, 00 \dots 01 \leq M \cdot 2^{-(p-1)} \leq 11, 00 \dots 00$, gdy $M < 0$
- warunek normalizacji trudny do sprawdzenia
- po złożeniu z dowolnym kodem wykładnika, porządek kodów liczb nie może być zgodny z porządkiem liczb (trudne porównanie liczb)

► W kodzie znak-moduł

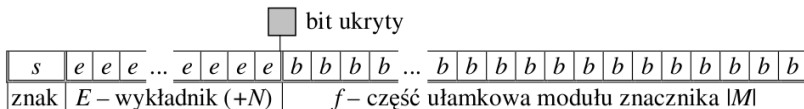
- warunek $2^{p-1} \leq |(-1)^s M| < 2^p$: upraszcza się do
 $2^{p-1} \leq M < 2^p$:
 $1, 00 \dots 00 \leq M \cdot 2^{-(p-1)} \leq 1, 11 \dots 11 \Rightarrow$
 $M \cdot 2^{-(p-1)} = 1, b_1 b_2 \dots b_m$
- warunek normalizacji łatwy do sprawdzenia
- po złożeniu z kodem wykładnika, porządek kodów zgodny z porządkiem bezwzględnych wartości liczb (łatwe porównanie)
- nie trzeba zapisywać wiodącej „1” („bit ukryty”)

Formaty zmiennoprzecinkowe (3)

Jak zakodować wykładnik?

- ▶ porządek kodów zgodny z porządkiem liczb całkowitych \rightarrow kod „ $+N$ ”
 - ▶ **liczba znormalizowana** (ukryty bit „1”)
 $F = (-1)^s 2^E (1 + f), 0 \leq f < 1$
 - ▶ brak reprezentacji zera
 - ▶ naturalna reprezentacja zera
kod postaci s **00...00 00...00**
 - ▶ **liczba zdenormalizowana** (ukryty bit „0”)
 $F = (-1)^s 2^E \min(0 + f), 0 \leq f < 1$
 - ▶ potrzebne są kody
 - ▶ $\pm\infty$
 - ▶ wyników, które nie są liczbami

Format zmiennoprzecinkowy IEEE 754/854



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

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

$$-2^7 + 2 = -126 \quad E \quad 127 = 2^7 - 1$$

$$-2^{11} + 2 = -1022 \quad E \quad 1023 = 2^{11} - 1$$

Wzorce kodów obiektów standardu IEEE 754

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