

Operacje zmiennoprzecinkowe

Liczbowe typy danych

Liczby całkowite ze znakiem

(C) KISI d.KIK PCz 2022 Programowanie niskopoziomowe 2

Typ BCD

spakowane 80 bitowe BCD

4 bity = 1 cyfra BCD

(C) KISI d.KIK PCz 2022 Programowanie niskopoziomowe 3

Liczbowe typy danych

Liczby zmiennoprzecinkowe

(C) KISI d.KIK PCz 2022 Programowanie niskopoziomowe 4

Liczby zmiennoprzecinkowe

- liczba = $(-1)^{\text{znak}} \cdot m \cdot 2^{\text{wykładnik}}$
- Liczba znormalizowana

$$\text{liczbaZ} = 1,010011010110 \cdot 2^{\text{wykładnik}}$$

↑
bit części całkowitej
- Liczba nieznormalizowana

$$\text{liczbaNZ} = 0,010110010110 \cdot 2^{\text{wykładnik}}$$

↑
bit części całkowitej

(C) KISI d.KIK PCz 2022 Programowanie niskopoziomowe 5

Liczby zmiennoprzecinkowe

Implementacja:

- liczba = $(-1)^{\text{znak}} \cdot m \cdot 2^{\text{wykładnik}}$
- wykładnik = przesunięta cecha - przesunięcie
- m = część całkowita + część ułamkowa = bit części całkowitej + mantysa
 - dla liczb pojedynczej i podwójnej precyzji część całkowita = 1 i nie jest zapamiętywana

Precyzja	Przesunięcie
pojedyncza	127
podwójna	1023
rozszerzona	16383

(C) KISI d.KIK PCz 2022 Programowanie niskopoziomowe 6

Wartości

poprawne i niepoprawne zawarte w rejestrach stosu koprocesora.

Klasa	Znak	Przesunięta cześć	Mantysa		
			Całość całkowita	Całość ułamkowa	
Dodatne liczby	Pasywne (czyste)	0	11.11	0	11.11
		0	11.11	0	10.00
	Aktywne (stbne)	0	11.11	0	01.11
		0	11.11	0	00.01
Dodatne zmiennoprzecinkowe	Nieskończoność	0	11.11	0	00.00
	Znormalizowane	0	11.10	1	11.11
		0	00.01	1	00.00
	Nieznormalizowane	0	11.10	0	11.11
		0	00.01	0	00.00
Pseudo-znormalizowane	0	00.00	1	11.11	
		0	00.00	0	00.00
Zero	0/1	00.00	0	00.00	
Ujemne zmiennoprzecinkowe	Pseudo-znormalizowane	1	00.00	1	11.11
		1	00.00	0	00.00
	Nieznormalizowane	1	11.10	0	11.01
		1	00.01	0	00.00
	Znormalizowane	1	11.10	1	11.01
		1	00.01	1	00.00
Minus nieskończoność	1	11.11	0	00.00	
Ujemne liczby	Aktywne (stbne)	1	11.11	0	01.11
		1	11.11	0	00.01
	Pasywne (czyste)	1	11.11	0	11.11
		1	11.11	0	10.00

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

7

Liczby zmiennoprzecinkowe

	precyzja		
	pojedyncza	podwójna	rozszerzona
cyfry znaczące	6	15	18
wartość największa	3,402823466E38	1,7976931348623158E308	1,189731495357231E4932
wartość najmniejsza	1,175494351E-38	2,2250738585072024E-308	3,3621031431120935E-4932

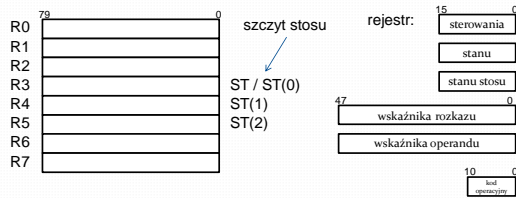
(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

8

Koprocesor - budowa

stos rejestrów



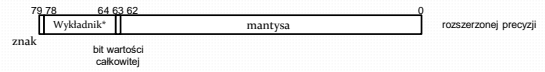
(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

9

Koprocesor

Rejestry R0, R1, R2, R3, R4, R5, R6, R7



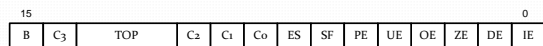
(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

10

Koprocesor

rejestr stanu



- B – koprocesor zajęty
- Co-C3 – bity rodzaju wyniku
- TOP - wskaźnik stosu
- ES - znacznik błędu
- SF - znacznik błędu stosu
- PE - błąd niedokładności wyniku
- UE - błąd niedomiaru
- OE - błąd nadmiaru
- ZE - błąd dzielenia przez zero
- DE - błąd zdenormalizowanego argumentu
- IE - błąd niedozwolonej operacji

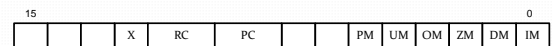
(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

11

Koprocesor

rejestr sterowania



- X – interpretacja nieskończoności (tylko 287)
- RC – sterowanie zaokrągleniem: do najbliższej(00), w dół (01), w górę (10), obcięcie (11)
- PC – sterowanie dokładnością obliczeń(23 (00), 53 (10) i 63 (11) bity)
- PM – maskowanie błędu niedokładności wyniku
- UM – maskowanie błędu niedomiaru
- OM – maskowanie błędu nadmiaru
- ZM – maskowanie błędu dzielenia przez zero
- DM – maskowanie błędu zdenormalizowanego argumentu
- IM – maskowanie błędu niedozwolonej operacji

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

12

Koprocesor

rejestr stanu zawartości rejestrów stosu

15								0
TAG(7)	TAG(6)	TAG(5)	TAG(4)	TAG(3)	TAG(2)	TAG(1)	TAG(0)	

TAG – pola określają zawartość poszczególnych rejestrów stosu:

00 – liczba prawidłowa

01 – zero

10 – wartość specjalna (nie liczba NaN, nieskończoność,...) lub zdenormalizowana

11 – rejestr pusty

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

13

Operacje przesyłania danych

- FLD załadowanie argumentu zmiennoprzecinkowego
- FST zapisanie wartości z wierzchołka stosu
- FSTP zapisanie wartości z wierzchołka stosu i usunięcie go za stosu
- FILD załadowanie liczby całkowitej
- FIST zapisanie liczby całkowitej
- FISTP zapisanie liczby całkowitej ze zdjęciem ze stosu
- FBLD załadowanie liczby BCD
- FBSTP zapisanie liczby BCD i zdjęcie jej ze stosu
- EXCH zamiana zawartości rejestrów
- FCMOVE przesłanie warunkowe (jeśli równe)
- FCMOVNE przesłanie warunkowe (jeśli nie równe)
- FCMOVNB przesłanie warunkowe (jeśli poniżej)
- FCMOVBE przesłanie warunkowe (jeśli poniżej lub równe)
- FCMOVNB przesłanie warunkowe (jeśli nie poniżej)
- FCMOVNBE przesłanie warunkowe (jeśli nie poniżej lub równe)
- FCMOVU przesłanie warunkowe (jeśli nieuporządkowane)
- FCMOVNU przesłanie warunkowe (jeśli uporządkowane)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

14

Instrukcja FLD

fld źródło

Przesyła liczbę zmiennoprzecinkową z rejestru *st(i)* lub z pamięci na wierzchołek stosu.

fpush(źródło)

fld *st(3)*

fld zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

15

Instrukcja FST

fst cel

Zapisuje liczbę zmiennoprzecinkową z wierzchołka stosu do rejestru *st(i)* lub pamięci.

cel := *st*

fst *st(3)*

fst zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

16

Instrukcja FSTP

fstp cel

Zapisuje liczbę zmiennoprzecinkową z wierzchołka stosu do rejestru *st(i)* lub pamięci i zdejmuje ją ze stosu.

cel := *st*

fpop

fstp *st(3)*

fstp zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

17

Instrukcja FILD

fild źródło

Przesyła liczbę całkowitą z pamięci na wierzchołek stosu.

fpush(źródło)

fild zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

18

Instrukcja FIST

fist cel

Zapisuje liczbę w formacie całkowitym z wierzchołka stosu do pamięci.

cel := int(st)

fist zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

19

Instrukcja FISTP

fistp cel

Zapisuje liczbę w formacie całkowitym z wierzchołka stosu do pamięci i zdejmuję ją ze stosu.

cel := int(st)

fpop

fistp zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

20

Instrukcja FBLD

fblld źródło

Przesyła liczbę całkowitą BCD z pamięci na wierzchołek stosu.

fpush(źródło)

fblld zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

21

Instrukcja FBSTP

fbstp cel

Zapisuje liczbę w formacie całkowitym BCD z wierzchołka stosu do pamięci i zdejmuję ją ze stosu.

cel := int(st)

fpop

fbstp zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

22

Instrukcja FXCH

fxch st(i)

fxch

Zamienia liczbę z wierzchołka stosu z wartością w rejestrze celu. Bez parametru celem jest st(1).

st(i) ↔ st

fxch st(5)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

23

Instrukcja FCMOVcc

FCMOVE	przesłanie warunkowe (jeśli równe, ZF=1)
FCMOVNE	przesłanie warunkowe (jeśli nie równe, ZF=0)
FCMOVBE	przesłanie warunkowe (jeśli poniżej, CF=1)
FCMOVBE	przesłanie warunkowe (jeśli poniżej lub równe, CF=1 lub ZF=1)
FCMOVNB	przesłanie warunkowe (jeśli nie poniżej, CF=0)
FCMOVNBE	przesłanie warunkowe (jeśli nie poniżej lub równe, CF=0 i ZF=0)
FCMOVU	przesłanie warunkowe (jeśli nieuporządkowane, PF=1)
FCMOVNU	przesłanie warunkowe (jeśli uporządkowane, PF=0)

fcmovcc st, st(i)

Jeśli jest spełniony warunek cc zapisuje do st wartość rejestru źródła st(i).
if cc then st := st(i)

fcmovnbe st, st(5)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

24

Operacje arytmetyczne

• FADD	dodawanie
• FADDP	dodawanie ze zdjeciem ze stosu
• FIADD	dodawanie liczby całkowitej
• FSUB	odejmowanie
• FSUBP	odejmowanie ze zdjeciem ze stosu
• FISUB	odejmowanie liczby całkowitej
• FSUBR	odejmowanie odwrotne
• FSUBRP	odejmowanie odwrotne ze zdjeciem ze stosu
• FISUBR	odejmowanie odwrotne liczby całkowitej
• FMUL	mnozenie
• FMULP	mnozenie ze zdjeciem ze stosu
• FIMUL	mnozenie liczby całkowitej
• FDIV	dzielenie
• FDIVP	dzielenie ze zdjeciem ze stosu
• FIDIV	dzielenie przez liczby całkowitą
• FDIVR	dzielenie odwrotne
• FDIVRP	dzielenie odwrotne ze zdjeciem ze stosu
• FIDIVR	dzielenie odwrotne liczby całkowitej
• FPREM	obliczenie reszty (częściowej) z dzielenia
• FPREM _i	obliczenie reszty (częściowej) z dzielenia zgodne z IEEE
• FABS	obliczenie wartości bezwzględnej
• FCHS	zmiana znaku
• FRNDINT	zaokrąglenie do liczby całkowitej
• FSCALE	skalowanie przez potęgę 2
• FSQRT	obliczenie pierwiastka kwadratowego
• FEXTRACT	obliczenie wykładnika i mantysy

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

25

Operacje arytmetyczne - formaty

- **Fop** **stosowy**
 - argumenty niejawnie: **cel - st(i), źródło - st**
 - wynik umieszczany jest w rejestrze celu, rejestr źródła zostaje zdjęty ze stosu
 - odpowiada **FopP st(i), st**
- **Fop st(i), st; Fop st, st(i)** rejestrowy
 - jednym z argumentów musi być rejestr wierzchołka stosu
- **FopP st(i), st** **rejestrowy ze zdjeciem ze stosu**
 - wynik umieszczany jest w st(i), argument źródłowy jest zdejmowany ze stosu
- **Fop zmienna** **z argumentem w pamięci**
 - argumentem celu jest wierzchołek stosu
- **Flop zmienna całk.** **z argumentem całkowitym w pamięci**
 - argumentem celu jest wierzchołek stosu

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

26

Instrukcja FADD/FADDP/FIADD

Dodaje do rejestru celu wartość źródła. Dla FIADD źródłem jest liczba całkowita. FADDP zdejmuje wierzchołek stosu.

$$\text{st(ce)} := \text{st(ce)} + \text{st(źródło)} | \text{zmienna}$$

fadd st(5), st

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

27

Instrukcja FSUB/FSUBP/FISUB

Odejmuje od rejestru celu wartość źródła. Dla FISUB źródłem jest liczba całkowita. FSUBP zdejmuje wierzchołek stosu.

$$\text{st(ce)} := \text{st(ce)} - \text{st(źródło)} | \text{zmienna}$$

fsubp st(3), st

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

28

Instrukcja FSUBR/FSUBRP/FISUBR

Odejmuje od źródła rejestr celu. Wynik umieszcza w rejestrze celu. Dla FISUBR źródłem jest liczba całkowita. FSUBRP zdejmuje wierzchołek stosu.

$$\text{st(ce)} := \text{st(źródło)} | \text{zmienna} - \text{st(ce)}$$

fsubr st(2), st

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

29

Instrukcja FMUL/FMULP/FIMUL

Mnoży rejestr celu przez wartość źródła. Dla FIMUL źródłem jest liczba całkowita. FMULP zdejmuje wierzchołek stosu.

$$\text{st(ce)} := \text{st(ce)} * \text{st(źródło)} | \text{zmienna}$$

fmulp st(4), st

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

30

Instrukcja FDIV/FDIVP/FIDIV

Dzieli rejestr celu przez wartość źródła. Dla FIDIV źródłem jest liczba całkowita. FDIVP zdejmuje wierzchołek stosu.

$$st(cel) := st(cel) / st(\text{źródło})|zmienna$$

```
fdiv st, st(4)
```

Instrukcja FDIVR/FDIVRP/FIDIVR

Dzieli źródło przez rejestr celu. Wynik umieszcza w rejestrze celu. Dla FIDIVR źródłem jest liczba całkowita. FDIVRP zdejmuje wierzchołek stosu.

$$st(cel) := st(\text{źródło})|zmienna/st(cel)$$

```
fdivr st, st(3)
```

Przykład

$$y = ax^3 + bx^2 + cx + d$$

fld d ;d	,wersja zoptymalizowana
fld x ;x; d	fld x ;x
fld st ;x; x; d	fld a ;a, x
fmul st, st(1) ;xx; x; d	fmul st, st(1) ;ax, x
fld st(1) ;x; xx; x; d	fadd b ;ax + b, x
fmul st, st(1) ;xxx; xx; x; d	fmul st, st(1) ;axx + bx, x
fmul a ;axxx; xx; x; d	fadd c ;axx + bx + c, x
faddp st(3), st ;xx; x; axxx+d	fmul ;axxx + bxx + cx
fmul b ;b*xx; x; axxx+d	fadd d ;axxx + bxx + cx + d
faddp st(2), st ;x; axxx+b*xx+d	fstp y
fmul c ;c*x; axxx+b*xx+d	
fadd ;axxx+b*xx+c*x+d	
fstp y	

Przykład

$$y = x_1 \cdots x_n \begin{matrix} z_1 \\ \vdots \\ z_n \end{matrix}$$

mov ecx, n		mov esi, x
mov esi, x		mov edi, y
mov edi, z		fld [esi] ;x
fld [esi] ;x		mov ecx, n
fld [edi] ;z; x		fmul [edi] ;s:=x*z
fmul ;s:=x*z		dec ecx
dec ecx	@:1	add esi, 8
@:1: add esi, 8		add edi, 8
add edi, 8		fld [esi] ;x; s
fld [esi] ;x; s		fmul [edi] ;x*z; s
fmul [edi] ;x*z; s		dec ecx
fadd ;s:=s+x*z		fadd ;s:=s+x*z
dec ecx		jnz @:1
jnz @:1		

Przykład

$$y = x_1 \cdots x_n \begin{matrix} z_1 \\ \vdots \\ z_n \end{matrix}$$

mov ecx, n		
fld zero		;s:=0
mov esi, x		
mov edi, z		
@:1: fld qword ptr[esi+8*ecx-8]		;x; s
fmul qword ptr[edi+8*ecx-8]		;x*z; s
fadd		;s:=s+x*z
dec ecx		
jnz @:1		

Instrukcja FPREM/FPREM1

Oblicza resztę z dzielenia st/st(1), wynik umieszcza w rejestrze st. Wynik jest dokładny. Jeśli st/st(1) jest zbyt duży (c2=1) w st umieszczona zostaje częściowa reszta i trzeba powtórzyć instrukcję. Zakres reszty:

```
FPREM <-|st(1)|, |st(1)|>
FPREM1 <-|st(1)/2|, |st(1)/2|>
Q := int|round(st/st(1))
st :=st - Q*st(1)
```

```
fprem
```

Instrukcja FABS

Oblicza wartość bezwzględną liczby z wierzchołka stosu.

$$st := |st|$$

fabs

Instrukcja FCHS

Zmienia znak liczby na wierzchołku stosu.

$$st := -st$$

fchs

Instrukcja FRNDINT

Zaokrągla liczbę na wierzchołku stosu.

$$st := \text{round}(st)$$

frndint

Instrukcja FSCALE

Skalowanie przez potęgę 2. Do wykładnika st dodaje część całkowitą st(i).

$$st := st * 2^{\text{int}(st(i))}$$

fscale

Instrukcja FSQRT

Oblicza pierwiastek kwadratowy z liczby na wierzchołku stosu.

$$st := \text{sqrt}(st)$$

fsqrt

Instrukcja FXTRACT

Oblicza wykładnik i mantysę liczby z rejestru st.

$$st := \text{wykładnik}(st)$$

$$fpush(\text{mantysa}(st))$$

fxtract

Przykład

Przekątna prostokąta $c = \sqrt{a^2 + b^2}$

```
fld a           ;a
fmul st, st    ;a*a
fld b           ;b, a*a
fmul st, st    ;b + b, a*a
fadd           ;a*a + b*b
fsqrt          ;c
fstp c
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

43

Przykład

$$y = \frac{\sqrt{|a-b|}}{a+b}$$

```
fld a           ;a
fld st         ;a, a
fld b           ;b, a, a
fadd st(2), st ;b, a, a+b
fsubp st(i), st ;a-b, a+b
fabs           ;|a-b|, (a+b)
fsqrt          ;sqrt(|a-b|), (a+b)
fdivr          ;sqrt(|a-b|)/(a+b)
fst y
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

44

Operacje ładowania stałych

- FLD1 zapisanie +1.0 na wierzchołku stosu
- FLDZ zapisanie +0.0 na wierzchołku stosu
- FLDPI zapisanie π na wierzchołku stosu
- FLDDL2E zapisanie $\log_2 e$ na wierzchołku stosu
- FLDLN2 zapisanie $\log_2 e$ ($\ln 2$) na wierzchołku stosu
- FLDDL2T zapisanie $\log_2 10$ na wierzchołku stosu
- FLDLG2 zapisanie $\log_{10} 2$ na wierzchołku stosu

Instrukcje zapisują stałe na wierzchołku stosu (st).

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

45

Przykład

$$y = 2\pi r$$

$$y = \pi r^2$$

```
fldpi          ;pi
fld r          ;r, pi
fmul          ;pi*r
fadd st, st    ;2*pi*r
fstp y

fldpi          ;pi
fld r          ;r, pi
fmul st, st    ;r*pi
fmul          ;pi*r*r
fstp y
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

46

Operacje funkcji przestępnych

- FSIN Oblicza sinus
- FCOS Oblicza cosinus
- FSINCOS Oblicza sinus i cosinus
- FPTAN Oblicza (częściowy) tangens
- FPATAN Oblicza (częściowy) arcus tangens
- F2XM1 Oblicza $2^x - 1$
- FYL2X Oblicza $y * \log_2 x$
- FYL2XP1 Oblicza $y * \log_2 (x+1)$

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

47

Instrukcja FSIN

Oblicza sinus liczby zawartej w st(o) i wynik umieszcza w st(o). Jeśli st nie zawiera się w $\langle -2^{63}, 2^{63} \rangle$, wówczas flaga C2 jest ustawiana. Argument można zredukować instrukcją FPREM z dzielnikiem 2π .

```
st := sin(st)
```

```
fsin
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

48

Instrukcja FCOS

Oblicza cosinus liczby zawartej w $st(o)$ i wynik umieszcza w $st(o)$. Jeśli st nie zawiera się w $\langle -2^{63}, 2^{63} \rangle$, wówczas flaga C2 jest ustawiana. Argument można zredukować instrukcją FPREM z dzielnikiem 2π .

```
st := cos(st)
```

```
fcos
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

49

Instrukcja FSINCOS

Oblicza sinus i cosinus liczby zawartej w $st(o)$ i wynik umieszcza w $st(o)$ i na wierzchołku stosu. Jeśli st nie zawiera się w $\langle -2^{63}, 2^{63} \rangle$, wówczas flaga C2 jest ustawiana. Argument można zredukować instrukcją FPREM z dzielnikiem 2π .

```
temp := cos(st)
```

```
st := sin(st)
```

```
fpush(temp)
```

```
fsincos
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

50

Instrukcja FPTAN

Oblicza tangens liczby zawartej w $st(o)$ i wynik umieszcza w $st(o)$ i 1.0 na wierzchołku stosu. Jeśli st nie zawiera się w $\langle -2^{63}, 2^{63} \rangle$, wówczas flaga C2 jest ustawiana. Argument można zredukować instrukcją FPREM z dzielnikiem 2π .

```
st := tan(st)
```

```
fpush(1.0)
```

```
fptan
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

51

Instrukcja FPATAN

Oblicza arcus tangens (kąt) ilorazu $st(1)/st$ i wynik umieszcza w $st(1)$, a st zdejmuje z wierzchołka stosu.

```
st(1) := arctg(st(1)/st)
```

```
fpop
```

```
fpatan
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

52

Instrukcja F2XM1

Oblicza $2^{st} - 1$. st musi być w przedziale $\langle -1, 1 \rangle$.

```
st:=2st - 1
```

```
f2xm1
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

53

Instrukcja FYL2X

Oblicza $y * \log_2 x$. $st > 0$

```
st(1) := st(1)* log2st
```

```
fpop
```

```
fyl2x
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

54

Instrukcja FYL2XP1

Oblicza $y * \log_2 x$. Liczba w rejestrze st musi spełniać

$$-(1 - \sqrt{5}/2) < st < (1 - \sqrt{5}/2)$$

$st(1) := st(1) * \log_2(st+1)$

fpop

fyl2xp1

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

55

Przykład

$$a = x^y = 2^{y \cdot \log_2 x}$$

```
fld y      ;y
fld x      ;x; y
fyl2x     ;y*log2x
fzxm1    ;2^(y*log2x) - 1  !!!!!
fldi     ;1; 2^(y*log2x) - 1
fadd      ;a
fstp a
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

56

Przykład

$$a = e^x = 2^{x \cdot \log_2 e}$$

```
fld x      ;x
fldl2e    ;log2e; x
fmul     ;x*log2e
fld st    ;x*log2e; x*log2e
frndint  ;round(x*log2e); x*log2e
fsub st(i), st ;round(x*log2e); x*log2e - round(x*log2e)
fchx st(i) ;x*log2e - round(x*log2e); round(x*log2e)
fzxm1    ;2^(x*log2e - round(x*log2e))-1; round(x*log2e)
fldi     ;1; 2^(x*log2e - round(x*log2e))-1; round(x*log2e)
fadd     ;2^(x*log2e - round(x*log2e)); round(x*log2e)
fscalc   ;2^(x*log2e - round(x*log2e))*2^round(x*log2e) ; round(x*log2e)
         ;2^(x*log2e - round(x*log2e)+round(x*log2e))= 2^(x*log2e) ; round(x*log2e)
fstp st(i) ;2^(x*log2e - round(x*log2e)+round(x*log2e))= 2^(x*log2e)
fstp a
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

57

Przykład

$$a = \log_b x = \log_2 x / \log_2 b$$

```
fldi     ;1
fld x     ;x; 1
fyl2x    ;log2x
fldi     ;1; log2x
fld b     ;b; 1; log2x
fyl2x    ;log2b; log2x
fdiv     ;a
fst a
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

58

Operacje porównania

- FCOM porównanie liczb zmiennoprzecinkowych
- FCOMP porównanie liczb zmiennoprzecinkowych i zdjęcie ze stosu
- FCOMPP porównanie liczb zmiennoprzecinkowych i podwójne zdjęcie ze stosu
- FUCOM nieuporządkowane porównanie liczb zmiennoprzecinkowych
- FUCOMP nieuporządkowane porównanie liczb zmiennoprzecinkowych i zdjęcie ze stosu
- FUCOMPP nieuporządkowane porównanie liczb zmiennoprzecinkowych i podwójne zdjęcie ze stosu
- FICOM porównanie z liczbą całkowitą
- FICOMP porównanie z liczbą całkowitą i zdjęcie ze stosu
- FCOMI porównanie liczb zmiennoprzecinkowych i ustawienie EFLAGS
- FUCOMI nieuporządkowane porównanie liczb zmiennoprzecinkowych i ustawienie EFLAGS
- FCOMIP porównanie liczb zmiennoprzecinkowych, ustawienie EFLAGS i zdjęcie ze stosu
- FUCOMIP nieuporządkowane porównanie liczb zmiennoprzecinkowych, ustawienie EFLAGS i zdjęcie ze stosu
- FTST porównanie z liczbą 0.0
- FXAM sprawdzenie liczby zmiennoprzecinkowej

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

59

Instrukcja

Wpływa na flagi: C3 C2 C0

FCOM/FCOMP/FCOMPP

fcom/fcomp źródło

fcompp

Porównanie liczb zmiennoprzecinkowych st i źródła. Źródłem może być rejestr st(i) lub zmienna. Jeśli źródło nie jest podane, to jest nim st(1). *Fcomp* zdejmuje liczbę z wierzchołka stosu, *fcompp* zdejmuje dwie liczby.

st(0) <=>? źródło

fpop ;dla fcomp,fcompp

fpop ;dla fcompp

fcom st(3)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

60

Stan flag po porównaniu

Flagi koprocesora C₃, C₂, C₀ odpowiadają flagom ZF, PF i CF procesora.

relacja \ flagi	C ₃	C ₂	C ₀
	ZF	PF	CF
st(o)>źródło	0	0	0
st(o)<źródło	0	0	1
st(o)=źródło	1	0	0
nieuporządkowane*	1	1	1

*flagi nie są ustawiane, jeśli wystąpi niezamaskowany wyjątek #IA

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

61

Instrukcja FUCOM/FUCOMP/FUCOMPP

Wpływa na flagi: C₃ C₂ C₀

fucom/fucomp st(i)

fucomp

Porównanie liczb zmiennoprzecinkowych st i st(i). Jeśli rejestr nie jest podany, to jest nim st(i). Fucomp odejmuje liczbę z wierzchołka stosu, fucomp odejmuje dwie liczby. Nie zgłaszają wyjątku #IA dla nieliczb pasywnych.

st(o) <=>? st(i)

fpop ;dla fucomp,fucomp

fpop ;dla fucomp

fucomp st(7)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

62

Instrukcja FICOM/FICOMP

Wpływa na flagi: C₃ C₂ C₀

ficom/ficomp zmienna

Porównanie st z liczbą całkowitą w pamięci (16/32). Ficomp odejmuje liczbę z wierzchołka stosu.

st(o) <=>? zmienna

fpop ;dla ficomp

ficom liczba_c

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

63

Instrukcja FCOMI/FCOMIP/FUCOMI/FUCOMIP

Wpływa na flagi: ZF PF CF

fcomi/fcomip st(i)

fucomi/fucomip st(i)

Porównanie st z liczbą w st(i). Fcompi/fucomip odejmuje liczbę z wierzchołka stosu. **Ustawia flagi: ZF, PF i CF.** Fucomi i fucomip nie zgłaszają wyjątku #IA dla nieliczb pasywnych

st(o) <=>? st(i)

fpop ;dla fcomip/fucomip

fucomi st(4)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

64

Instrukcja FTST

Wpływa na flagi: C₃ C₂ C₀

ftst

Porównanie liczb zmiennoprzecinkowych st i o.o.

st(o) <=>? o.o

ftst

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

65

Instrukcja FXAM

Wpływa na flagi: C₃ C₂ C₁ C₀

fxam

Sprawdza liczbę na wierzchołku stosu. C₁ = znak liczby.

znaczenie \ flagi	C ₃	C ₂	C ₀
	ZF	PF	CF
nieznormalizowana, pseudo (nie)liczba	0	0	0
nieliczba	0	0	1
znormalizowana	0	1	0
nieskończoność	0	1	1
zero	1	0	0
rejestr pusty	1	0	1
zdenormalizowana	1	1	0

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

66

Operacje sterowania koprocesorem

- FINCSTP zwiększenie rejestru wskaźnika stosu koprocatora
- FDECSTP zmniejszenie rejestru wskaźnika stosu koprocatora
- FFREE zwolnienie rejestru zmienoprzecinkowego
- FINIT inicjalizacja koprocatora po sprawdzeniu zgłoszenia błędu numerycznego
- FNINIT inicjalizacja koprocatora bez sprawdzenia zgłoszenia błędu numerycznego
- FCLEX zerowanie flag błędów numerycznych po sprawdzeniu zgłoszenia błędu numerycznego
- FNCLEX zerowanie flag błędów numerycznych bez sprawdzenia zgłoszenia błędu numerycznego
- FSTCW zapamiętanie rejestru sterowania po sprawdzeniu zgłoszenia błędu numerycznego
- FNSTCW zapamiętanie rejestru sterowania bez sprawdzenia zgłoszenia błędu numerycznego
- FLDCW wczytanie rejestru sterowania
- FSTENV zapamiętanie środowiska koprocatora po sprawdzeniu zgłoszenia błędu numerycznego
- FNSTENV zapamiętanie środowiska koprocatora bez sprawdzenia zgł. błędu numerycznego
- FLDENV wczytanie środowiska koprocatora
- FSAVE zapamiętanie zawartości koprocatora po sprawdzeniu zgłoszenia błędu numerycznego
- FNSAVE zapamiętanie zawartości koprocatora bez sprawdzenia zgłoszenia błędu numerycznego
- FRSTOR wczytanie zawartości koprocatora
- FSTSW zapamiętanie rejestru stanu po sprawdzeniu zgłoszenia błędu numerycznego
- FNSTSW zapamiętanie rejestru stanu bez sprawdzenia zgłoszenia błędu numerycznego
- WAIT/FWAIT czekanie na koprocator
- FNOP nic nie robi

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

67

Instrukcja FINCSTP

fincstp

Zwiększenie rejestru wskaźnika stosu koprocatora. Nie usuwa liczby ze stosu.

top := (top + 1) mod 8

fincstp

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

68

Instrukcja FDECSTP

fdecstp

Zmniejszenie rejestru wskaźnika stosu koprocatora.

top := (top - 1) mod 8

fdecstp

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

69

Instrukcja FFREE

ffree st(i)

Zwolnienie rejestru koprocatora. Rejestr wskaźnika stosu koprocatora nie jest zmieniany.

tag(i) := 11 b

ffree st(3)

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

70

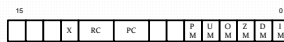
Instrukcja FINIT/FNINIT

finit/fninit

Inicjalizacja koprocatora. Fninit inicjalizacja koprocatora bez sprawdzenia zgłoszenia błędu numerycznego.

finit

rejestr stanu = 0
rejestr stanu zawartości rejestrów stosu = offh h
rejestr sterowania = 37f h



- X - interpretacja niekolejności (15bit 14bit)
- RC - sterowanie zaokrągleniem (do najbliższej(oo), w dół (oo), w górę (oo), obcięcie (o))
- PC - sterowanie dokładnością obliczeń(13 (oo), 15 (oo) i 63 (oo) bity)
- PM - maskowanie błędów niedokładności wyniku
- UM - maskowanie błędów niedostępu
- OM - maskowanie błędów nadmiaru
- ZM - maskowanie błędów dzielenia przez zero
- DM - maskowanie błędów zdenerwowanego argumentu
- DM - maskowanie błędów nieokreślonej operacji

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

71

Instrukcja FCLEX/FNCLEX

fclex/fnclex

Zerowanie flag błędów numerycznych. Fnclex - bez sprawdzenia zgłoszenia błędu numerycznego.

fnclex

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

72

Instrukcja FSTCW/FNSTCW

fstcw/fnstcw cel

Zapamiętanie rejestru sterowania. Cel jest dwubajtową zmienną albo rejestrem AX. Fnstcw - bez sprawdzenia zgłoszenia błędu numerycznego.

fnstcw

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

73

Instrukcja FLDCW

fldcw źródło

Wczytanie rejestru sterowania. Źródło jest dwubajtową zmienną.

fldcw zmienna

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

74

Instrukcja FSTENV/FNSTENV

fstenv/fnstenv cel

Zapamiętanie środowiska koprocatora . Cel jest 14/28 bajtowym obszarem (tryb 16/32 bitowy). Fnstenv - bez sprawdzenia zgłoszenia błędu numerycznego.

fnstenv fsrodowisko

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

75

Instrukcja FLDENV

fldenv źródło

Wczytanie środowiska koprocatora . Źródło jest 14/28 bajtowym obszarem (tryb 16/32 bitowy).

fldenv fsrodowisko

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

76

Instrukcja FSAVE/FNSAVE

fsave/fnsave cel

Zapamiętanie zawartości koprocatora (środowisko + rejestry zmiennoprzecinkowe). Cel jest 94/108 bajtowym obszarem (tryb 16/32 bitowy). Fnsave - bez sprawdzenia zgłoszenia błędu numerycznego.

fnsave fstan

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

77

Instrukcja FRSTOR

frstor źródło

Wczytanie zawartości koprocatora . Źródło jest 94/108 bajtowym obszarem (tryb 16/32 bitowy).

frstor fstan

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

78

Instrukcja FSTSW/FNSTSW

fstsw/fnstsw cel

Zapamiętanie rejestru stanu. Cel jest dwubajtową zmienną albo rejestrem AX. Fnstsw - bez sprawdzenia zgłoszenia błędu numerycznego.

fnstsw

Instrukcja WAIT/FWAIT

wait/fwait

Czekanie przez procesor na gotowość koprocatora (na zakończenie wykonywania instrukcji).

fwait

Instrukcja FNOP

fnop

Nic nie robi.

fnop

Przykład

$$\sqrt{\Delta} = \sqrt{b^2 - 4ac}$$

fld b	jb	fld b	jb
fld st	jb; b	fld st	jb; b
fmul	jb	fmul	jb
fld a	a; bb	fld a	a; bb
fmul c	ac; bb	fmul c	ac; bb
fadd st, st	2ac; bb	fadd st, st	2ac; bb
fadd st, st	4ac; bb	fadd st, st	4ac; bb
fsub	bb-4ac	fsub	bb-4ac
fist	porównanie z 0,0	fldz	0; bb-4ac
fstsw ax	; ax-stan	fcomip st, st(1)	porównanie z 0,0
sahf	stan do flag	jp blad	flagi już ustawione
jp blad		jz rowne	
jz rowne		jc wieksze	
jc mniejsze		mc mniejsze:	
wieksze:			

Przykład - maksimum

```

32 bity
;adresy 3 zmiennych na stosie
mov eax,a
mov edx,b
fld qword ptr[eax] ;a
fld qword ptr[edx] ;b,a
fcomi st(1)
fcmovb st,st(1) ;max,a
fstp st(1) ;max
mov eax,m
fstp qword ptr[eax]
ret

;2 wartości na stosie
fld a ;a
fld b ;b,a
fcomi st(1) ;b,a
fcmovb st,st(1) ;max,a
fstp st(1) ;max
ret

64 bity
;adresy 3 zmiennych w rejestrach
fld qword ptr[rcx] ;a
fld qword ptr[rdx] ;b,a
fcomi st(1)
fcmovb st,st(1) ;max,a
fstp st(1) ;max
fstp qword ptr[r8]
ret

```