

10

3



5

e

Wartości

poprawne i niepoprawne zawarte w rejestrach stosu koprocesora.

Klasa	Znak	Przesunięta cecha	Mantysa		
			Część całkowita	Część ułamkowa	
Dodatkne liczby	Passywna (cisza)	0	11.11	0	11.11
	-	0	11.11		10.00
	Aktywne (szkła)	0	11.11	0	01.11
Dodatkne zmienneprzecinkowe	Nieskończoność	0	11.11	0	00.00
	Znormalizowane	0	11.10	1	11.11
	-	0	00.01		00.00
	Nieznormalizowane	0	11.10	0	11.11
	-	0	00.01		00.00
	Pseudo-zdenormalizowane	0	00.00	1	11.11
-	0	00.00		00.00	
Zero	0/1	00.00	0	00.00	
Ujemne zmienneprzecinkowe	Pseudo-zdenormalizowane	1	00.00	1	11.11
	-	1	00.00		00.00
	Nieznormalizowane	1	11.10	0	11.01
	-	1	00.01		00.00
	Znormalizowane	1	11.10	1	11.01
	-	1	00.01		00.00
Minus nieskończoność	1	11.11	0	00.00	
Ujemne liczby	Aktywne (szkła)	1	11.11	0	01.11
	-	1	11.11		00.01
	Passywna (cisza)	1	11.11	0	11.11
-	1	11.11		10.00	

(C) IISI d.KIK PCz 2013

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

8

Koprocesor - budowa

stos rejestrów

79

0

R7

R6

R5

R4

R3

R2

R1

R0

ST(2)

ST(1)

ST(0)

szczyt stosu

rejestr:

15

0

sterowania

stanu

stanu stosu

47

0

wskaźnika rozkazu

wskaźnika operandu

10

0

kod operacyjny

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

9

Koprocesor

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

79

64

63

62

0

znak

wykładnik

bit wartości całkowitej

mantysa

rozszerzonej precyzji

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

10

Koprocesor

rejestr stanu

15

0

B

C3

TOP

C2

C1

Co

ES

SF

PE

UE

OE

ZE

DE

IE

B – koprocesor zajęty

Co-C3 – bity rodzaju wyniku

TOP – wskaźnik stosu

ES – znacznik błędu

SF – znacznik błędu stosu

PE – błąd niedokładności wyniku

UE – błąd nadmiaru

OE – błąd nadmiaru

ZE – błąd dzielenia przez zero

DE – błąd zdenormalizowanego argumentu

IE – błąd niedozwolonej operacji

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

11

Koprocesor

rejestr sterowania

15

0

X

RC

PC

PM

UM

OM

ZM

DM

IM

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 nadmiaru

OM – maskowanie błędu nadmiaru

ZM – maskowanie błędu dzielenia przez zero

DM – maskowanie błędu zdenormalizowanego argumentu

IM – maskowanie błędu niedozwolonej operacji

(C) IISI d.KIK PCz 2013

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) IISI d.KIK PCz 2013

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 ze 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
- FXCH zamiana zawartości rejestrów
- FCMOVE przesłanie warunkowe (jeśli równe)
- FCMOVNE przesłanie warunkowe (jeśli nie równe)
- FCMOVB 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) IISI d.KIK PCz 2013

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

15

Instrukcja FST

`fst cel`

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

`cel:=st(0)`

`fst st(3)`

`fst zmienna`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

16

Instrukcja FSTP

`fstp cel`

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

`cel:=st(0)`

`fpop`

`fstp st(3)`

`fstp zmienna`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

17

Instrukcja FILD

`fild źródło`

Przesyła liczbę całkowitą z pamięci na wierzchołek stosu.

`fpush(źródło)`

`fild zmienna`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

18

Instrukcja FIST

`fist cel`

Zapisuje liczbę w formacie całkowitym z wierzchołka stosu do pamięci.

`cel:=int(st(o))`

`fist zmienna`

(C) IISI d.KIK PCz 2013

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(o))`

`fpop`

`fistp zmienna`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

20

Instrukcja FBLD

`fbld źródło`

Przesyła liczbę całkowitą BCD z pamięci na wierzchołek stosu.

`fpush(źródło)`

`fbld zmienna`

(C) IISI d.KIK PCz 2013

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(o))`

`fpop`

`fbstp zmienna`

(C) IISI d.KIK PCz 2013

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(o)`

`fxch st(5)`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

23

Instrukcja FCMOVcc

FCMOVE	przesłanie warunkowe (jeśli równe, ZF=1)
FCMOVNE	przesłanie warunkowe (jeśli nie równe, ZF=0)
FCMOVB	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(o),st(i)`

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

`fcmovnbe st(o),st(5)`

(C) IISI d.KIK PCz 2013

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 mnożenie
- FMULP mnożenie ze zdjeciem ze stosu
- FIMUL mnożenie liczby całkowitej
- FDIV dzielenie
- FDIVP dzielenie ze zdjeciem ze stosu
- FIDIV dzielenie przez liczbę całkowitą
- FDIVR dzielenie odwrotne
- FDIVRP dzielenie odwrotne ze zdjeciem ze stosu
- FIDIVR dzielenie odwrotne liczby całkowitej
- FPREM obliczenie reszty (całkowitej) z dzielenia
- FPREM1 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
- FXTRACT obliczenie wykładnika i mantysy

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

25

Operacje arytmetyczne - formaty

- **Fop** **stosowy**
 - argumenty niejawne: **cel - st(i), źródło - st(o)**
 - 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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

26

Instrukcja FADD/FADDP/FIADD

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

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

fadd st(5),st

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

27

Instrukcja FSUB/FSUBP/FISUB

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

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

fsubp st(3),st

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

28

Instrukcja FSUBR/FSUBRP/FISUBR

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

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

fsubr st(2),st

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

29

Instrukcja FMUL/FMULP/FIMUL

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

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

fmulp st(4),st

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

30

Instrukcja FDIV/FDIVP/FIDIV

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

$$st(cel) := st(cel) / st(źródło) | zmienna$$

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

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

31

Instrukcja FDIVR/FDIVRP/FIDIVR

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

$$st(cel) := st(źródło) | zmienna / st(cel)$$

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

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

32

Przykład

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

```
fld  d      ;d
fld  x      ;x; d
fld  st     ;x; x; d
fmul st,st(1) ;xx; x; d
fld  st(1)  ;x; xx; x; d
fmul st,st(1) ;xxx; xx; x; d
fmul a      ;axxx; xx; x; d
faddp st(3),st ;xx; x; axxx+d
fmul b      ;b*xx; x; axxx+d
faddp st(2),st ;x; axxx+b*xx+d
fmul c      ;c*x; axxx+b*xx+c*x+d
fadd      ;axxx+b*xx+c*x+d
fstp  y
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

33

Przykład

$$y = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$

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

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

34

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 $<-|st(1)/2|, |st(1)/2|>$

$$Q := \text{int} \lfloor \text{round}(st/st(1)) \rfloor$$

$$st := st - Q * st(1)$$

```
fprem
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

35

Instrukcja FABS

Oblicza wartość bezwzględną liczby z wierzchołka stosu.

$$st := |st|$$

```
fabs
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

36

Instrukcja FCHS

Zmienia znak liczby na wierzchołku stosu.

`st:=-st`

`fchs`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

37

Instrukcja FRNDINT

Zaokrągla liczbę na wierzchołku stosu.

`st:=round(st)`

`frndint`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

38

Instrukcja FSCALE

Skalowanie przez potęgę 2. Do wykładnika st dodaje część całkowitą st(1).

`st:=st*2int(st(1))`

`fscale`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

39

Instrukcja FSQRT

Oblicza pierwiastek kwadratowy z liczby na wierzchołku stosu.

`st:=sqrt(st)`

`fsqrt`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

40

Instrukcja FXTRACT

Oblicza wykładnik i mantysę liczby z rejestru st.

`st:=wykładnik(st)`
`fpush(mantysa(st))`

`fxtract`

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

41

Przykład

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

<code>fld a</code>	<code>a</code>
<code>fld st(0)</code>	<code>a; a</code>
<code>fld b</code>	<code>b; a; a</code>
<code>fadd st(2),st</code>	<code>b; a; a+b</code>
<code>fsubp st(1),st</code>	<code>a-b; a+b</code>
<code>fabs</code>	<code> a-b ; (a+b)</code>
<code>fsqrt</code>	<code>sqrt(a-b); (a+b)</code>
<code>fdivr</code>	<code>aqrt(a-b)/(a+b)</code>
<code>fst y</code>	

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

42

Operacje ładowania stałych

- FLD1 zapisanie +1.0 na wierzchołku stosu
- FLDZ zapisanie +0.0 na wierzchołku stosu
- FLDPI zapisanie π na wierzchołku stosu
- FLDL2E zapisanie $\log_2 e$ na wierzchołku stosu
- FLDLN2 zapisanie $\log_2 2$ ($\ln 2$) na wierzchołku stosu
- FLDL2T zapisanie $\log_2 10$ na wierzchołku stosu
- FLDLG2 zapisanie $\log_{10} 2$ na wierzchołku stosu

Instrukcje zapisują stałe na wierzchołku stosu (st).

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

43

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

44

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

45

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

46

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

47

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

48

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

49

Instrukcja F2XM1

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

$st := 2^{st} - 1$

f2xm1

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

50

Instrukcja FYL2X

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

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

fpop

fyl2x

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

51

Instrukcja FYL2XP1

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

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

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

fpop

fyl2xp1

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

52

Przykład

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

```
fld y      ;y
fld x      ;x; y
fyl2x      ;y*log2x
f2xm1      ;2^(y*log2x)-1  !!!!!!!
fldi       ;1; 2^(y*log2x)-1
fadd       ;a
fstp a
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

53

Przykład

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

```
fld x      ;x
fldl2e     ;log2e; x
fmul       ;x*log2e
fld st(0)  ;x*log2e; x*log2e
frndint    ;round(x*log2e); x*log2e
fsub st(1), st ;round(x*log2e); x*log2e - round(x*log2e)
fexch st(1) ;x*log2e - round(x*log2e); round(x*log2e)
f2xm1      ;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)
fscale     ;2^(x*log2e - round(x*log2e))*2^round(x*log2e)
           ;2^(x*log2e - round(x*log2e)+round(x*log2e))= 2^(x*log2e)
fstp a
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

54

Przykład

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

```
fldi      ;1
fld x     ;x; 1
fyl2x    ;log2 x
fldi     ;1; log2 x
fld b    ;b; 1; log2 x
fyl2x    ;log2 b; log2 x
fdiv     ;a
fst a
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

55

Operacje porównania

- FCOM
- FCOMP
- FCOMPP
- FUCOM
- FUCOMP
- FUCOMPP
- FICOM
- FICOMP
- FCOMI
- FUCOMI
- FCOMIP
- FUCOMIP
- FTST
- FXAM

porównanie liczb zmiennoprzecinkowych

porównanie liczb zmiennoprzecinkowych i zdjęcie ze stosu

porównanie liczb zmiennoprzecinkowych i podwójne zdjęcie ze stosu

nieuporządkowane porównanie liczb zmiennoprzecinkowych

nieuporządkowane porównanie liczb zmiennoprzecinkowych i zdjęcie ze stosu

nieuporządkowane porównanie liczb zmiennoprzecinkowych i podwójne zdjęcie ze stosu

porównanie z liczbą całkowitą

porównanie z liczbą całkowitą i zdjęcie ze stosu

porównanie liczb zmiennoprzecinkowych i ustawienie EFLAGS

nieuporządkowane porównanie liczb zmiennoprzecinkowych i ustawienie EFLAGS

porównanie liczb zmiennoprzecinkowych, ustawienie EFLAGS i zdjęcie ze stosu

nieuporządkowane porównanie liczb zmiennoprzecinkowych, ustawienie EFLAGS i zdjęcie ze stosu

porównanie z liczbą 0.0

sprawdzenie liczby zmiennoprzecinkowej

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

56

Instrukcja

Wpływa na flagi: C3 C2 C0

FCOM/FCOMP/FCOMPP

```
fcom/fcomp źródło
fcomp
```

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, fcomp zdejmuje dwie liczby.

```
st(o)<=>? źródło
fpop ;dla fcomp,fcomp
fpop ;dla fcomp
```

```
fcom st(3)
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

57

Stan flag po porównaniu

Flagi koprocatora C3, C2, Co odpowiadają flagom ZF, PF i CF procesora.

relacja \ flagi	C3	C2	Co
	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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

58

Instrukcja

Wpływa na flagi: C3 C2 C0

FUCOM/FUCOMP/FUCOMPP

```
fucom/fucomp st(i)
fucomp
```

Porównanie liczb zmiennoprzecinkowych st i st(i). Jeśli rejestr nie jest podany, to jest nim st(1). Fucomp zdejmuje liczbę z wierzchołka stosu, fucomp zdejmuje dwie liczby. Nie zgłaszają wyjątku #IA dla nieliczb pasywnych.

```
st(o)<=>? st(i)
fpop ;dla fucomp,fucomp
fpop ;dla fucomp
```

```
fucomp st(7)
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

59

Instrukcja FICOM/FICOMP

Wpływa na flagi: C3 C2 C0

```
ficom/ficomp zmienna
```

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

```
st(o)<=>? zmienna
fpop ;dla ficomp
```

```
ficomp liczba_c
```

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

60

Wpływa na flagi: ZF PF CF

Instrukcja FCOMI/FCOMIP/FUCOMI/FUCOMIP

fcomi/fcomip st(i)
fucomi/fucomip st(i)

Porównanie st z liczbą w st(i). Fcompi/fucomip zdejmuję 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

fcomi st(4)

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

61

Wpływa na flagi: C3 C2 C0

Instrukcja FTST

ftst

Porównanie liczb zmiennoprzecinkowych st i o.o.

st(o)<=>? o.o

ftst

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

62

Wpływa na flagi: C3 C2 C1 C0

Instrukcja FXAM

fxam

Sprawdza liczbę na wierzchołku stosu. C1 = znak liczby.

znaczenie \ flagi	C3	C2	C0
	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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

63

Operacje sterowania koprocesorem

- FINCSTP zwiększenie rejestru wskaźnika stosu koprocesora
- FDECSTP zmniejszenie rejestru wskaźnika stosu koprocesora
- FFREE zwolnienie rejestru zmiennoprzecinkowego
- FINIT inicjalizacja koprocesora po sprawdzeniu zgłoszenia błędu numerycznego
- FNINIT inicjalizacja koprocesora 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 koprocesora po sprawdzeniu zgłoszenia błędu numerycznego
- FNSTENV zapamiętanie środowiska koprocesora bez sprawdzenia zgl. błędu numerycznego
- FLDENV wczytanie środowiska koprocesora
- FSAVE zapamiętanie zawartości koprocesora po sprawdzeniu zgłoszenia błędu numerycznego
- FNSAVE zapamiętanie zawartości koprocesora bez sprawdzenia zgłoszenia błędu numerycznego
- FRSTOR wczytanie zawartości koprocesora
- FSTSW zapamiętanie rejestru stanu po sprawdzeniu zgłoszenia błędu numerycznego
- FNSTSW zapamiętanie rejestru stanu bez sprawdzenia zgłoszenia błędu numerycznego
- WAIT/FWAIT oczekiwanie na koprocesor
- FNOP nic nie robi

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

64

Instrukcja FINCSTP

fincstp

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

top:= (top+1) mod 8

fincstp

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

65

Instrukcja FDECSTP

fdecstp

Zmniejszenie rejestru wskaźnika stosu koprocesora.

top:= (top-1) mod 8

fdecstp

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

66

Instrukcja FFREE

ffree st(i)

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

tag(i):= 11 b

ffree st(3)

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

67

Instrukcja FINIT/FNINIT

finit/fninit

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

finit

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

X - interpretacja nieskończoności (tylko 28b)
RC - sterowanie zaokrągleniem: do najbliższej(oo), w dół (so), w górę (no), obcięcie (u)
PC - sterowanie dokładnością obliczeń(13 (oo), 53 (so) i 63 (u) bity)
PM - maskowanie błędów niedokładności wyniku
UM - maskowanie błędów niedomiaru
OM - maskowanie błędów nadmiaru
ZM - maskowanie błędów dzielenia przez zero
DM - maskowanie błędów znormalizowanego argumentu
IM - maskowanie błędów niedozwolonej operacji

15												0			
			X	RC	PC			P	U	O	Z	D	I		
								M	M	M	M	M	M		

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

68

Instrukcja FCLEX/FNCLEX

fclex/fnclex

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

fnclex

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

69

Instrukcja FSTCW/FNSTCW

fstcw/fnstcw cel

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

fnstcw

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

70

Instrukcja FLDCW

fldcw źródło

Wczytanie rejestru sterowania. Źródło jest dwubajtową zmienną.

fldcw zmienna

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

71

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) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

72

Instrukcja FLDENV

fldenv źródło

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

fldenv fsrodowisko

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

73

Instrukcja FSAVE/FNSAVE

fsave/fnsave cel

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

fnsave fstan

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

74

Instrukcja FRSTOR

frstor źródło

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

frstor fstan

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

75

Instrukcja FSTSW/FNSTSW

fstsw/fnstsw cel

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

fnstsw

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

76

Instrukcja WAIT/FWAIT

wait/fwait

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

fwait

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

77

Instrukcja FNOP

fnop

Nic nie robi.

fnop

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

78

Przykład

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

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

(C) IISI d.KIK PCz 2013

Programowanie niskopoziomowe

79