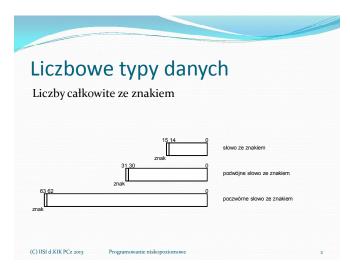
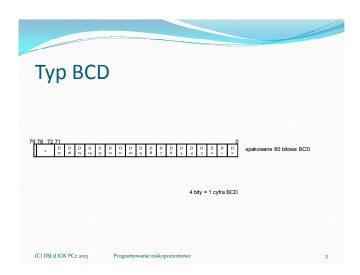
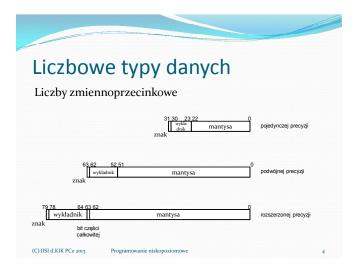
(C) IISI d.KIK PCz







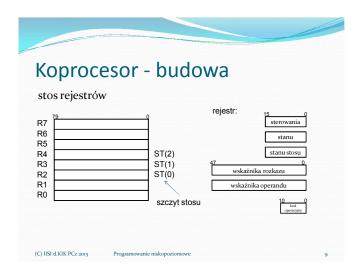


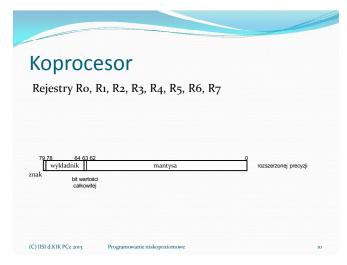


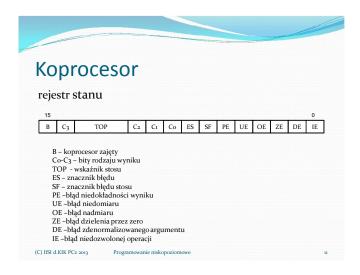










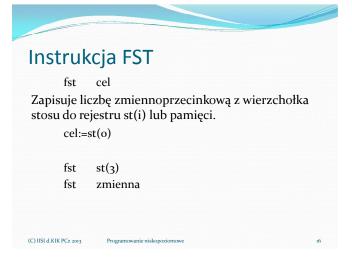


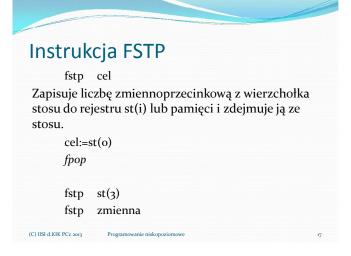














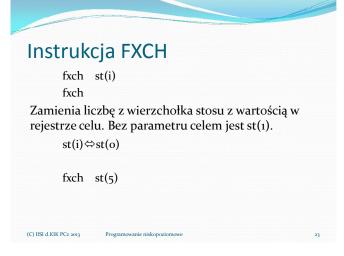
(C) IISI d.KIK PCz 2013

Instrukcja FIST fist cel Zapisuje liczbę w formacie całkowitym z wierzchołka stosu do pamięci. cel:=int(st(o)) fist zmienna

Instrukcja FISTP fistp cel Zapisuje liczbę w formacie całkowitym z wierzchołka stosu do pamięci i zdejmuje ją ze stosu. cel:=int(st(o)) fpop fistp zmienna



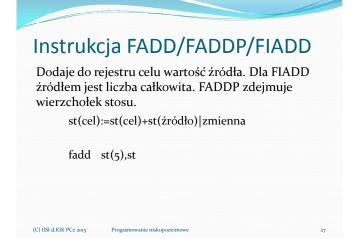














Instrukcja FSUBR/FSUBRP/FISUBR Odejmuje od źródła rejestr celu. Wynik umieszcza w rejestrze celu. Dla FISUBR źródłem jest liczba całkowita. FSUBRP zdejmuje wierzchołek stosu. st(cel):=st(źródło)|zmienna-st(cel) fsubr st(2),st



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

Instrukcja FDIVR/FDIVRP/FIDIVR

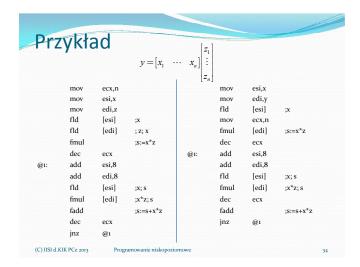
<u>Dzieli źródło przez rejestr celu</u>. Wynik umieszcza w rejestrze celu. Dla FIDIVR źródłem jest liczba całkowita. FDIVRP zdejmuje wierzchołek stosu.

st(cel):=st(źródło)|zmienna/st(cel)

fdivr st,st(3)

(C) IISI d.KIK PCz 2013 Programowanie niskopoziomow





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

(C) IISI d.KIK PCz 2013

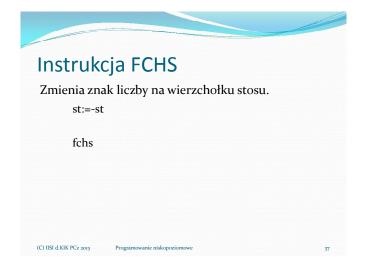
Instrukcja FABS

Oblicza wartość bezwzględną liczby z wierzchołka stosu.

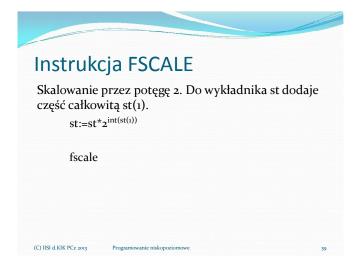
st = |st|

fabs

(C) IISI d.KIK PCz 2013 Programowanie niskopoziomowe











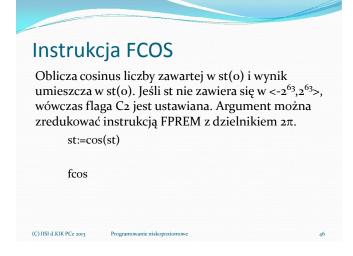


(C) IISI d.KIK PCz 2013

Operacje ładowania stałych • FLD1 zapisanie +1.0 na wierzchołku stosu FLDZ zapisanie +o.o na wierzchołku stosu FLDPI zapisanie π na wierzchołku stosu FLDL₂E zapisanie log, e na wierzchołku stosu • FLDLN₂ zapisanie loge2 (ln2) na wierzchołku stosu zapisanie log₂10 na wierzchołku stosu FLDL2T • FLDLG2 zapisanie log102 na wierzchołku stosu Instrukcje zapisują stałe na wierzchołku stosu (st).

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*log₂x FYL2XP1 Oblicza y*log,(x+1) (C) IISI d.KIK PCz 2013

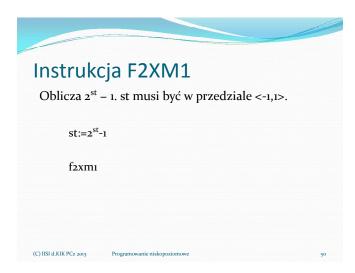
Instrukcja FSIN Oblicza sinus liczby zawartej w st(o) i wynik umieszcza w st(o). Jeśli st nie zawiera się w <-2⁶³,2⁶³>, wówczas flaga C2 jest ustawiana. Argument można zredukować instrukcją FPREM z dzielnikiem 2π. st:=sin(st) fsin



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 <-2⁶³,2⁶³>, wówczas flaga C2 jest ustawiana. Argument można zredukować instrukcją FPREM z dzielnikiem 2π. temp:=cos(st) st:=sin(st) fpush(temp) fsincos (C) IISI ALKIK PC2 2013 Programowanie niskopoziomowe 47







```
Instrukcja FYL2X

Oblicza y*log_2x. st>0

st(1):=st(1)*log_2st
fpop

fyl_2x

(C) IISI d.KIK PCz 2013 Programowanie niskopoziomowe 51
```

```
Instrukcja FYL2XP1

Oblicza y*\log_2 x. Liczba w rejestrze st musi spełniać -(1-\sqrt[3]{2}) < st < (1-\sqrt[3]{2})

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

fpop

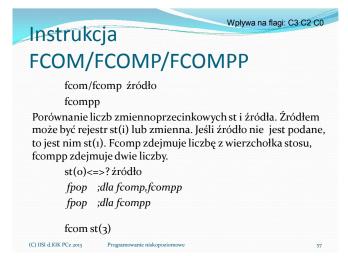
fyl2xp1

(C) ISI d.KIK PCz 2013 Programowanie niskopoziomowe 52
```

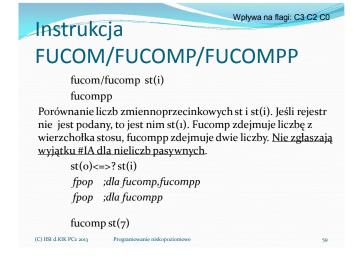
```
Przykład
  a = e^x = 2^{x \cdot \log_2 e}
      fld x
      fldl2e
                      ; log2e; x
      fmul
                      ; x*log2e
      fld st(o)
                      ; x*log2e; x*log2e
                      ; round(x*log2e); x*log2e
      fsub st(1), st
                      ; round(x*log2e); x*log2e - round(x*log2e)
                      ; x*log2e - round(x*log2e); round(x*log2e)
      fxch st(1)
                      ; 2^{(x*log2e - round(x*log2e))-1}; round(x*log2e)
      f2xm1
                      ; 1,2^(x*log2e - round(x*log2e))-1; round(x*log2e)
      fadd
                      ; 2^(x*log2e - round(x*log2e)); round(x*log2e)
                      ;_2^{(x*log2e - round(x*log2e))*_2^{round(x*log2e)}}
      fscale
                      ;2^(x*log2e - round(x*log2e)+round(x*log2e))= 2^(x*log2e)
     fstp a
 (C) IISI d.KIK PCz 2013
```



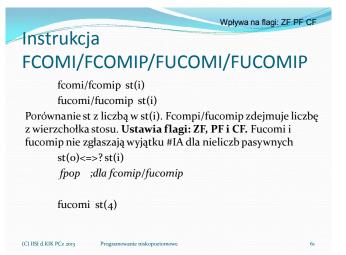
Operacje porównania FCOMP porównanie liczb zmiennoprzecinkowych i zdjęcie ze stosu FCOMPP porównanie liczb zmiennoprzecinkowych i podwójne zdjęcie ze stosu FUCOM nieuporządkowane porównanie liczb zmiennoprzecinkowych nieuporządkowane porównanie liczb zmiennoprzecinkowych i zdjęcie ze stosu $\,$ FUCOMP 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 nieuporządkowane porównanie liczb zmiennoprzecinkowych i FUCOMI FCOMIF porównanie liczb zmiennoprzecinkowych, ustawienie EFLAGS i zdjęcie ze stosu FUCOMIP nieuporządkowane porównanie liczb zmiennoprzecinkowych, ustawienie EFLAGS i zdjęcie ze stosu porównanie z liczbą o.o FTST FXAM sprawdzenie liczby zmiennoprzecinkowej (C) IISI d.KIK PCz 2013













Wpływa na flagi: C3 C2 C1 C0 Instrukcja FXAM Sprawdza liczbę na wierzchołku stosu. C1 = znak liczby. C3 C₂ Co znaczenie ZF nieznormalizowana. o o o pseudo (nie)liczba nieliczba 0 znormalizowana 0 nieskończoność o zero o o

