

POLITECHNIKA CZĘSTOCHOWSKA
KATEDRA INTELIGENTNYCH SYSTEMÓW INFORMATYCZNYCH



PROGRAMOWANIE NISKOPOZIOMOWE

LABORATORIUM 7

OBLICZENIA Z WYKORZYSTANIEM ZBIORU
INSTRUKCJI X87

dr inż. Bartosz Kowalczyk

Częstochowa, 28 maja 2021

1 Asembler koprocatora x87

Koprocesor arytmetyczny x87 posiada implementację wielu generycznych operacji matematycznych. Niestety, ich lista jest ograniczona przez co istnieją tożsamości matematyczne, których implementacja w koprocesorze wymaga zastosowania odpowiednich przekształceń.

2 Przydatne tożsamości matematyczne

- $a^b = c \Leftrightarrow \log_a c = b$
- $\log_b c = \frac{\log_a c}{\log_a b}$
- $a^b = c^{b \log_c a}$
- $a^n b^n = (ab)^n$
- $a^m a^n = a^{m+n}$
- $\frac{a^n}{b^n} = \left(\frac{a}{b}\right)^n$
- $\frac{a^m}{a^n} = a^{m-n}$
- $(a^m)^n = a^{mn}$

3 Przydatne instrukcje

- `fist`
- `fistp`
- `fcom`
- `fcomi`
- `flde`
- `fchs`
- `fxch`
- `frndint`
- `f2xm1`
- `fscale`

3.1 Wartość funkcji wykładniczej dla liczb typu double

Asembler x87 nie posiada instrukcji, która realizuje jawną operację potęgowania dla dowolnego wykładnika. Należy zatem skorzystać z własności:

$$a^b = c^{b \log_c a} \text{ oraz } a^m a^n = a^{m+n} \quad \text{gdzie } c = 2 \text{ i } a > 0 \quad (1)$$

Operację $2^{st} - 1$ realizuje instrukcja `f2xm1`. Jednak posiada ona pewne ograniczenie. Wartość znajdująca się na wierzchołku stosu (`st`) musi być liczbą z przedziału $st \in [-1, 1]$. W celu spełnienia tego wymagania należy zastosować działanie:

$$m = z - \lfloor z \rfloor \quad \text{gdzie } z = b \log_c a \quad (2)$$

Wartość m jest wówczas z przedziału $m \in [-1, 1]$. Zobaczmy to na przykładzie:

$$\begin{aligned} \text{Niech } z &= 3.5235 \\ m &= z - \lfloor z \rfloor \\ m &= 3.5235 - 3 = 0.5235 \\ \text{więc } m &\in [-1, 1] \end{aligned} \quad (3)$$

Następnie, należy użyć instrukcji `fscale`, która realizuje działanie $st \cdot 2^{st(1)}$. Tym razem nie ma żadnych ograniczeń. W ten sposób korzystamy z własności potęgowania liczb o

tej samej podstawie i różnych wykładnikach ($a^m a^n = a^{m+n}$).

Zadanie. Napisać wstawkę asemblerową obliczającą wartość funkcji:

$$y = |a|^x$$

```

1      fld x                ;// x
2      fld a                ;// a, x
3      fabs                 ;// |a|, x
4      fyl2x                ;// xlog_2|a|
5      fld st               ;// xlog_2|a|, xlog_2|a|
6      frndint              ;// int(xlog_2|a|), xlog_2|a|
7      fsub st(1), st       ;// int(xlog_2|a|), xlog_2|a| -
                           ;// int(xlog_2|a|)
8      fxch st(1)           ;// xlog_2|a| - int(xlog_2|a|), xlog_2|a|
9      f2xm1                ;// 2^(xlog_2|a| - int(xlog_2|a|)) - 1,
                           ;// xlog_2|a|
10     fld1                 ;// 1, 2^(xlog_2|a| - int(xlog_2|a|)) -
                           ;// 1, xlog_2|a|
11     fadd                 ;// 2^(xlog_2|a| - int(xlog_2|a|)),
                           ;// xlog_2|a|
12     fscale               ;// 2^(xlog_2|a| - int(xlog_2|a|)) *
                           ;// 2^(xlog_2|a|), xlog_2|a|
13     fstp yAsm            ;// xlog_2|a|
14     fstp st              ;// <empty>

```

3.2 Logarytm o dowolnej podstawie dla liczb typu double

Asembler koprocatora x87 realizuje operację logarytmu przy podstawie 2 instrukcją o mnemoniku fyl2x. Jednak podobnie jak w przypadku funkcji wykładniczej, koprocesor nie posiada dedykowanej instrukcji dla obliczenia logarytmu o dowolnej podstawie. Z pomocą przychodzi tożsamość matematyczna, określana jako *zamiana podstawy logarytmu* i dana wzorem:

$$\log_b c = \frac{\log_a c}{\log_a b} \quad (4)$$

gdzie w przypadku koprocatora x87 $a = 2$.

Zadanie. Napisać wstawkę asemblerową obliczającą wartość funkcji:

$$y = \log_{10} (|1 + x|)$$

```

1      fld1                 ;// 1
2      fld1                 ;// 1
3      mov eax, 10          ;// immediate value cannot be passed
                           ;// directly to the FPU stack
4      push eax             ;// so we use CPU registers and
                           ;// memory stack
5      fld dword ptr [esp]  ;// 10, 1, 1
6      pop eax              ;// clear the CPU stack

```

```

10      fyl2x                ;// log_2(10), 1
11      fdiv                 ;// 1/log_2(10)
12
13      fld1                 ;// 1, 1/log_2(10)
14      fld x                ;// x, 1, 1/log_2(10)
15      fadd                 ;// 1+x, 1/log_2(10)
16      fabs                 ;// |1+x|, 1/log_2(10)
17      fyl2x                ;// 1/log_2(10) * log_2|1+x|
18
19      fstp yAsm            ;// <empty>

```

3.3 Instrukcje warunkowe dla liczb typu double

Koprocesor x87 nie posiada dedykowanych instrukcji, które mogą realizować sprawdzanie warunków i wykonywanie skoków. To jest zadanie dla głównej jednostki obliczeniowej — procesora. Zbiór instrukcji x87 posiada jednak dedykowane instrukcje do analizowania zależności pomiędzy dwiema liczbami. Są to m.in. instrukcje `fcom` oraz `fcomi`. Instrukcja `fcom` wykonuje porównanie dwóch liczb i następnie ustawia znaczniki C3, C2 i C0 koprocesora. Instrukcja `fcomi` wykonuje to samo sprawdzenie, jednak zamiast znaczników FPU ustawia flagi procesora: ZF, PF oraz CF. Po wykonaniu sprawdzenia możliwe jest wykonanie skoku korzystając z instrukcji skoków warunkowych sprawdzających flagi (`jb`, `jp`, `jc`, itp.) lub **dla liczb bez znaku** (`ja`, `jb`, itp.).

Zadanie. Napisać wstawkę assemblerową obliczającą wartość funkcji:

$$y = \begin{cases} ax^2 & x \leq -2 \\ \sin^2\left(\frac{\pi x}{180}\right) & -2 < x \leq 0 \\ a\log_2(x) & 0 < x \leq 2 \\ \sqrt{16x\pi} & x > 2 \end{cases}$$

```

1      fld x                ;// x
2      mov eax, -2
3      call _fld_eax        ;// -2, x
4      fcomip st, st(1)     ;// compare st againsts st(1) and
5                          ;// set EFLAGS pop st
6      ja _case1            ;// x <= -2
7      mov eax, 0
8      call _fld_eax        ;// 0, x
9      fcomip st, st(1)     ;// compare st againsts st(1) and
10     ja _case2             ;// x <= 0
11     mov eax, 2
12     call _fld_eax        ;// 2, x
13     fcomip st, st(1)     ;// compare st againsts st(1) and
14     ja _case3             ;// x <= 2
15
16 _default:
17     mov eax, 16
18     call _fld_eax        ;// 16, x
19     fldpi                ;// pi, 16, x

```

```

20     fmul                ;// 16*pi, x
21     fmul                ;// 16*pi*x
22     fsqrt               ;// sqrt(16*pi*x)
23     jmp _end
24
25 _case1:
26     fld st              ;// x, x
27     fmul                ;// xx
28     fmul a              ;// axx
29     jmp _end
30
31 _case2:
32     fldpi               ;// pi, x
33     fmul                ;// pi*x
34     mov eax, 180
35     call _fld_eax       ;// 180, pi*x
36     fdiv                ;// pi*x/180
37     fsin                ;// sin(pi*x/180)
38     fld st              ;// sin(pi*x/180), sin(pi*x/180)
39     fmul                ;// (sin(pi*x/180))^2
40     jmp _end
41
42 _case3:
43     fld a               ;// a, x
44     fxch st(1)          ;// x, a
45     fyl2x               ;// a*log_2(x)
46     jmp _end
47
48 _fld_eax:
49     push eax
50     fild dword ptr [esp];// eax
51     pop eax
52     ret
53
54 _end:
55     fstp yAsm           ;// <empty>

```

3.4 Tablicowanie funkcji dla liczb typu double

Obliczyć wektor $\mathbf{y} \in \mathbb{R}^n$ dla liczb typu double, którego wartości są wyrażone przez następującą funkcję:

$$y_i = ax_i^2 \log_2(|1 + x_i|) \quad x_i \in \langle x_{\min}, x_{\max} \rangle$$

Wartość kroku jest wyrażona jako:

$$s = \frac{x_{\max} - x_{\min}}{n}$$

Kolejne wartości x_i są wyznaczane z:

$$x_i = x_{i-1} + s \quad i = 1 \dots n$$

Należy zauważyć, że:

$$x_0 = x_{\min}$$

Liczby x_{\min} , x_{\max} oraz n stanowią argumenty (ich wartości należy wylosować lub pobrać od użytkownika).

```

1      mov esi, xAsm
2      mov edi, yAsm
3      xor ecx, ecx          ;// ecx = 0
4      mov eax, n           ;// eax = n
5      fld a                ;// a
6      fld xmin             ;// xmin , a
7      fld st               ;// xmin, xmin , a
8      fsubr xmax           ;// xmax-xmin, xmin , a
9      fidiv n              ;// (xmax-xmin)/n = step, xmin , a
10     fxch st(1)           ;// xmin, step , a
11     _loop:
12     fld st               ;// x, x, step , a
13     fld1                 ;// 1, x, x, step , a
14     fld st(1)            ;// x, 1, x, x, step , a
15     fmul st, st(2)        ;// xx, 1, x, x, step , a
16     fmul st, st(5)        ;// axx, 1, x, x, step , a
17     fxch st(2)           ;// x, 1, axx, x, step , a
18     fadd st, st(1)        ;// 1+x, 1, axx, x, step , a
19     fabs                 ;// |1+x|, 1, axx, x, step , a
20     fyl2x                ;// 1*log_2|1+x|, axx, x, step , a
21     fmul                 ;// axx*log_2|1+x|, x, step , a
22     fstp qword ptr [edi + 8 * ecx];// <save and pop y>
23     ;// x, step , a
24     fst qword ptr [esi + 8 * ecx] ;// <save x> x, step , a
25     fadd st, st(1)        ;// x = x+step, step , a
26     inc ecx              ;// ++i
27     cmp ecx, eax
28     jl _loop             ;// i < n
29     fstp st              ;// step, a
30     fstp st              ;// a
31     fstp st              ;// <empty>

```

4 Zadania do samodzielnego wykonania

1. Dla zmiennych a , b , c , d oraz x typu double oblicz wartość wyrażenia:

$$y = ax^b - bx^c + cx + d$$

2. Dla zmiennych a oraz x typu double oblicz wartość wyrażenia:

$$y = a \ln(1 + x^4) + 8\sqrt{ax + 6}$$

3. Dla zmiennych a , b , c oraz x typu double oblicz wartość wyrażenia:

$$y = \begin{cases} ax^2 + bx + c & x \leq -5 \\ \sin^2\left(\frac{\pi x}{180}\right) + \cos\left(\frac{\pi x}{180}\right)^2 & -5 < x \leq 0 \\ ax^2 \log_2(1+x) & 0 < x \leq 5 \\ \sqrt{16x\pi} - \sqrt{32x^2} & x > 5 \end{cases}$$

4. Obliczyć wektor $\mathbf{y} \in \mathbb{R}^n$ dla liczb typu double, którego wartości są wyrażone przez następującą funkcję:

$$\mathbf{y} = a \sin\left(\frac{\pi x}{180}\right)^2 + b \cos\left(\frac{\pi x}{180}\right)^2 \quad x \in \langle x_{\min}, x_{\max} \rangle$$

Wartość kroku jest wyrażona jako:

$$s = \frac{x_{\max} - x_{\min}}{n}$$

Kolejne wartości x_i są wyznaczane z:

$$x_i = x_{i-1} + s \quad i = 1 \dots n$$

Należy zauważyć, że:

$$x_0 = x_{\min}$$

Liczby x_{\min} , x_{\max} oraz n stanowią argumenty (ich wartości należy wylosować lub pobrać od użytkownika).