Wrocław, 21.05.2018

Adam Karczewski, 235039
WT-N-7

prowadzący: Aleksandra Postawka

Laboratorium Architektury Komputerów

(5) Jednostka zmiennoprzecinkowa- FPU

5.1 Treść ćwiczenia

Zakres i program ćwiczenia:
Celem ćwiczenia była nauka korzystania z wbudowanej jednostki zmiennoprzecinkowej

Zrealizowanie zadania:
Pierwsze zadanie polegało na napisaniu programu w języku C, którzy umożliwi sprawdzenie ustawionego trybu zaokrąglania i jego zmianę. Operacje te miały zostać wykonane z użyciem funkcji napisanej w języku assemblera.
Drugie zadanie polegało na napisaniu funkcji w języku assemblera, która obliczy przybliżenie funkcji e^x z wykorzystaniem szeregu Taylora.

5.2 Budowa kodu źródłowego

5.2.1 Sprawdzenie i zmiana trybu zaokrąglania

Program w języku C wywołuję funkcję `sprawdz` napisaną w języku assemblera. Pobierane jest słowo kontrolne jednostki FPU, które zostaje zapisane do zmiennej `control_word`, a następnie skopiowane do wyzerowanego rejstru `ax`. Wykonując operacje na jednostce zmiennoprzecinkowej należy użyć dodatkowego polecenia `fwait`- oczekiwanie z wykonaniem kolejnego rozkazu na zakończenie operacji zmiennoprzecinkowych, jest to warunkiem otrzymania poprawnego wyniku.

```
movq $0, %rax
fstcw  control_word
fwait
mov  control_word, %ax
```

O trybie zaokrąglania informują nas 10 i 11 bit słowo kontrolnego, dlatego nakładamy odpowiednią maskę logiczną i przesuwamy odpowiednio wynik iloczynu logicznego w prawo.

```
and $0xC00, %ax
shr $10, %ax
```

Zwracany wynik znajduje się już w rejestrze `rax`. Program interpretuje zwracany wynik w następujący sposób:

0- round to nearest
1- round down
2- round up
3- truncate

Aby zmienić tryb zaokrąglania funkcja `ustaw` pobiera argument zakodowany tak jak przedstawiono wyżej. Cyfra 0-3 znajduje się w rejestrze `rdi`. Pobierane jest słowo kontrolne FPU tak jak pokazano wcześniej. Następnie za pomocą odpowiedniej maski zerujemy bity 10 i 11, przesuwamy wybrany tryb zaokrąglania o 10 pozycji w lewo i dokonujemy operacji xor. W ten sposób zmienimy tylko bity trybu zaokrąglania, pozostałe informacje zawarte w słowie kontrolnym pozostaną nienaruszone.

```
and $0xF3FF, %ax
shl $10, %di
xor %di, %ax
```

Ostatnim etapem jest przesłanie zmienianego słowo kontrolnego.

```
mov %ax,  control_word
fldcw control_word
```

5.2.2 Szereg Taylora

Wzór na funkcji e^x z wykorzystaniem szeregu Taylora

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots, \ -\infty < x < \infty$$

Program pobiera od użytkownika wykładnik x i liczbę wyrazów szeregu. Zapisane są one odpowiednio w rejestrach `rdi` i `xmm0`. Funkcja będzie korzystała ze zmiennych globalnych, dlatego należy je najpierw zadeklarować i odpowiednio zainicjować.

```
.data
szereg: .double 1.0
potega: .double 1.0
silnia: .double 1.0
x: .double 0.0
licznik: .double 0.0
```

Rejestr `xmm0` nie jest bezpośrednio połączony z rejestrami zmiennoprzecinkowymi, dlatego należy odpowiednio przysłać dane pomiędzy jednostkami. Dokonujemy tego z użyciem stosu. Wartość umieszczamy na stosie, a następnie pobieramy jego szczyt do pierwszego rejestru zmiennoprzecinkowego- `st(0)` , po czym zapisujemy jego wartość do zmiennej x.

```
sub $8, %rsp
movsd %xmm0, (%rsp)
fldl (%rsp)
fstpl x
```

Następnie w pętli obliczmy kolejne wyrazy szeregu. Najpierw generujemy kolejną potęgę x.

```
fldl potega
fmull x
fstpl potega
```

W kolejnym kroku obliczamy silnię. W tym celu umieszczamy 1 w rejestrze `st(0)` i dodajemy jego zawartość do zmiennej licznik, po czym mnożymy licznik razy poprzedni wyraz silni i zapisujemy wynik w pamięci.

```
fld1
faddl licznik
fstpl licznik
fldl licznik
fmull silnia
fstpl silnia
```

Ostatnim etapem jest wytworzenie wyrazu szeregu, po przez podzielenie wartości zapisanej w zmiennej potęga przez wartość silni oraz dodanie tego iloczynu do sumy obliczonych wcześniej wyrazów.

```
fldl potega
fdivl silnia
faddl szereg
fstpl szereg
```

Zwracany wynik musi być w rejestrze `xmm0`, dlatego tak jak poprzednio dokonujemy kopiowania wartość po przez stos.

```
fldl szereg
fstpl  (%rsp)
movsd  (%rsp), %xmm0
```

5.3 Wnioski

Na zajęciach nauczyłem się wykorzystywać i sterować jednostką zmiennoprzecinkową- FPU.