

Architektura komputerów 2 – Projekt

Sprawozdanie

Mateusz Chalik 252735

Prowadzący: Dr inż. Dominik Żelazny

Termin zajęć: TN czwartek 17:05

Kod grupy: E07-01e

1. Temat

Tematem projektu jest kalkulator liczb zmiennoprzecinkowych wzbogacony o ciekawe funkcjonalności:

- 1) Najprostszymi funkcjonalnościami są działania algebraiczne: dodawanie, odejmowanie, mnożenie oraz dzielenie.
- 2) Kalkulator oblicza funkcje trygonometryczne: sinus, cosinus, tangens i cotangens.
- 3) Dostępne jest obliczanie pierwiastka oraz potęgi danej liczby.
- 4) Można również rozwiązać równanie kwadratowe.
- 5) Dodatkowo kalkulator posiada możliwość wyliczenia podanego wyrazu ciągu Fibonacciego.
- 6) Dla każdej z powyższych funkcji wybieramy jeden z czterech dostępnych trybów zaokrąglania otrzymanego wyniku.

2. Wstęp

Kalkulator w głównej mierze opiera się na liczbach zmiennoprzecinkowych 64-bitowych. Większość działań jest wykonywanych za pomocą FPU – układu scalonego wspomagającego procesor w obliczeniach głównie zmiennoprzecinkowych, ale również na liczbach całkowitych.

W projekcie połączyłem plik C z plikiem asemblera. Interfejs użytkownika oraz komunikaty znajdują się w pliku .c a funkcje wykonujące podane działania oraz wszystkie obliczenia zawierają się w pliku .s.

Program w części asemblera jest napisany w składni AT&T, w formacie 32-bitowym.

3. Opis i działanie kodu

3.1 Dodawanie

Po załadowaniu dwóch liczb do ST(0) oraz ST(1) wywoływana jest instrukcja FADDP. FADDP – wykonaj dodawanie ST(0) i ST(1), zapisz wynik w ST(1) i wykonaj pop ST(0).

```
Wpisz a: 1.000111000111
Wpisz b: 2.3335556661234
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Suma: 3.333667
```

3.2 Odejmowanie

Po załadowaniu dwóch liczb do ST(0) oraz ST(1) wywoływana jest instrukcja FSUBP. FSUBP – wykonaj odejmowanie ST(0) od ST(1), zapisz wynik w ST(1) i wykonaj pop ST(0).

```
Wpisz a: 1.23456789
Wpisz b: 0.123456
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 3
Roznica: 1.111111
```

3.3 Mnożenie

Po załadowaniu dwóch liczb do ST(0) oraz ST(1) wywoływana jest instrukcja FMULP. FMULP – wykonaj mnożenie ST(1) przez ST(0), zapisz wynik w ST(1) i wykonaj pop ST(0).

```
Wpisz a: 4321.333333333333
Wpisz b: 33.33
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 2
Iloczyn: 144030.040000
```

3.4 Dzielenie

Po załadowaniu dwóch liczb do ST(0) oraz ST(1) wywoływana jest instrukcja FDIVP. FDIVP – wykonaj dzielenie ST(1) przez ST(0), zapisz wynik w ST(1) i wykonaj pop ST(0).

```
Wpisz a: 1010.99775533
Wpisz b: 20.5
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 1
Iloraz: 49.316963
```

3.5 Pierwiastek

Po załadowaniu liczby do ST(0) wywoływana jest instrukcja FSQRT.

FSQRT – oblicz pierwiastek kwadratowy liczby w ST(0) i zapisz wynik w ST(0).

```
Wpisz a: 81.818181
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Pierwiastek: 9.045340
```

3.6 Sinus

Po załadowaniu liczby do ST(0) wywoływana jest instrukcja FSIN.

FSIN – zamienia liczbę znajdującą się w ST(0) na jej sinus.

```
Wpisz a: (liczba a nie kat) 3.14
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Sinus: 0.001593
```

```
Wpisz a: (liczba a nie kat) 3.141592
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Sinus: 0.000001
```

3.7 Cosinus

Po załadowaniu liczby do ST(0) wywoływana jest instrukcja FCOS.

FCOS – zamienia liczbę znajdującą się w ST(0) na jej cosinus.

```
Wpisz a: (liczba a nie kat) 3.14
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 3
Cosinus: -0.999999
```

```
Wpisz a: (liczba a nie kat) 6.283
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 3
Cosinus: 0.999999
```

3.8 Tangens

Po załadowaniu liczby do ST(0) wywoływana jest instrukcja FPTAN.

FPTAN – zamienia liczbę znajdującą się w ST(0) na jej tangens i wykonuje push \$1.

```
Wpisz a: (liczba a nie kat) 3.1415
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 2
Tangens: -0.000092
```

```
Wpisz a: (liczba a nie kat) 123
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 2
Tangens: 0.517928
```

3.9 Cotangens

Najpierw obliczany jest tangens danej liczby, następnie korzystając z właściwości funkcji FPTAN dzielimy 1 przez tangens.

```
Wpisz a: (liczba a nie kat) 2
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Cotangens: -0.457658
```

```
Wpisz a: (liczba a nie kat)3.141592
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Cotangens: -1530011.652437
```

3.10 Potęga

Wykładnik ma postać liczby całkowitej dlatego w pętli mnożona jest podstawa przez samą siebie a wynik jest zapisywany do tymczasowej zmiennej niezainicjalizowanej w sekcji .bss.

```
a^b
Wpisz a: 5
Wpisz b: (calkowite wieksze od 0) 5
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 4
Potega: 3125.000000
```

```
a^b
Wpisz a: 10.8765
Wpisz b: (calkowite wieksze od 0) 0
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 1
Potega: 1.000000
```

3.11 Równanie kwadratowe

Pętle oraz instrukcje warunkowe są wykonane w pliku .c. Do obliczeń używane są funkcje wykonane w asemblerze wywoływane z poziomu C. Sprawia to że sterowanie funkcją jest teoretycznie w C, lecz wszystkie obliczenia wykonywane są z poziomu asemblera. Funkcja oblicza deltę oraz na jej podstawie sprawdza, czy równanie ma 2 rozwiązania, jedno lub brak.

```
ax^2 + bx + c
Wpisz a: 1
Wpisz b: 9
Wpisz c: 1
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 1
Pierwiastki rownania kwadratowego: -0.112517 -8.887482
```

```
ax^2 + bx + c
Wpisz a: 2
Wpisz b: -4
Wpisz c: 2
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 2
Pierwiastek rownania kwadratowego: 1.000000
```

```
ax^2 + bx + c
Wpisz a: 1
Wpisz b: 4
Wpisz c: 16
Zaokraglanie: 1-cut 2-up 3-down 4-nearest 3
Brak rozwiazan
```

3.12 Ciąg Fibonacciego

Funkcja jest napisana w C. W funkcji została użyta rekurencja. Z definicji ciągu Fibonacciego pierwszy i drugi wyraz mają wartość 1. Kolejne wyrazy są sumą dwóch poprzednich wyrazów. W funkcji wybieramy wyraz ciągu i otrzymujemy jego wartość. Aby zatrzymać funkcję należy wpisać liczbę ujemną.

```
1 wyraz ciagu = 1
Wpisz n: (zakoncz dla liczby ujemnej) 2
2 wyraz ciagu = 1
Wpisz n: (zakoncz dla liczby ujemnej) 3
3 wyraz ciagu = 2
Wpisz n: (zakoncz dla liczby ujemnej) 4
4 wyraz ciagu = 3
Wpisz n: (zakoncz dla liczby ujemnej) 5
5 wyraz ciagu = 5
Wpisz n: (zakoncz dla liczby ujemnej) 6
6 wyraz ciagu = 8
Wpisz n: (zakoncz dla liczby ujemnej) 7
7 wyraz ciagu = 13
Wpisz n: (zakoncz dla liczby ujemnej) 8
8 wyraz ciagu = 21
Wpisz n: (zakoncz dla liczby ujemnej) 9
9 wyraz ciagu = 34
Wpisz n: (zakoncz dla liczby ujemnej) 10
10 wyraz ciagu = 55
Wpisz n: (zakoncz dla liczby ujemnej) 20
20 wyraz ciagu = 6765
```

3.13 Zaokrąglenia

```
.data
round_cut: .short 0xC3F
round_up: .short 0x83F
round_down: .short 0x43F
round_nearest: .short 0x03F
```

Wyniki działań są zaokrąglane poprzez załadowanie powyższych wartości do słowa kontrolnego FPU. Zaokrąglanie jest możliwe dla wszystkich funkcji w kalkulatorze oprócz ciągu Fibonacciego.

4. Podsumowanie

Korzystanie z FPU ułatwiło realizację projektu. Z poziomu C przekazywane są do funkcji w asemblerze liczby typu double (64-bitowe). Możliwe jest również przekazywanie liczb typu long double – wtedy mają one długość 80 bitów.

Makefile:

```
all: calculator

calculator: calculator.c
    gcc -m32 calculator.c calc.s -g -o program

clean:
    rm program
```

Uruchamianie:

- 1) Otwieramy terminal w miejscu, w którym znajdują się pliki
- 2) Używamy komendy *make* aby zbudować projekt
- 3) Dalej wpisujemy *./program* aby uruchomić program
- 4) Korzystając ze wskazówek wybieramy interesujące nas opcje