Maciej Kaźmierczyk

Lab 2 - Liczby zmiennoprzecinkowe

1. **Cel powstania wyjątków:** Wyjątki zostały zdefiniowane, aby zminimalizować dla użytkowników komplikacje wynikające z wyjątkowych warunków. System arytmetyczny ma działać na obliczeniach tak długo, jak to możliwe, obsługując nietypowe sytuacje z rozsądnymi domyślnie zdefiniowanymi reakcjami.

2. **Działania do generowania wyjątków:** Wybrane wyjątki zostały wygenerowane w przykładowy sposób, opierając się o podstawowe założenia matematyczne.

3. **Kluczowy kod:**

```
FINIT # inicjowanie jednostki FPU

# +nieskonczonosc: dzielenie liczby dodatniej przez 0
FLDS dodatnia
FDIV zero

# -nieskonczonosc: dzielenie liczby ujemnej przez 0
FLDS ujemna
FDIV zero

# +zero: mnozenie zera przez liczbe dodatnia
FLDS zero
FMUL dodatnia

# -zero: mnozenie zera przez liczbe ujemna
FLDS zero
FMUL ujemna

# NaN: pierwiastek z liczby ujemnej
FLDS ujemna
FSQRT
```

4. **Wyniki debuggera:**

```
(gdb) info float
  R7: Special 0x7fff8000000000000000 +Inf
  R6: Special 0xffff8000000000000000 -Inf
  R5: Zero    0x00000000000000000000 +0
  R4: Zero    0x80000000000000000000 -0
=>R3: Special 0xffffc000000000000000 Real Indefinite (QNaN)
  R2: Empty   0x00000000000000000000
  R1: Empty   0x00000000000000000000
  R0: Empty   0x00000000000000000000
```

5. **Opis**

Na początku programu, w sekcji *data*, deklarowane są 3 liczby: dodatnia, ujemna i zero, do przeprowadzania na nich operacji arytmetycznych.

Następnie, w sekcji *text* program rozpoczyna pracę od zainicjalizowania w procesorze jednostki *FPU (Floating point unit)*. Po czym wykonywane jest 5 operacji w celu otrzymania poszczególnych wyjątków:

a) +Nieskończoność: Na stos przenoszona jest liczba dodatnia, która następnie, z wykorzystaniem instrukcji *FDIV* dzielona jest przez 0. Wynik tej operacji znajduje się w 7 rejestrze, oznaczony jako *Special*, i wskazuje *+Inf*.

b) -Nieskończoność: Powtarzana jest operacja z poprzedniego punktu, z wyjątkiem zamiany liczby dodatniej na ujemną, dzięki czemu widoczny w 6 rejestrze wynik to *-Inf*.

c) +Zero: W celu otrzymania zera dodatniego, po ówczesnym dodaniu go na stos, jest ono mnożone przy pomocy polecenia *FMUL*, przez liczbę dodatnią, w wyniku czego otrzymujemy, zapisane w 5 rejestrze *+0*.

d) -Zero: Operacja z poprzedniego punktu jest powtarzana, z wyjątkiem zamiany liczby dodatniej na ujemną, w wyniku czego otrzymujemy, zapisane w 4 rejestrze *-0*.

e) NaN: W celu otrzymania wartości *Not a Number*, wykonywana jest operacja pierwiastkowania liczby ujemnej, z wykorzystaniem polecenia *FSQRT*, w wyniku czego w 3 rejestrze, również oznaczonym jako *Special*, znajduje się wartość *QNaN*.

6. **GDB**

W celu zdebugowania programu:

a) Został on uruchomiony w debugerze za pomocą: `gdb ./wyjatki.s`
b) Następnie na pierwszej linii dodany został breakpoint: `b 1`
c) Program został wystartowany: `r`
d) Następnie korzystając z polecenia: `n`, w celu przejścia linijka po linijce, oraz polecenia: `info float`, wyświetlony został wynik rejestrów przed, w trakcie, i po operacjach