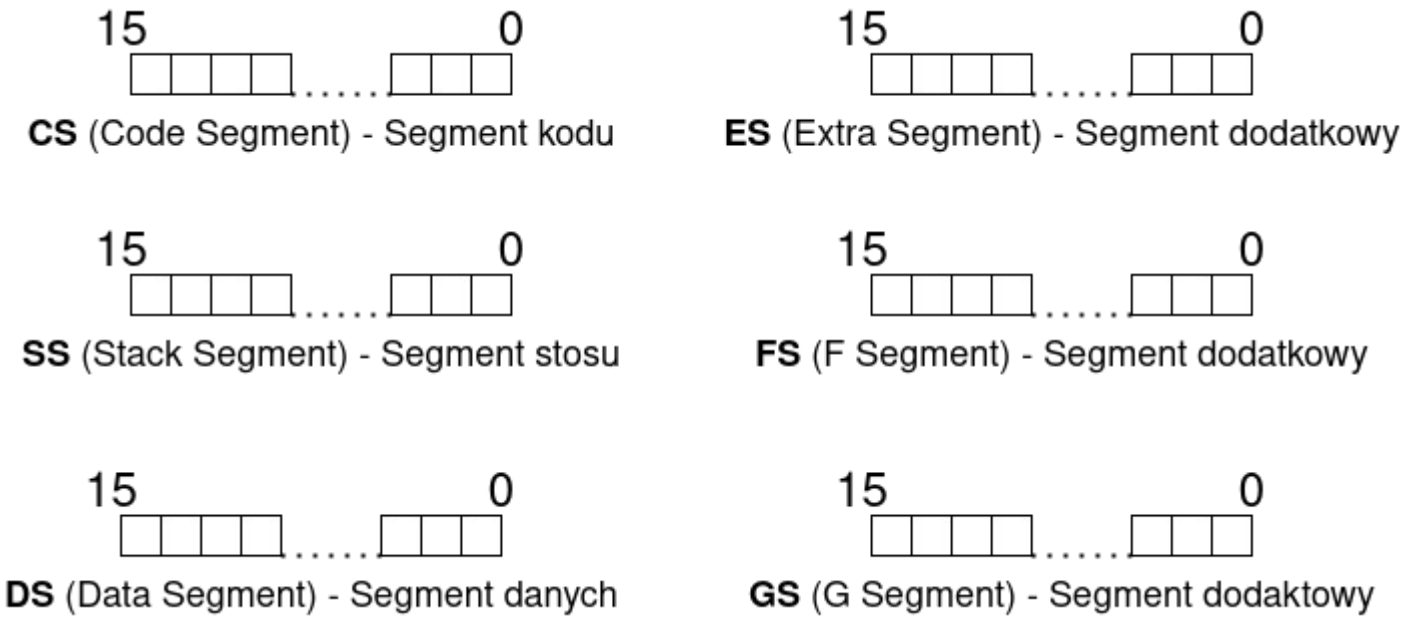
Rejestry Segmentowe

Przeznaczenie konkretnych rejestrów segmentowych jest zależne od systemu operacyjnego. Niezależnie od procesora mają one rozmiar 16-bitów (rysunek 2.2.13).

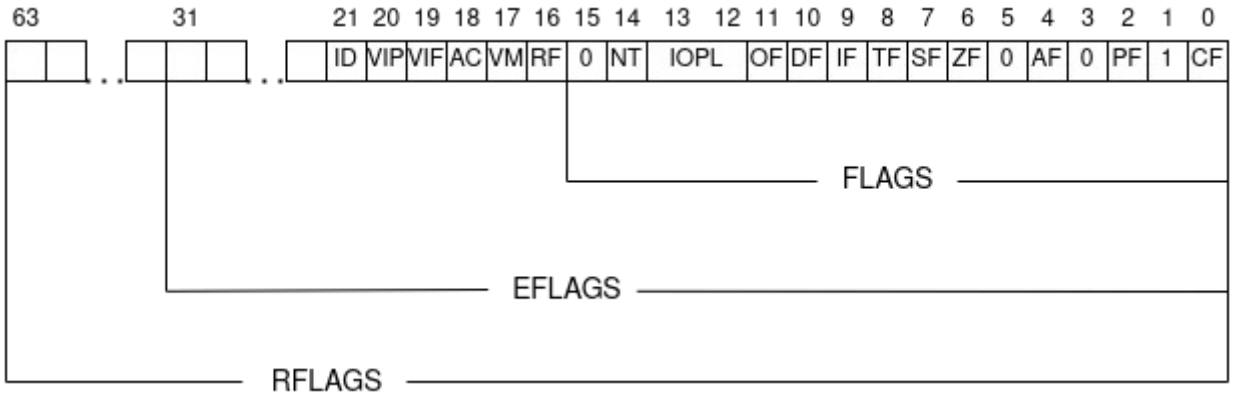


Rysunek 2.2.13. Rejestry segmentowe x86-64

Ważna kwestia dotycząca działania rejestrów segmentowych jest związana z wykorzystywanym modelem pamięci. Do dokładniejszego opisu i wizualizacji tych modeli przejdziemy w rozdziale **Pamięć Operacyjna**, tam też wyjaśnię wszystkie zawiłości związane z rejestrami segmentowymi.

Rejestr Flag

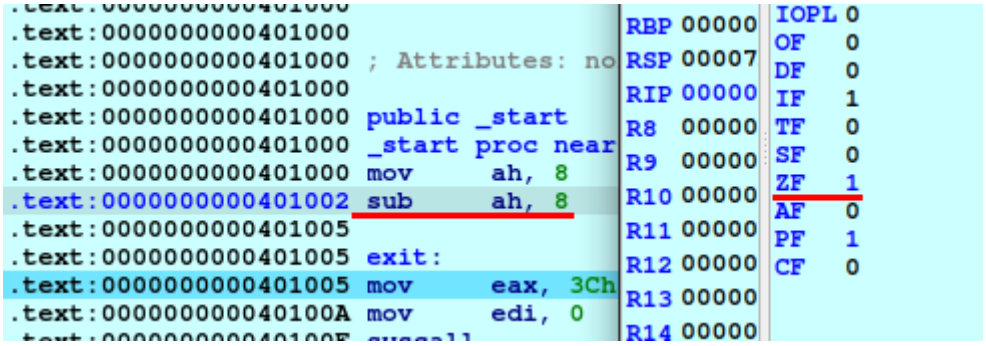
Rejestr flag przechowuje sporo informacji na temat ostatniej wykonanej operacji arytmetycznej. Poszczególne flagi mają rozmiar jednego bita, analogicznie, mogą przyjąć wartość **0** lub **1**.
Cały rejestr flag jest przedstawiony na rysunku 2.2.14. Rejestr RFLAGS jest dostępny tylko w trybie 64-bitowym.
Patrząc na rysunek 2.2.14. możesz zastanawiać się czym są najmłodsze bity: 15, 5, 3 i 1. Są to pewne archaizmy z procesorów pentium, obecnie te flagi stanowią tylko numer w rejestrze, nie są dłużej wykorzystywane. Są tam tylko przez wzgląd na kompatybilność wsteczną.



Rysunek 2.2.14. Rejestr flag x86-64

Opiszę tutaj i przedstawię na przykładzie najczęściej używane flagi, czyli: **ZF**, **CF**, **SF**, **PF**, **OF**. Wszystkie przedstawione na zrzutach ekranów przykłady pochodzą z debuggera IDA.

ZF (Zero Flag) Flaga Zerowa – jest ustawiana, gdy wynik ostatniej operacji arytmetycznej lub logicznej, jest równy zero (rysunek 2.2.15.).



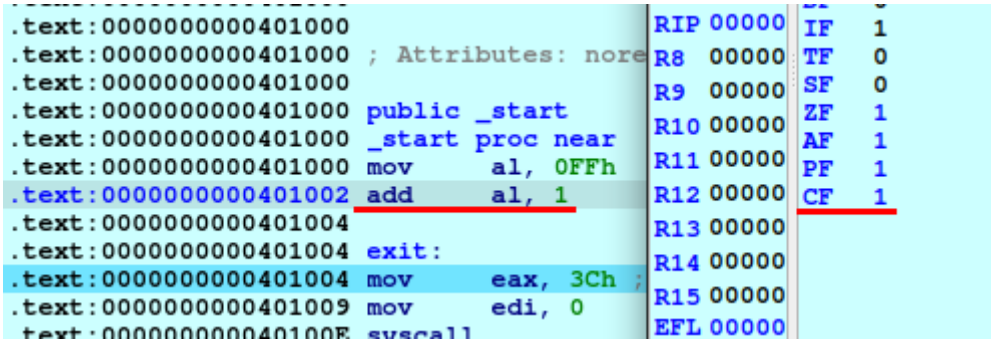
Rysunek 2.2.15. Przykład Zero Flag

CF (Carry Flag) Flaga Przeniesienia – jest ustawiana, gdy w wyniku ostatniej operacji arytmetycznej pojawiło się przeniesienie na najbardziej znaczący bit lub też pożyczenie z bitu wyniku operacji. Czyli **CF**, wskazuje na to, że wynik operacji zawiera się w większej, niż dostępna liczba bitów w rejestrze.

Scenariusz 1: Przeniesienie najbardziej znaczącego bitu (rysunek 2.2.16).

Przykład (dla 8-bitowego rejestru AL):

```
11111111 + 00000001 = 00000000
```

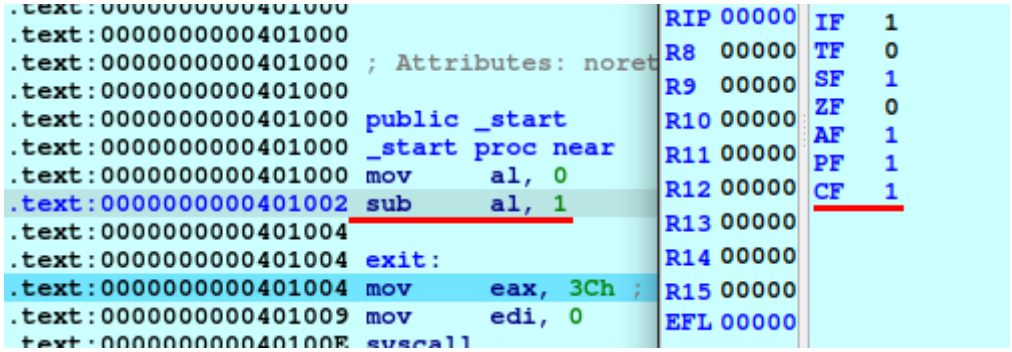


Rysunek 2.2.16. Przykład scenariusz 1 Carry Flag

Scenariusz 2: Pożyczenie najbardziej znaczącego bitu przy odejmowaniu dwóch liczb (rysunek 2.2.17).

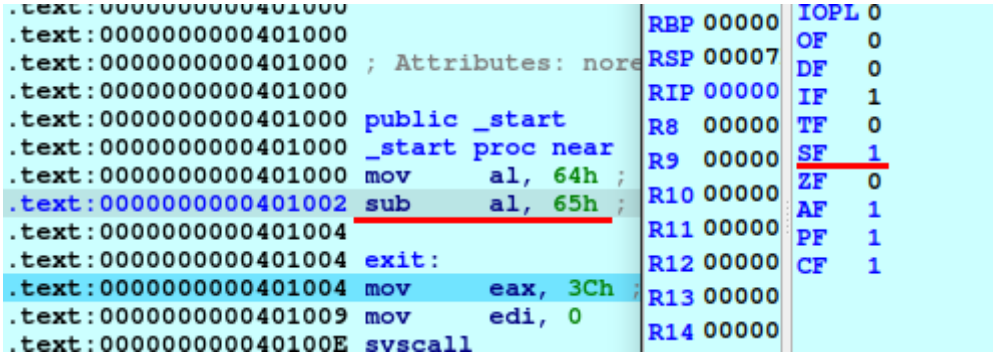
Przykład (dla 8-bitowego rejestru AL):

```
0000 0000 - 0000 0001 = 1111
```



Rysunek 2.2.17. Przykład scenariusz 2 Carry Flag

SF (Sign Flag) Flaga Znaku – jest ustawiana względem najstarszego bitu (bitu znaku). Czyli jeśli wynik ostatniej operacji arytmetycznej jest ujemny (rysunek 2.2.18).

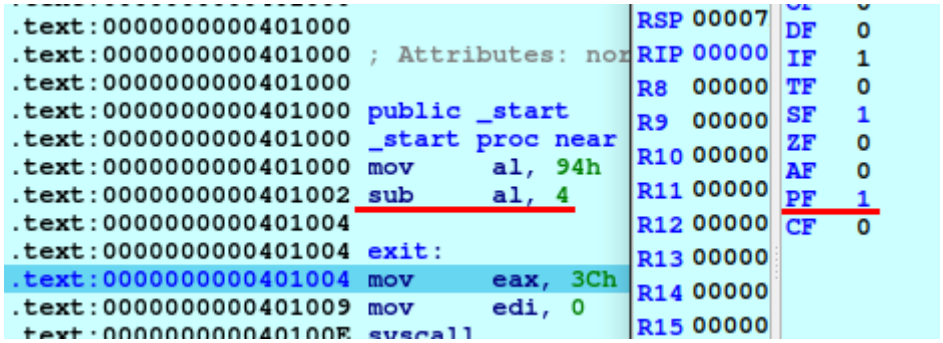


Rysunek 2.2.18. Przykład Sign Flag

PF (Parity Flag) Flaga Parzystości – jest ustawiana jeżeli najmniej znaczący bajt wyniku zawiera parzystą liczbę bitów (rysunek 2.2.19).

Przykład (dla 8-bitowego rejestru AL):

$$94_{16} - 4_{16} = 90_{16}$$
$$1001\ 0100 - 0100 = 1001\ 0000$$



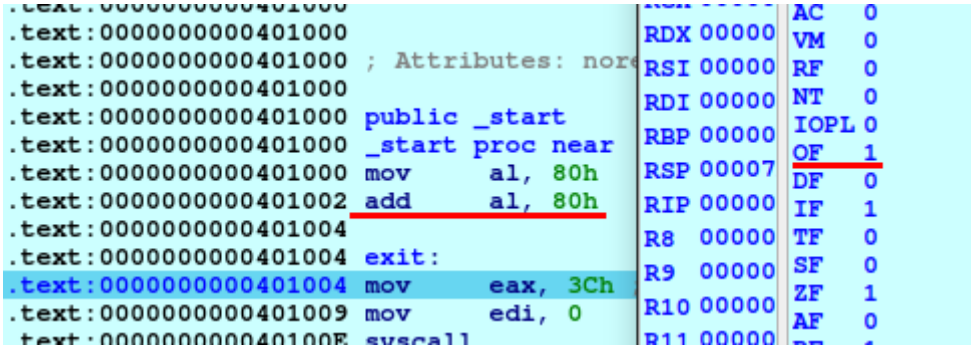
Rysunek 2.2.19. Przykład Parity Flag

OF (Overflow Flag) Flaga Przepełnienia – jest ustawiana w wyniku przepełnienia, gdy wynik ostatniej operacji, jest **zbyt dużą** liczbą dodatnią lub **zbyt małą** liczbą ujemną, aby zmieścić ją w rejestrze docelowym (rysunek 2.2.20).

Przykład (dla 8-bitowego rejestru AL):

$$0100\ 0000 + 0100\ 0000 = 1000\ 0000$$
$$80_{16} + 80_{16} = 0$$

```
1000 0000 + 1000 0000 = 0000 0000
```



Rysunek 2.2.20. Przykład Overflow Flag

Pozostają do omówienia skoki warunkowe, ale do nich przejdziemy w rozdziale **Podstawowe instrukcje**.

Jeśli masz jakieś pytania, to pisz śmiało na naszym [serwerze discord](https://discord.gg/XFazfUXvCx) https://discord.gg/XFazfUXvCx.