TESTBENCH

**Autor:** Konrad Krupski

**Numer indeksu:** 310729

**Kontakt:**

Email: [00158726@pw.edu.pl](mailto:00158726@pw.edu.pl)

MS Teams: Konrad Krupski (STUD)

**Data:** 28.11.2021

**Platforma sprzętowa:**

Działanie układu zostało przeprowadzone na systemie Fedora 35 Workstation 64 bit

**Sposób uruchamiania**

W celu przeprowadzenia samodzielnej syntezy należy uruchomić terminal w katalogu /WORK oraz wprowadzić polecenie  
make rtl

W celu samodzielnego przeprowadzenia testu działania należy wpisać polecenie

make sim

W celu wyświetlenia przebiegów sygnałów w programie gtkwave należy wpisać polecenie

make wave

Contents

[Sprawdzenie działania operacji 4](#_Toc90834472)

[Operacja dodawania i\_oper = 0000 4](#_Toc90834473)

[Sygnał 1 4](#_Toc90834474)

[Sygnał 2 5](#_Toc90834475)

[Sygnał 3 5](#_Toc90834476)

[Sygnał 4 5](#_Toc90834477)

[Komentarz 5](#_Toc90834478)

[Operacja NOR i\_oper = 0001 6](#_Toc90834479)

[Sygnał 1 6](#_Toc90834480)

[Sygnał 2 6](#_Toc90834481)

[Sygnał 3 7](#_Toc90834482)

[Sygnał 4 7](#_Toc90834483)

[Komentarz 7](#_Toc90834484)

[Operacja NAND i\_oper = 0010 7](#_Toc90834485)

[Sygnał 1 8](#_Toc90834486)

[Sygnał 2 8](#_Toc90834487)

[Sygnał 3 8](#_Toc90834488)

[Sygnał 4 8](#_Toc90834489)

[Komentarz 9](#_Toc90834490)

[Operacja onehot -> nkb i\_oper = 0111 9](#_Toc90834491)

[Sygnał 1 9](#_Toc90834492)

[Sygnał 2 9](#_Toc90834493)

[Sygnał 3 10](#_Toc90834494)

[Sygnał 4 10](#_Toc90834495)

[Komentarz 10](#_Toc90834496)

[Operacja przesunięcia bitowego w lewo i\_oper = 0011 10](#_Toc90834497)

[Sygnał 1 11](#_Toc90834498)

[Sygnał 2 11](#_Toc90834499)

[Sygnał 3 11](#_Toc90834500)

[Sygnał 4 11](#_Toc90834501)

[Komentarz 12](#_Toc90834502)

[Operacja przesunięcia arytmetycznego w prawo i\_oper = 0100 12](#_Toc90834503)

[Sygnał 1 12](#_Toc90834504)

[Sygnał 2 13](#_Toc90834505)

[Sygnał 3 13](#_Toc90834506)

[Sygnał 4 13](#_Toc90834507)

[Komentarz 13](#_Toc90834508)

[Operacja zamiany NKB -> Gray i\_oper=0101 14](#_Toc90834509)

[Sygnał 1 14](#_Toc90834510)

[Sygnał 2 15](#_Toc90834511)

[Sygnał 3 15](#_Toc90834512)

[Sygnał 4 15](#_Toc90834513)

[Komentarz 15](#_Toc90834514)

[Operacja zliczania zer i\_oper = 0110 16](#_Toc90834515)

[Sygnał 1 16](#_Toc90834516)

[Sygnał 2 16](#_Toc90834517)

[Sygnał 3 16](#_Toc90834518)

[Sygnał 4 17](#_Toc90834519)

[Komentarz 17](#_Toc90834520)

[Konwersja kod ZNAK-MODUŁ -> U2 i\_oper = 1011 17](#_Toc90834521)

[Sygnał 1 17](#_Toc90834522)

[Sygnał 2 18](#_Toc90834523)

[Sygnał 3 18](#_Toc90834524)

[Sygnał 4 18](#_Toc90834525)

[Komentarz 18](#_Toc90834526)

[Zamiana na kod CRC-4 bitowy i\_oper = 1000 19](#_Toc90834527)

[Sygnał 1 19](#_Toc90834528)

[Sygnał 2 19](#_Toc90834529)

[Sygnał 3 19](#_Toc90834530)

[Sygnał 4 20](#_Toc90834531)

[Komentarz 20](#_Toc90834532)

[Operacja sprawdzania kodu CRC-3 bitowego i\_oper = 1001 21](#_Toc90834533)

[Sygnał 1 21](#_Toc90834534)

[Sygnał 2 21](#_Toc90834535)

[Sygnał 3 21](#_Toc90834536)

[Sygnał 4 22](#_Toc90834537)

[Komentarz 22](#_Toc90834538)

[Sprawdzenie działania flag 22](#_Toc90834539)

[Sygnał 1 23](#_Toc90834540)

[Sygnał 3 23](#_Toc90834541)

[Sygnał 4 24](#_Toc90834542)

# Sprawdzenie działania operacji

Testy zostały przeprowadzone dla liczb 4 bitowych, gdyż tak początkowo zaprojektowano układ. Niemniej jednak w celu przeprowadzenia testu dla większej bitów należy zmienić wartość zmiennej:

M = 8

w pliku /WORK/testbench.sv.

Dodatkowo flagi zapisane są w tabelach jako jedna liczba binarna. Każda z cyfr odpowiada poszczególnej fladze np. liczba 0101 oznacza VF=0 PF=1 BF=0 OF=1.

## Operacja dodawania i\_oper = 0000

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 00000000 | 00000000 | 00000000 | 00000000 | 0000 | 0000 |
| 2 | 01100110 | 01001010 | 10110000 | 10110000 | 1100 | 1100 |
| 3 | 11001011 | 11101100 | 10100100 | 10100100 | 0100 | 0100 |
| 4 | 00000001 | 10100011 | 10100100 | 10100100 | 0100 | 0100 |

### Sygnał 1

**Graphical user interface, text, application

Description automatically generated**

### Sygnał 2

Graphical user interface, text, application

Description automatically generated

### Sygnał 3

Graphical user interface, text, application

Description automatically generated

### Sygnał 4

**Graphical user interface, text

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja dodawania realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 2.

o\_result = = 01100110 + 01001010 = 102 + 74 = 176 = 10110000

VF=1 - Oznacza to, że nastąpiło przepełnienie, ponieważ pracujemy na liczbach u2 więc zakres liczb [-127, 127].

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja NOR i\_oper = 0001

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 00000000 | 00111000 | 11000111 | 11000111 | 0100 | 0100 |
| 2 | 10010101 | 00100101 | 01001010 | 01001010 | 0100 | 0100 |
| 3 | 11110110 | 10110101 | 00001000 | 00001000 | 0110 | 0110 |
| 4 | 00100111 | 00111000 | 11000000 | 11000000 | 0000 | 0000 |

### Sygnał 1

Graphical user interface, text

Description automatically generated

### Sygnał 2

Graphical user interface, text, table

Description automatically generated

### Sygnał 3

Graphical user interface, text, table

Description automatically generated

### Sygnał 4

**Graphical user interface, text

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja NOR realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 3.

o\_result = 11110110 NOR 10110101= 00001000

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=1 – Oznacza to, że w wyniku jest tylko jedna jedynka

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja NAND i\_oper = 0010

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | falgi po syntezie |
| 1 | 10000110 | 01011100 | 11111011 | 11111011 | 0100 | 0100 |
| 2 | 00010111 | 10101001 | 11111110 | 11111110 | 0100 | 0100 |
| 3 | 00110111 | 10010111 | 11101000 | 11101000 | 0000 | 0000 |
| 4 | 10110011 | 11111110 | 01001101 | 01001101 | 0000 | 0000 |

### Sygnał 1

Graphical user interface, text, application, table

Description automatically generated

Sygnał 2

**Graphical user interface, text, application, table, Excel

Description automatically generated**

### Sygnał 3

**Graphical user interface, text, application

Description automatically generated**

### Sygnał 4

**Graphical user interface, text, application, table

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja NAND realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 3.

o\_result = 00110111 NAND 10010111 = 11101000

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=0 – Oznacza to, że liczba jedynek w wyniku jest parzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja onehot -> nkb i\_oper = 0111

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 00000101 | 11111111 | 11111111 | 0001 | 0001 |
| 2 | 00001000 | 00000011 | 00000011 | 0000 | 0000 |
| 3 | 00000010 | 00000001 | 00000001 | 0110 | 0110 |
| 4 | 11100001 | 11111111 | 11111111 | 0001 | 0001 |

### Sygnał 1

Text

Description automatically generated

## Sygnał 2

Graphical user interface, text, application

Description automatically generated

### Sygnał 3

Graphical user interface, text

Description automatically generated

### Sygnał 4

Graphical user interface, text

Description automatically generated

### Komentarz

Z wyników symulacji można stwierdzić, że operacja zamiany kodu one hot na nkb realizowana jest poprawnie. Rozważmy przykład numer 2

I\_argA = 00001000

Kod sprawdza na którym miejscu znajduje się jedynka. Jedynka znajduje się na miejscu czwartym pod indeksem numer 3. Dlatego powinna zostać zwrócona wartość 3 i tak też się dzieje.

o\_result = 00000011

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=0 – Oznacza to, że liczba jedynek w wyniku jest parzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja przesunięcia bitowego w lewo i\_oper = 0011

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 11001110 | 11000111 | 00000000 | 00000000 | 0000 | 0000 |
| 2 | 11101111 | 00101010 | 00000000 | 00000000 | 0000 | 1000 |
| 3 | 11100111 | 11000011 | 00000000 | 00000000 | 0000 | 1000 |
| 4 | 01011100 | 10011011 | 00000000 | 00000000 | 0000 | 0000 |

### Sygnał 1

Graphical user interface, text, application

Description automatically generated

### Sygnał 2

Graphical user interface, text, application

Description automatically generated

### Sygnał 3

**Graphical user interface, text, application

Description automatically generated**

### Sygnał 4

Graphical user interface, text

Description automatically generated

### Komentarz

Z wyników symulacji można stwierdzić, że operacja przesunięcia bitowego w lewo realizowana jest poprawnie. W tym wypadku przesuwamy liczby bitowo o bardzo duże wartości, więc wynik w każdym wypadku powinien być 0.

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=0 – Oznacza to, że liczba jedynek w wyniku jest parzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja przesunięcia arytmetycznego w prawo i\_oper = 0100

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 11000110 | 11110011 | 11111111 | 11111111 | 0001 | 0001 |
| 2 | 00110000 | 10101010 | 00000000 | 00000000 | 0000 | 0000 |
| 3 | 00110101 | 11010110 | 00000000 | 00000000 | 0000 | 0000 |
| 4 | 00111001 | 11010010 | 00000000 | 00000000 | 0000 | 0000 |

### Sygnał 1

**Graphical user interface, text

Description automatically generated**

### Sygnał 2

**Graphical user interface, text, table

Description automatically generated**

### Sygnał 3

**Graphical user interface, text, application

Description automatically generated**

### Sygnał 4

**Graphical user interface, text, application

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja przesunięcia arytmetycznego w prawo realizowana jest poprawnie. Dla sprawdzenia przeanalizuję sygnał 1.

11000110 >>> 11110011

A picture containing text

Description automatically generated

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=0 – Oznacza to, że liczba jedynek w wyniku jest parzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=1 – Oznacza to, że wynikiem operacji jest ‘1.

## Operacja zamiany NKB -> Gray i\_oper=0101

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | falgi po syntezie |
| 1 | 11000011 | 11111111 | 11111111 | 0001 | 0001 |
| 2 | 00000000 | 00000000 | 00000000 | 0000 | 0000 |
| 3 | 01000100 | 01100110 | 01100110 | 0000 | 0000 |
| 4 | 01110011 | 01001010 | 01001010 | 0100 | 0100 |

### Sygnał 1

Table

Description automatically generated

### Sygnał 2

**Graphical user interface, text

Description automatically generated**

### Sygnał 3

**Graphical user interface, application, table

Description automatically generated**

### Sygnał 4

**Graphical user interface, text, application, table

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja zamiany nkb->gray realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 4.

(NKB)01110011 -> Gray(01001010)

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja zliczania zer i\_oper = 0110

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 01000111 | 10001001 | 00001001 | 00001001 | 0000 | 0000 |
| 2 | 10100111 | 01101110 | 00000110 | 00000110 | 0000 | 0000 |
| 3 | 00000010 | 11001111 | 00001001 | 00001001 | 0000 | 0000 |
| 4 | 10111010 | 11100000 | 00001000 | 00001000 | 0110 | 0110 |

### Sygnał 1

**Graphical user interface, application, table

Description automatically generated**

### Sygnał 2

**Graphical user interface, text, application, table

Description automatically generated**

### Sygnał 3

**Graphical user interface, text, application

Description automatically generated**

### Sygnał 4

**Graphical user interface, text, application, table

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja zliczania zer realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 4.

10111010 – liczba zer: 3

11100000 – liczba zer: 5

o\_result = 00001000 = 8

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=1 – Oznacza to, że w wyniku jest jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Konwersja kod ZNAK-MODUŁ -> U2 i\_oper = 1011

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | falgi po syntezie |
| 1 | 11100011 | 10011101 | 10011101 | 0100 | 0100 |
| 2 | 11001011 | 10110101 | 10110101 | 0100 | 0100 |
| 3 | 10110000 | 11010000 | 11010000 | 0100 | 0100 |
| 4 | 10001000 | 11111000 | 11111000 | 0100 | 0100 |

### Sygnał 1

**Graphical user interface, text, table

Description automatically generated**

### Sygnał 2

**Graphical user interface, text, application

Description automatically generated**

### Sygnał 3

**Graphical user interface, text, application

Description automatically generated**

Sygnał 4

**Graphical user interface, text, table

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja zamiany kodu SM na U2 realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 3.

(sm)10110000->(u2) 11010000

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Zamiana na kod CRC-4 bitowy i\_oper = 1000

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | falgi po syntezie |
| 1 | 1101 | 0001 | 1101 | 1101 | 0100 | 0100 |
| 2 | 0100 | 1110 | 1000 | 1000 | 0110 | 0110 |
| 3 | 0010 | 1011 | 0110 | 0110 | 0000 | 0000 |
| 4 | 1100 | 0001 | 0110 | 0110 | 0000 | 0000 |

### Sygnał 1

**Graphical user interface, text

Description automatically generated with medium confidence**

### Sygnał 2

**Graphical user interface, text

Description automatically generated**

### Sygnał 3

**Graphical user interface

Description automatically generated with medium confidence**

### Sygnał 4

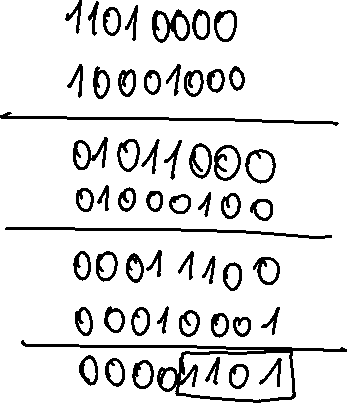
**Table

Description automatically generated**

### Komentarz

Z wyników symulacji można stwierdzić, że operacja konwersji na kod crc4-bitowy realizowana jest poprawnie. Operacja została przeprowadzono dla liczb 4 bitowych. Dla sprawdzenia można przeanalizować sygnał numer 1.

I\_argA = 1101 i i\_poly = 10001



VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

## Operacja sprawdzania kodu CRC-3 bitowego i\_oper = 1001

Wyjście układu exe\_unit jest 40-bitowe, więc mimo tego, że mamy do czynienia z kodem crc3 to w celu uniknięcia problemy z ilością bitów zdecydowałem się dopisywać 0 na początku o\_result.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | i\_argA | i\_argB | o\_result przed syntezą | o\_result po syntezie | flagi przed syntezą | flagi po syntezie |
| 1 | 1110 | 1111 | 0101 | 0101 | 0000 | 0000 |
| 2 | 0010 | 0100 | 0111 | 0111 | 0100 | 0100 |
| 3 | 1110 | 0101 | 0001 | 0001 | 0110 | 0110 |
| 4 | 0011 | 1110 | 0101 | 0101 | 0000 | 0000 |

### Sygnał 1

**Text

Description automatically generated with medium confidence**

### Sygnał 2

**Table

Description automatically generated**

Sygnał 3

**Graphical user interface

Description automatically generated**

### Sygnał 4

Graphical user interface

Description automatically generated with medium confidence

### Komentarz

Z wyników symulacji można stwierdzić, że operacja sprawdzania kodu crc3-bitowego realizowana jest poprawnie. Dla sprawdzenia można przeanalizować sygnał numer 3.

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

# Sprawdzenie działania flag

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp. | I\_oper | I\_argA | I\_argB | O\_result | Flagi przed syntezą | Falig po syntezie |
| 1 | 1000 | 01011101 | 10010001 | 00001101 | 0100 | 0100 |
| 2 | 1010 | Zostanie zwrócona | Wartość domyślna | 11111111 | 0001 | 0001 |
| 3 | 0000 | 01100110 | 01001010 | 10110000 | 1100 | 1100 |
| 4 | 0110 | 10111010 | 11100000 | 00001000 | 0110 | 0110 |

### Sygnał 1

Graphical user interface

Description automatically generated

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji nie jest ‘1.

Graphical user interface, text

Description automatically generated

VF=0 – Oznacza to, że nie nastąpiło przepełnienie.

PF=0 – Oznacza to, że liczba jedynek w wyniku jest parzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=1 – Oznacza to, że wynikiem operacji jest ‘1.

### Sygnał 3

Graphical user interface, text, application

Description automatically generated

VF=1 – Oznacza to, że nastąpiło przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=0 – Oznacza to, że w wyniku jest więcej niż jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji jest ‘1.

### Sygnał 4

**Graphical user interface, text, application, table

Description automatically generated**

VF=0 – Oznacza to, że nastąpiło nie przepełnienie.

PF=1 – Oznacza to, że liczba jedynek w wyniku jest nieparzysta.

BF=1 – Oznacza to, że w wyniku jest jedna jedynka.

OF=0 – Oznacza to, że wynikiem operacji jest ‘1.