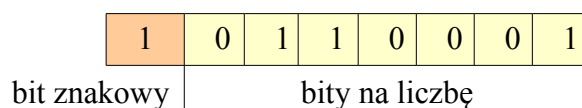


ARCHITEKTURA KOMPUTERÓW

Kodowanie liczb ze znakiem

27.10.2010

Do zapisu liczby ze znakiem mamy tylko 8 bitów, pierwszy od lewej bit to bit znakowy, a pozostałym 7 to bity na liczbę.



Znak modułu (ZM)

Na pierwszym bicie od lewej strony zapisujemy znak plus czyli 0 lub minus czyli 1.

$01111111 \rightarrow +127$

$11111111 \rightarrow -127$

$00000000 \rightarrow 0$

$10000000 \rightarrow NaN$

W przypadku gdy liczba będzie większa niż 7 bitów dodaje się kolejne bity aż do 16, następnie po przekroczeniu tego zakresu do 32 bitów.

$0000000011111111 \rightarrow +255$

$1000000011111111 \rightarrow -255$

Przykład zapisania liczb dziesiętnej ze znakiem na kod Znak Modułu

$+96_{DEC}$

$96_{DEC} = 1100000_{BIN}$

przeliczamy liczbę na system binarny
 jeżeli ilość bitów jest mniejsza niż 8 to dopisujemy tyle ZER
 przed liczbą aby zostało zajęte 7 bitów, a na 8 bicie
 wstawiamy znak w tym przypadku 0 bo liczba jest dodatnia

01100000_{ZM}

$+96_{DEC} = 01100000_{ZM}$

Przykład zapisania liczb ze znakiem metodą Znak Modułu

-159_{DEC}

$159_{DEC} = 10011111_{BIN}$

| przeliczamy liczbę na system binarny

1000000010011111_{ZM}

| liczba mieści się na 8 bitach więc dopisujemy ZERA przed
| liczbą aż do uzyskania 15 bitów, a na 16 bicie wstawiamy
| znak – czyli 1

$-159_{DEC} = 1000000010011111_{ZM}$

W przypadku zapisania kodu ZM na liczbę dziesiętną postępujemy odwrotnie.

Przykład zapisania liczby w kodzie Znak Modułu na liczbę dziesiętną

00110101_{ZM}

0 0110101

| odcinamy bit znaku

$0110101_{BIN} = 53_{DEC}$

| przeliczamy bity liczbowe na system dziesiętny

$+53_{DEC}$

| sprawdzamy bit znakowy który w tym przypadku wynosi 0
| czyli liczba jest dodatnia

$00110101_{ZM} = +53_{DEC}$

Koda spolaryzowany (SPOL)

Ustalamy wartość środkową jaka mieści się na 8 bitach i ona będzie zerem, wszystko bo będzie mniejsze od wartości środkowej będzie ujemne, a to co będzie większe będzie dodatnie.

$$11111111_{BIN} = 255_{DEC}$$

$$01111111_{BIN} = 127_{DEC} \quad \text{wartość środkowa która będzie zerem}$$

$$00000000_{BIN} = 0_{DEC}$$

czyli:

$$11111111_{BIN} = +127_{DEC}$$

$$01111111_{BIN} = 0_{DEC} \quad \text{wartość środkowa}$$

$$00000000_{BIN} = -127_{DEC}$$

W kodzie spolaryzowanym znak + oznaczamy za pomocą 1, a znak – za pomocą 0.

W przypadku liczby mieszczącej się na mniejszej ilości bitów niż 7 dopisujemy zera do uzyskania 7 bitowej liczby, a na 8 bit wstawiamy znak.

Przykład zapisania liczby dziesiętnej ze znakiem kodem spolaryzowanym

$$+13_{DEC}$$

METODA I

$$+13 + 127 = 140$$

dodajemy do liczby 13 liczbę 127 co w wyniku da nam liczbę która będzie o 13 większa od wartości środkowej w tym przypadku 140

$$140_{DEC} = 10001100_{BIN}$$

przekształcamy otrzymaną wartość na system binarny

$$10001100_{SPOL}$$

otrzymana liczba binarna jest liczbą spolaryzowaną

$$+13_{DEC} = 10001100_{SPOL}$$

METODA II

$$13_{DEC} = 1101_{BIN}$$

przekształcamy naszą liczbę dziesiętną na binarną

$$00001101_{BIN}$$

dopisujemy zera do uzyskania 8 bitów

$$\begin{array}{r} 01111111_{BIN} \\ + 00001101_{BIN} \\ \hline \end{array}$$

dodajemy do binarnej wartości środkowej naszą liczbę w postaci binarnej

$$10001100_{BIN}$$

$$10001100_{SPOL}$$

otrzymana liczba jest szukany kodem spolaryzowanym

$$+13_{DEC} = 10001100_{SPOL}$$

$$-20_{DEC}$$

METODA I – podobnie jak wyżej dodajemy 127 do naszej liczby ze znakiem i wynik przekształcamy na system binarny i mamy wartość spolaryzowaną.

$$-20_{DEC} = 01101011_{SPOL}$$

METODA II – podobnie jak wyżej tylko że od wartości środkowej odejmujemy naszą liczbę bez znaku w postaci binarnej i otrzymujemy wartość spolaryzowaną.

$$-20_{DEC} = 01101011_{SPOL}$$

Przykład zapisania kodu spolaryzowanego w postaci dziesiętnej

10101101_{SPOL}

METODA I

$$10101101_{BIN} = 173_{DEC}$$

| przekształcamy wartość binarną na dziesiętną

$$173 - 127 = 46$$

| odejmujemy od otrzymanej wartości wartość środkową 127

$$+46_{DEC}$$

| w wyniku otrzymujemy szukaną wartość dziesiętną

$$10101101_{SPOL} = +46_{DEC}$$

METODA II – działa tylko w przypadku liczb dodatnich

$$1 \quad 0101101$$

| odcinamy bit znaku

$$0101101_{BIN} = 12_{DEC}$$

| przeliczamy bity z liczbą na system dziesiętny

$$12 + 1 = 13$$

| dodajemy do otrzymanej wartości dziesiętnej wartość 1

$$+13_{DEC}$$

| w sumie otrzymujemy szukaną wartość dziesiętną

$$10101101_{SPOL} = +46_{DEC}$$

01101011_{SPOL}

METODA I – podobnie jak wyżej.

METODA II – działa tylko w przypadku liczb ujemnych

$$0 \quad 1101011$$

| odcinamy bit znaku

$$0010100_{BIN} = 20_{DEC}$$

| bity z liczbą negujemy i przeliczamy na system dziesiętny

$$-20_{DEC}$$

| otrzymana liczba jest szukaną liczbą dziesiętną

$$01101011_{SPOL} = -20_{DEC}$$

Kod U2

W kodzie **U2** podobnie jak w kodzie **ZM** bit znaku dla + przyjmuje wartość 0, a dla znaku – wartość 1. W przypadku liczby dodatniej sprawa wygląda identycznie jak w kodzie **ZM**. W przypadku liczby ujemnej przekształcenie wygląda inaczej.

Przykład zapisania liczby dziesiętnej ze znakiem w kodzie U2

-30_{DEC}

$30_{BIN} = 11110_{BIN}$

| przekształcamy liczbę bez znaku na postać binarną

00011110_{BIN}

| uzupełniamy brakujące zera do zapewnienia 8 bitów

11100001

| negujemy liczbę

11100001

| zwiększamy liczbę o 1

+1

 11100010

| wynik jest szukaną wartością w kodzie U2

11100010_{U2}

$$-30_{DEC} = 11100010_{U2}$$

Przekształcenie kodu **U2** na liczbę dziesiętną dla liczb dodatnich postępujemy podobnie jak w kodzie **ZM**. W przypadku liczby ujemnej wygląda to inaczej.

Przykład przekształcenia liczby w kodzie U2 na liczbę dziesiętną

10110101_{U2}

01001010

| negujemy liczbę

01001010

| zwiększamy liczbę o 1

+1

 01001011

$01001011_{BIN} = 75_{DEC}$

| przeliczamy otrzymaną liczbę na system dziesiętny

-75_{DEC}

| że pierwszy znak w kodzie U2 miał wartość 1 to liczba jest ujemna

$$10110101_{U2} = -75_{DEC}$$