



# Podstawowe operacje arytmetyczne na liczbach binarnych

# URZĄDZENIA TECHNIKI KOMPUTEROWEJ



## DODAWANIE

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

## ODEJMOWANIE

$$\begin{array}{r} 0 \\ -0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ -0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ -1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ -1 \\ \hline -1 \end{array}$$

## MNOŻENIE

$$\begin{array}{r} 0 \\ \times 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

## DZIELENIE

$$\begin{array}{r} 0 \\ :0 \\ \hline \text{ERROR} \end{array}$$

$$\begin{array}{r} 0 \\ :1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ :0 \\ \hline \text{ERROR} \end{array}$$

$$\begin{array}{r} 1 \\ :1 \\ \hline 1 \end{array}$$



Przykłady:

**DODAWANIE**

$$\begin{array}{r} 10001100_{(2)} \\ +00001101_{(2)} \\ \hline 10011001_{(2)} \end{array}$$

**DODAWANIE**

$$\begin{array}{r} 10011101_{(2)} \\ +00111111_{(2)} \\ \hline 11011100_{(2)} \end{array}$$



Przykład odejmowania:

$$\begin{array}{r}
 1100 \\
 - 0101 \\
 \hline
 0111
 \end{array}$$

Sprawdzenie:

$$\begin{array}{l}
 1100 = 12_{(10)} \\
 - 0101 = -5_{(10)} \\
 \hline
 0111 = 7_{(10)}
 \end{array}$$

	0	0	<del>1</del> + <del>1</del> → 1 + 1	
<b>x=</b>	<del>1</del>	<del>1</del>	0	0
<b>y=</b>	0	1	0	1
<b>x-y=</b>	0	1	1	1



Przykłady:

## ODEJMOWANIE

$$\begin{array}{r} 10001100 \\ - 00001101 \\ \hline \end{array}$$

**01111111**

## ODEJMOWANIE

$$\begin{array}{r} 10011101 \\ - 00111111 \\ \hline \end{array}$$

**01011110**



Przykład mnożenia:

$$\begin{array}{r}
 0100 \\
 \times 0101 \\
 \hline
 10100
 \end{array}$$

Sprawdzenie:

$$\begin{array}{l}
 0100 = 4_{(10)} \\
 \times 0101 = \times 5_{(10)} \\
 \hline
 10100 = 20_{(10)}
 \end{array}$$

x=			0	1	0	0
y=			0	1	0	1
			0	1	0	0
		0	0	0	0	
	0	1	0	0		
x*y=	0	1	0	1	0	0



Przykłady:

## MNOŻENIE

$$\begin{array}{r} 10001100_{(2)} \\ \times 00001101_{(2)} \\ \hline 11100011100_{(2)} \end{array}$$

## MNOŻENIE

$$\begin{array}{r} 10011101 = 157_{(10)} \\ \times 00111111 = 63_{(10)} \\ \hline 10011010100011 = 9891_{(10)} \end{array}$$



# Zamiana liczb dziesiętnych na binarne



**Założenie:** zamiana liczb będzie dotyczyła liczb całkowitych w zakresie od 0 do 255

Zwykła książkowa metoda zamiany liczby dziesiętnej na binarną polega na kolejnym dzieleniu ilorazu przez liczbę 2 aż iloraz będzie wynosił 0.

W algorytmie zapamiętywane są reszty z dzielenia i to one tworzą liczbę binarną.

W tym celu będzie nam potrzebna znajomość dwóch operatorów:

## **div i mod**

**div** – dzieli całkowicie (zwraca iloraz)

**mod** – zwraca resztę z dzielenia całkowitego

Np.

$$5 \text{ div } 6 = 0$$

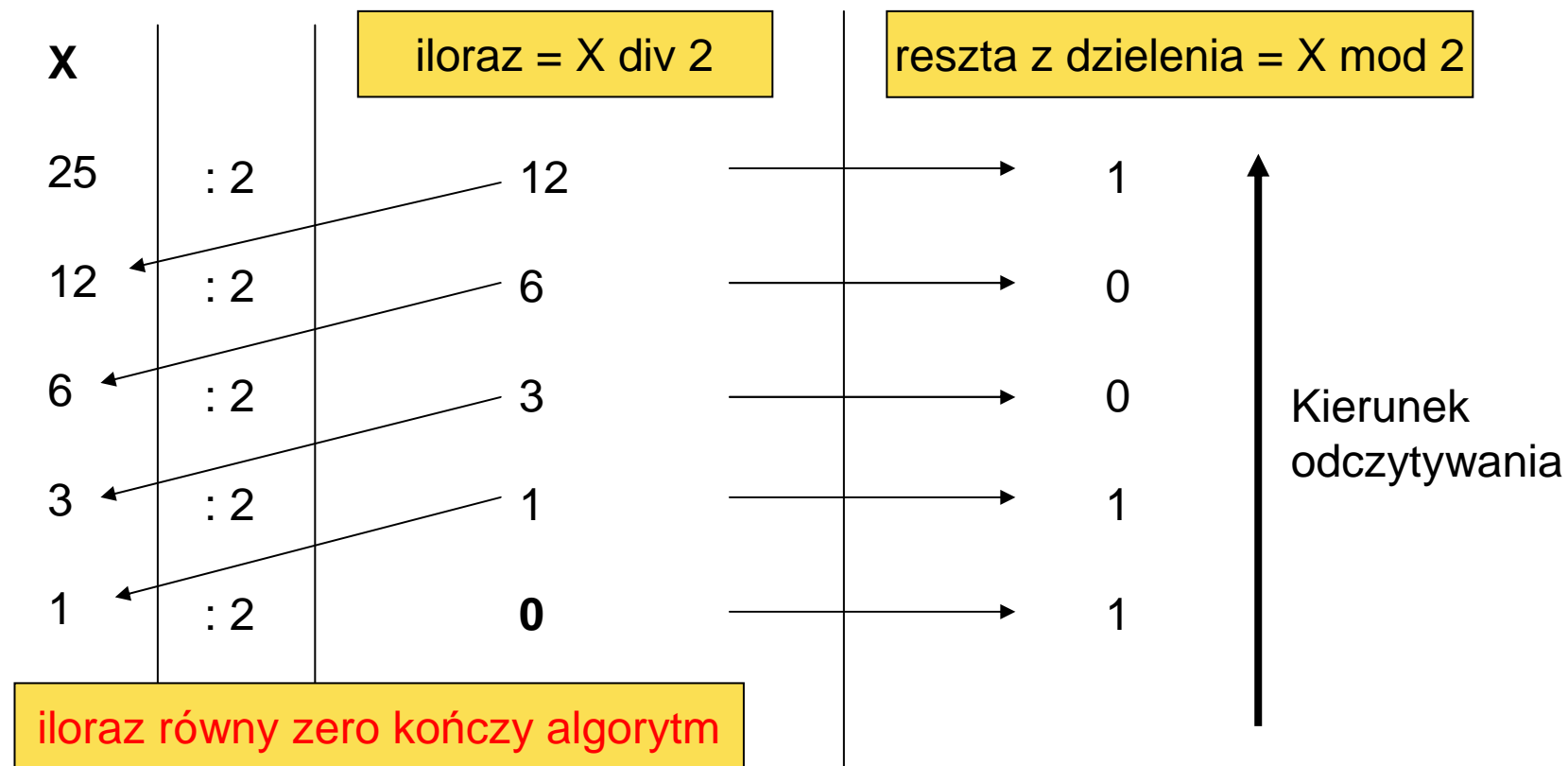
$$5 \text{ mod } 6 = 5$$

$$5 / 6 = 0.8333$$



Zamień liczbę dziesiętną  $X=25_{(10)}$  na binarną  $Y=?_{(2)}$

Schemat postępowania:



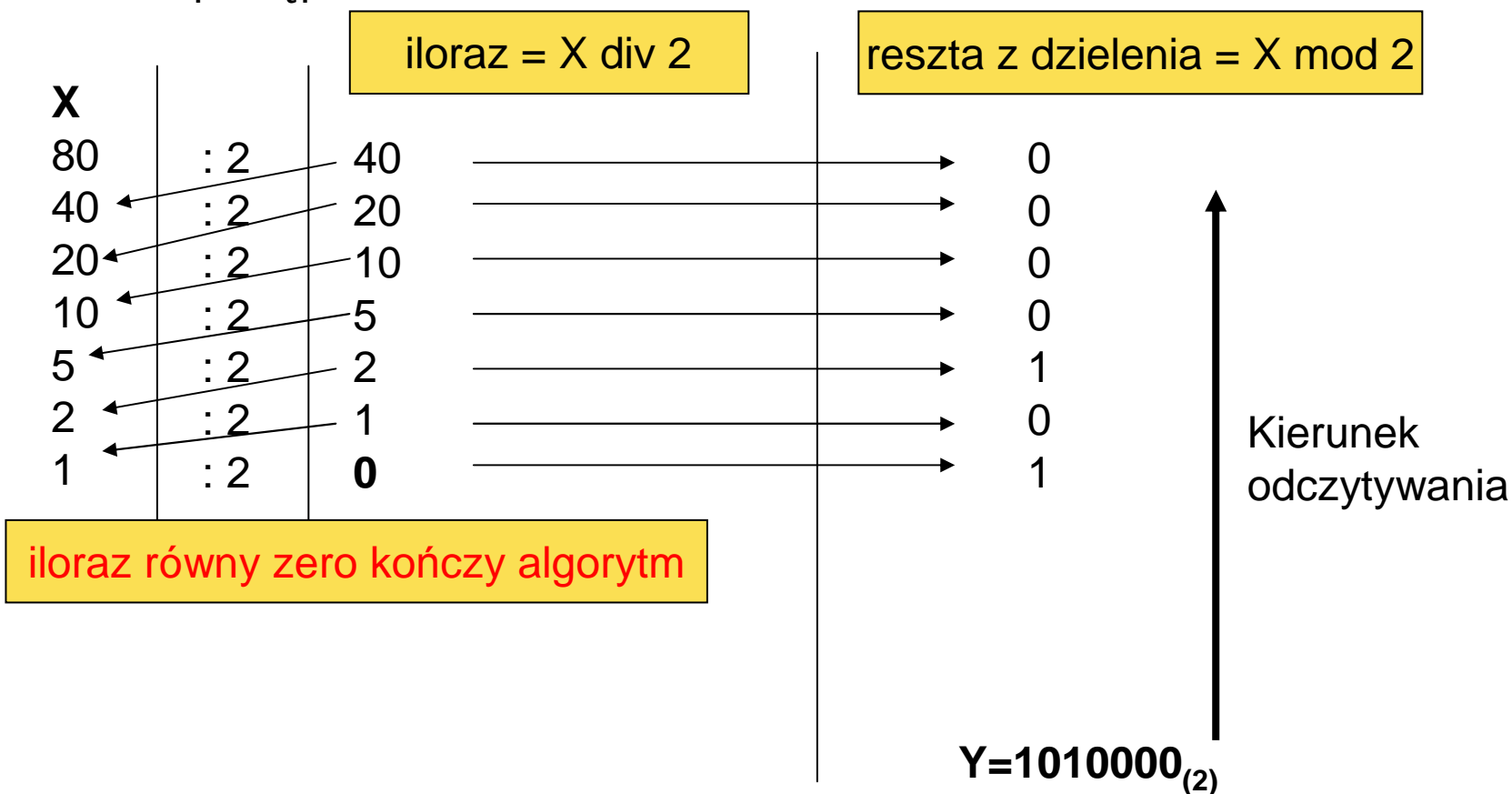
iloraz równy zero kończy algorytm

$Y=11001_{(2)}$



Zamień liczbę dziesiętną  $X=80_{(10)}$  na binarną  $Y=?_{(2)}$

Schemat postępowania:





Algorytm w Pascalu:

```
const Dwojkowa : String[8] = '00000000';  
  
var Pozycja : Byte;  
  
begin  
  Pozycja := 8;  
  while X <> 0 do  
    begin  
      if (X mod 2) = 1 then Dwojkowa[Pozycja] := '1';  
      X := X div 2;  
      Pozycja := Pozycja - 1;  
    end;  
  dec2bin := Dwojkowa;  
end;
```