

4. MATERIAŁ NAUCZANIA

4.1. Systemy i kody liczbowe

4.1.1. Materiał nauczania

Systemy pozycyjne

Powszechnie używane systemy liczenia to systemy pozycyjne. Każdy system pozycyjny posiada swoją podstawę p , (np. podstawą systemu dziesiętnego jest 10) oraz zbiór znaków służących do zapisu liczb. Wartość, jaką reprezentuje znak liczby systemu pozycyjnego, zależy od pozycji jaką zajmuje. Każda pozycja ma swoją wagę w czyli znaczenie wyrażone zależnością:

$$w = p^k$$

gdzie w - waga, p – podstawa systemu, k – pozycja.

Zapis liczby w systemie pozycyjnym jest umownym zapisem współczynników (znaków) przy odpowiednich potęgach podstawy systemu.

Dowolna liczba zapisana w dowolnym systemie pozycyjnym ma postać:

$$L_p = a_n a_{n-1} \dots a_0 a_{-1} a_{-2} \dots a_{-k} = \sum_{i=-k}^{i=n} a_i p^i$$

W systemie pozycyjnym o podstawie p :

- liczba znaków graficznych potrzebnych do zapisu dowolnej liczby równa jest podstawie systemu p ,
- wagi poszczególnych pozycji mają wartość p^k gdzie k jest pozycją cyfry.

System dwójkowy (binarny)

Podstawą systemu dwójkowego jest $p = 2$, liczby zapisywane są za pomocą cyfr 0, 1. Bit to jednostka logiczna, najmniejsza wartość informacji, która może przyjąć wartość 0 lub 1. Skrajna pozycja z lewej strony ma najwyższą wagę (największe znaczenie) i nazywa się najstarszym bitem - MSB, natomiast skrajna pozycja z prawej strony ma najniższą wagę (najmniejsze znaczenie) i nazywa się najmłodszym bitem - LSB.

Zamiana liczby całkowitej dziesiętnej na liczbę w zapisie dwójkowym polega na dzieleniu jej przez podstawę systemu dwójkowego, czyli 2. Wynik dzielenia zapisywany jest jako część całkowita i reszta. W kolejnym kroku z częścią całkowitą postępujemy jak poprzednio. Reszta dzielenia jest zawsze równa 1 lub 0. Ciąg tych reszt stanowi liczbę binarną, przy czym reszta pierwszego dzielenia jest najmłodszym bitem.

Zamiana liczby ułamkowej dziesiętnej na liczbę w zapisie dwójkowym polega na mnożeniu jej przez podstawę systemu dwójkowego, czyli 2. Wynik mnożenia zapisywany jest jako część całkowita i ułamkowa. W kolejnym kroku z częścią ułamkową postępujemy jak poprzednio. Część całkowita jest zawsze równa 1 lub 0. Ciąg części całkowitych stanowi liczbę binarną, przy czym część całkowita pierwszego mnożenia jest najstarszym bitem.

Przykład 1

Wykonać konwersję dziesiętno-dwójkową liczby 215.

$$215 : 2 = 107 + r.1 \text{ LSB}$$

$$107 : 2 = 53 + r.1$$

$$53 : 2 = 26 + r. 1$$

$$26 : 2 = 13 + r. 0$$

$$13 : 2 = 6 + r. 1$$

$$6 : 2 = 3 + r. 0$$

$$3 : 2 = 1 + r. 1$$

$$1 : 2 = 0 + r. 1 \text{ MSB}$$

Liczba 215 w zapisie dwójkowym to 11010111

Przykład 2

Wykonać konwersję dziesiętno-dwójkową liczby 0,125

$$\begin{array}{c} \text{MSB} \\ \downarrow \\ 0,125 \cdot 2 = 0,25 \\ 0,25 \cdot 2 = 0,5 \\ 0,5 \cdot 2 = 1,0 \\ \uparrow \\ \text{LSB} \end{array}$$

Liczba 0,125 w zapisie dwójkowym to 0,001.

W przypadku konwersji dziesiętno-dwójkowej liczb ułamkowych zazwyczaj należy przyjąć pewne przybliżenie.

Konwersji dwójkowo-dziesiętnej dokonuje się mnożąc każdy bit liczby binarnej przez wagę jej pozycji, a następnie sumując uzyskane wartości.

Przykład 3

Wykonać konwersję dwójkowo-dziesiętną liczby 11100011

$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 2^3 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 64 + 32 + 16 + 2 + 1 = 115$$

System binarny wykorzystywany jest między innymi w technice cyfrowej i informatyce.

Działania na liczbach binarnych

Działania arytmetyczne takie jak: dodawanie odejmowanie, dzielenie i mnożenie wykonywane są na liczbach binarnych tak jak w systemie dziesiętnym. Należy jedynie pamiętać, że podstawą systemu jest 2 i wynik każdego działania musi równać się 0 lub 1.

Przykład 4

Należy dodawać liczby binarne 1000111 i 10101.

$$\begin{array}{r} 1000111 \\ + 10101 \\ \hline 1011100 \end{array}$$

Przykład 5

Należy odjąć od liczby binarnej 1000111 liczbę binarną 10101.

$$\begin{array}{r} 1000111 \\ - 10101 \\ \hline 110010 \end{array}$$

System ósemkowy

Podstawą systemu ósemkowego jest $p = 8$, liczby zapisywane za pomocą cyfr ze zbioru $\{0, 1, 2, 3, 4, 5, 6, 7\}$.

W celu dokonania konwersji liczby dziesiętnej na ósemkową, należy najpierw dokonać konwersji dziesiętno-dwójkowej, następnie podzielić liczbę binarną na grupy po trzy bity (poczynając od najmłodszego bitu, jeśli grupa z najstarszych bitów jest niepełna, należy ją uzupełnić o nieznaczące zera), następnie każdą trójkę bitów zapisać jako liczbę dziesiętną z zakresu od 0 do 7.

Przykład

Wykonać konwersję dziesiętno-ósemkową liczby 215.

$$215_{10} = 11010111_2 = (11 \ 010 \ 111)_2 = \begin{matrix} 011 & 010 & 111 \\ \Downarrow & \Downarrow & \Downarrow \\ 3 & 2 & 7 \end{matrix} = 327_8$$

Konwersję ósemkowo-dziesiętną przeprowadza się w sposób odwrotny: każdy znak liczby ósemkowej konwertuje się na wartość dwójkową, następnie dokonuje się konwersji dwójkowo-dziesiętnej.

Przykład

Wykonać konwersję ósemkowo-dziesiętną liczby 435_8

$$\begin{matrix} (4 & 3 & 5)_8 \\ \Downarrow & \Downarrow & \Downarrow \\ 100 & 011 & 101 \end{matrix} = 269_{10}$$

System ósemkowy wykorzystywany jest między innymi w technice cyfrowej i informatyce.

System szesnastkowy (heksadecymalny)

Podstawą systemu szesnastkowego jest $p = 16$, liczby zapisywane za pomocą znaków ze zbioru $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

W celu dokonania konwersji liczby dziesiętnej na szesnastkową, należy najpierw dokonać konwersji dziesiętno-dwójkowej, następnie podzielić liczbę binarną na grupy po cztery bity (poczynając od najmłodszego bit, jeśli grupa z najstarszych bitów jest niepełna, należy ją uzupełnić o nieznaczące zera), następnie każdej czwórce bitów należy przyporządkować znak liczby szesnastkowej.

Przykład

Wykonać konwersję dziesiętno-szesnastkową liczby 215.

$$215_{10} = 11010111_2 = (1101 \ 0111)_2 = \begin{matrix} 1101 & 0111 \\ \Downarrow & \Downarrow \\ D & 7 \end{matrix} = D7_{16}$$

Konwersję szesnastkowo-dziesiętną przeprowadza się w sposób odwrotny: każdy znak liczby szesnastkowej konwertuje się na wartość dwójkową, następnie dokonuje się konwersji dwójkowo-dziesiętnej.

Przykład

Wykonać konwersję ósemkowo-dziesiętną liczby $1A9_{16}$

$$\begin{matrix} (1 & A & 9)_{16} \\ \Downarrow & \Downarrow & \Downarrow \\ 0001 & 1010 & 1001 \end{matrix} = 425_{10}$$

System szesnastkowy wykorzystywany jest między innymi w technice cyfrowej oraz informatyce.

4.1.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń

1. Czym charakteryzuje się pozycyjny system liczenia?
2. Jakie znasz systemy pozycyjne?
3. Jaka liczba jest podstawą systemu binarnego?
4. Za pomocą jakich znaków zapisuje się liczby binarne?
5. Jaka liczba jest podstawą systemu ósemkowego?
6. Za pomocą jakich znaków zapisuje się liczby w systemie ósemkowym?
7. Jaka liczba jest podstawą systemu heksadecymalnego?
8. Za pomocą jakich znaków zapisuje się liczby w systemie heksadecymalnym?

4.1.3. Ćwiczenia

Ćwiczenie 1

Zapisz w systemie dwójkowym liczby 597 oraz 354.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać konwersji dziesiętno-dwójkowej,
- 3) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- kalkulator,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Zapisz w systemie dziesiętnym liczby 110111001_2 oraz 10110011101_2

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać konwersji dwójkowo-dziesiętnej,
- 3) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- kalkulator,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zapisz w systemie szesnastkowym liczby 439_{10} oraz 248_{10} .

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać konwersji dziesiętno-szesnastkowej,
- 3) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- kalkulator,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 4

Zapisz w systemie dziesiętnym liczby 37_8 oraz 75_8

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać konwersji ósemkowo-dziesiętnej,
- 3) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- kalkulator,
- literatura uzupełniająca zgodna z punktem 6.

4.1.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) zapisać liczbę dziesiętną w systemie binarnym?	<input type="checkbox"/>	<input type="checkbox"/>
2) dokonać konwersji dwójkowo-dziesiętnej?	<input type="checkbox"/>	<input type="checkbox"/>
3) zapisać liczbę dziesiętną w systemie ósemkowym?	<input type="checkbox"/>	<input type="checkbox"/>
4) dokonać konwersji ósemkowo-dziesiętnej?	<input type="checkbox"/>	<input type="checkbox"/>
5) zapisać liczbę dziesiętną w systemie heksadecymalnym?	<input type="checkbox"/>	<input type="checkbox"/>
6) dokonać konwersji szesnastkowo-dziesiętnej?	<input type="checkbox"/>	<input type="checkbox"/>
7) wykonać działania arytmetyczne na liczbach binarnych?	<input type="checkbox"/>	<input type="checkbox"/>

4.2. Kody liczbowe

4.2.1. Materiał nauczania

Kodowanie to przypisywanie różnym informacjom pewnych symboli. Kodem nazywamy zestaw symboli przypisanej danej informacji.

Podstawowe kody używane w technice cyfrowej i informatyce to kod naturalny BCD, pierwotny kod 1 z n, kod Graya, stosuje się również kod Aikena, Johnsona, Hamminga.

Kod Aikena jest kodem wagowym, nie jest jednak kodem pozycyjnym. Charakteryzuje się osią asymetrii przebiegająca między kodem cyfry 4 i cyfry 5. Wyrazy kodu Aikena w jednakowej odległości od tej osi różnią się negacją wszystkich bitów, np. kod Aikena cyfry 2 to 0010, natomiast kod Aikena cyfry 7 to 1101.

Kod Johnsona zwany też pseudopierścieniowym wymaga pięciu bitów do zakodowania każdej cyfry dziesiętnej. Cechuje go specyficzny rozkład zer i jedynek np. kod Johnsona cyfry 3 to 00111, natomiast kod Johnsona cyfry 6 to 11110. Kod Johnsona nie jest kodem wagowym.

Kod Hamminga jest kodem z zabezpieczeniem, odpornym na zakłócenia. Jego konstrukcja umożliwia odtworzenia pierwotnej informacji w urządzeniu odbierającym dopiero gdy wystąpi błąd.

Kod naturalny BCD

Kod naturalny BCD utworzony jest przez zapisanie kolejnych cyfr dziesiętnych w systemie dwójkowym. Kod ten nazywany jest również kodem BCD 8421.

Przykład

Zapisać liczbę 967_{10} w kodzie BCD.

Każdej cyfrze należy przyporządkować odpowiednik binarny zatem

9	6	7
↓	↓	↓
1001	0110	0111

Kod pierwotny 1 z n

Kod pierwotny 1 z n jest często kodem wejściowym urządzenia, n oznacza liczbę bitów kodu. Liczba zapisana w tym kodzie ma tylko na jednym bicie 1 natomiast pozostałe bity są równe 0.

Przykład

Zapisać liczby od 0_{10} do 4_{10} w kodzie 1 z 10

0_{10}	w kodzie 1 z 10 odpowiada	0000000001
1_{10}	w kodzie 1 z 10 odpowiada	0000000010
3_{10}	w kodzie 1 z 10 odpowiada	0000000100
4_{10}	w kodzie 1 z 10 odpowiada	0000001000

Kod Graya

Kod Graya zwany też refleksyjnym charakteryzuje się tym, że dwa sąsiadujące wyrazy kodowe różnią się tylko jednym bitem. Jednobitowy kod Graya posiada jedynie dwa wyrazy 0 oraz 1.

A
0
1

W celu rozszerzenia go do dwóch bitów należy ponownie lecz w odwrotnej kolejności przepisać ciąg wyrazów kodu jednobitowego.

0
1
-
1
0

Następnie do początkowych wyrazów dopisać na początku dodatkowy bit 0, a do dopisanych dopisać na początku dodatkowy bit 1.

A B
0 0
0 1
1 1
1 0

Kod Graya n-bitowy tworzy się analogicznie, powtarzając n-1 bitowy kod Graya następnie dopisując dodatkowy bit 0 do pierwotnej części, 1 do części dopisanej.

4.2.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Na czym polega kodowanie?
2. Jakie znasz kody stosowane w technice cyfrowej?
3. Czym charakteryzuje się kod wagowy Aikena?
4. W jaki sposób zapisuje się liczbę w kodzie BCD?
5. W jaki sposób zapisuje się informacje w kodzie 1 z n?
6. Czym charakteryzuje się kod Graya?

4.2.3. Ćwiczenia

Ćwiczenie 1

Zapisz w kodzie BCD liczbę 536.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać konwersji dziesiętnej-dwójkowej każdej cyfry kodowanej liczby dziesiętnej,
- 3) zapisać liczbę w kodzie BCD,
- 4) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Dokonaj konwersji liczby 1001 0011 0111 zapisanej w kodzie BCD na postać dziesiętną.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać konwersji dwójkowo-dziesiętnej każdej czwórki bitów na cyfrę,
- 3) zapisać liczbę w systemie dziesiętnym,
- 4) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zapisz trzybitowy kod Graya.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) zapisać jednobitowy kod Graya,
- 3) na jego podstawie stworzyć dwubitowy kod Graya,
- 4) na podstawie dwubitowego kodu Graya zapisać wyrazy trzybitowego kodu Graya,
- 5) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

4.2.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) zapisać dowolną liczbę dziesiętną w kodzie BCD?	<input type="checkbox"/>	<input type="checkbox"/>
2) zapisać liczbę kodowaną w kodzie BCD w postaci dziesiętnej?	<input type="checkbox"/>	<input type="checkbox"/>
3) zapisać kolejne wyrazy w kodzie 1 z n?	<input type="checkbox"/>	<input type="checkbox"/>
4) zapisać wyrazy n-bitowego kodu Graya?	<input type="checkbox"/>	<input type="checkbox"/>

4.3. Algebra Boole'a

4.3.1. Materiał nauczania

Algebra Boole'a operuje zmiennymi dwuwartościowymi o wartościach 1 i 0. Podstawowe operacje algebry Boole'a to:

- suma logiczna (alternatywa, dysjunkcja),
- iloczyn logiczny (koniunkcja),
- negacja (dopełnienie).

Suma logiczna jest równa 1, gdy którykolwiek z e składników jest równy 1. Sumę argumentów a i b oznacza się jako $a + b$.

Iloczyn logiczny jest równy 1, gdy wszystkie czynniki są równe 1. Iloczyn logiczny argumentów a i b oznacza się jako $a \cdot b$, lub ab .

Suma logiczna i iloczyn logiczny mogą mieć dowolną ilość argumentów.

Negacja jest działaniem jednoargumentowym i równa się 1 gdy argument ma wartość 0. Negację oznacza się \bar{a} i czyta „nie a”

Tabela 1. Definicje operacji bulowskich

a	b	$a + b$	ab	\bar{a}
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Tabela 2. Własności funkcji logicznych

Suma logiczna	Iloczyn logiczny	Negacja
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\bar{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\bar{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

Tabela 3. Prawa i twierdzenia algebry Boole'a

Prawa algebry Boole'a	Podstawowe twierdzenia algebry Boole'a
$a + b = b + a$	Tw.1. $a + ac = a$ $a(a + b) = a$
$a(b + c) = ab + ac$	Tw.2. $a + \bar{a}b = a + b$ $a(\bar{a} + b) = ab$
$(a + b) + c = a + (b + c)$	Tw.3. $a + a = a$ $aa = a$
$a + 0 = a$	
$a + 1 = 1$	
$a + \bar{a} = 1$	
$ab = ba$	
$a + bc = (a + b)(a + c)$	
$(ab)c = a(cb)$	
$a1 = a$	
$a0 = 0$	
$a\bar{a} = 0$	
$\bar{\bar{a}} = a$	
Prawa de Morgana	
$\overline{a + b} = \bar{a}\bar{b}$ $\overline{ab} = \bar{a} + \bar{b}$	

Funkcje specjalne algebry Boole'a:

Funkcja Pirce'a $a \downarrow b$ przyjmuje wartość 1, jeśli $a = 0$ i $b = 0$, można ją wyrazić za pomocą podstawowych funkcji algebry Boole'a:

$$a \downarrow b = \overline{a \cdot b} = \overline{a + b}$$

Funkcja ta jest negacją sumy logicznej

Funkcja Sheffera $a | b$ przyjmuje wartość 1, jeśli $a = 0$ lub $b = 0$ można ją wyrazić za pomocą podstawowych funkcji algebry Boole'a:

$$a | b = \overline{a} + \overline{b} = \overline{a \cdot b}$$

Funkcja ta jest negacją iloczynu logicznego.

Suma modulo 2 (różnica symetryczna, nierównoważność) $a \oplus b$ przyjmuje wartość 1, tylko jeden argument ma wartość 1. Można ją wyrazić za pomocą podstawowych funkcji algebry Boole'a:

$$a \oplus b = a \overline{b} + \overline{a} b$$

Równoważność $a \otimes b$ ($a \equiv b$) przyjmuje wartość 1, jeśli argumenty mają tę samą wartość, można ją wyrazić za pomocą podstawowych funkcji algebry Boole'a:

$$a \otimes b = a \equiv b = a \cdot b + \overline{a} \cdot \overline{b}$$

4.3.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Na jakich wartościach operuje algebra Boole'a?
2. Jakie są podstawowe funkcje algebry Boole'a?
3. Kiedy suma logiczna przyjmuje wartość 1?
4. Kiedy iloczyn logiczny przyjmuje wartość 1?
5. Ilu argumentową jest funkcja negacji?
6. Kiedy suma modulo dwa przyjmuje wartość 1?
7. Dla jakich argumentów równoważność przyjmuje wartość 1?
8. Jak brzmią prawa de Morgana?

4.3.3. Ćwiczenia

Ćwiczenie 1

Sprawdź prawdziwość wyrażenia $(a + \overline{b})(ac + \overline{a}\overline{b} + bc)\overline{a}b = 0$, wykorzystując metodę przekształceń algebraicznych oraz prawa algebry Boole'a.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać przekształceń algebraicznych,
- 3) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Doprowadź do prostszej postaci wyrażenie

$$[a\bar{b}(c + \bar{c}) + ab](\bar{c}\bar{a} + a)$$

wykorzystując metodę przekształceń algebraicznych oraz prawa algebry Boole'a.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) dokonać przekształceń algebraicznych,
- 3) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

4.3.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) określić wartość sumy logicznej?	<input type="checkbox"/>	<input type="checkbox"/>
2) określić wartość iloczynu logicznego?	<input type="checkbox"/>	<input type="checkbox"/>
3) zastosować prawa de Morgana?	<input type="checkbox"/>	<input type="checkbox"/>
4) określić właściwości sumy logicznej?	<input type="checkbox"/>	<input type="checkbox"/>
5) określić właściwości iloczynu logicznego?	<input type="checkbox"/>	<input type="checkbox"/>
6) zastosować prawa algebry Boole'a do minimalizacji wyrażeń?	<input type="checkbox"/>	<input type="checkbox"/>
7) zastosować prawa algebry Boole'a do sprawdzania prawdziwości wyrażeń?	<input type="checkbox"/>	<input type="checkbox"/>

4.4. Bramki logiczne

4.4.1. Materiał nauczania

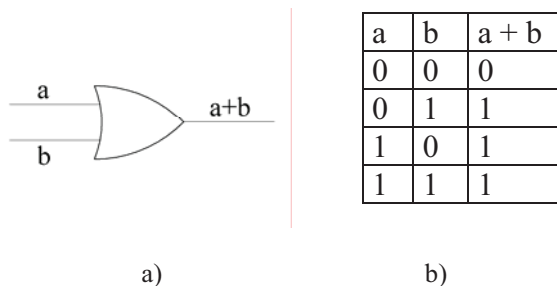
Bramki logiczne nazywane też funktorami realizują funkcję logiczną jednej lub wielu zmiennych. Zmienna logiczna to sygnał elektryczny (napięcie) dwustanowy. W konwencji dodatniej niski poziom napięcia oznaczany „L” odpowiada stanowi logicznemu 0, natomiast wysoki poziom napięcia oznaczany „H” odpowiada stanowi logicznemu 1. W konwencji ujemnej niski poziom napięcia oznaczany „L” odpowiada stanowi logicznemu 1, natomiast wysoki poziom napięcia oznaczany „H” odpowiada stanowi logicznemu 0. Do analizy układów cyfrowych zazwyczaj stosuje się konwencję dodatnią.

Działanie bramek logicznych opisuje się za pomocą:

- tablic prawdy, które ujmują wszystkie kombinacje sygnałów wejściowych i odpowiadające im kombinacje sygnałów wyjściowych,
- funkcji logicznych wyrażonych w języku algebry Boole’a.

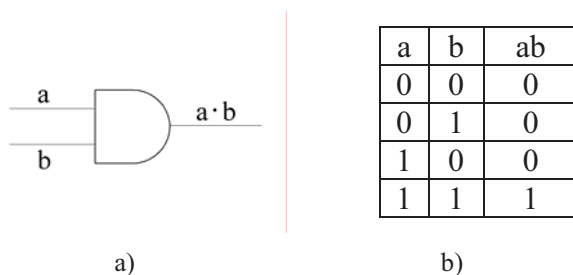
Zbiór funktorów, który pozwala zrealizować dowolną funkcję logiczną nazywa się systemem funkcjonalnie pełnym.

Bramka OR (LUB) może mieć dwa lub więcej wejść, realizuje funkcję sumy logicznej $a + b$



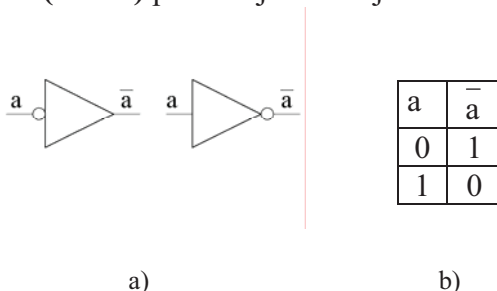
Rys. 1. Dwuwejściowa bramka OR a) symbol graficzny, b) tablica prawdy. [2, s.34]

Bramka AND (I) może mieć dwa lub więcej wejść, realizuje funkcję iloczynu logicznego ab



Rys. 2. Dwuwejściowa bramka AND a) symbol graficzny; b) tablica prawdy. [2, s.34]

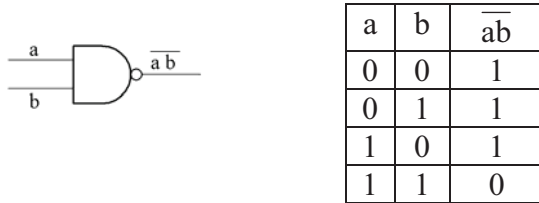
Bramka NOT (NIE) posiada jedno wejście i realizuje funkcję negacji \bar{a}



Rys. 3. Bramka NOT a) symbole graficzne; b) tablica prawdy. [2, s.34]

W symbolu graficznym bramki NOT można wyróżnić symbol wzmacniacza i kółko (może być umieszczone na wejściu i na wyjściu) reprezentujące inwersję.

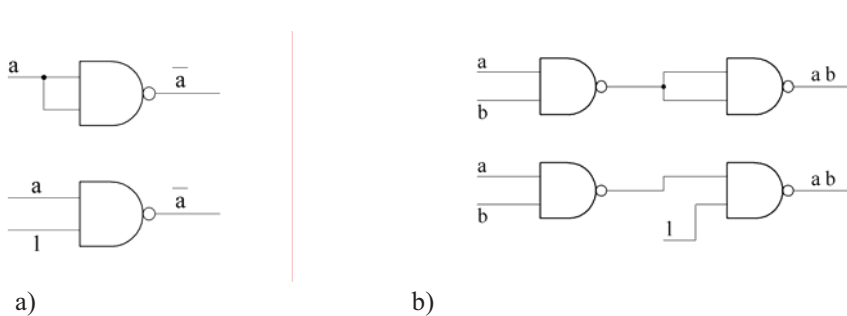
Bramka NAND (NIE I) może mieć dwa lub więcej wejść, realizuje funkcję zanegowanego iloczynu logicznego \overline{ab}



a) b)

Rys. 4. Dwuwejściowa bramka NAND a) symbole graficzne; b) tablica prawdy. [2, s.35]

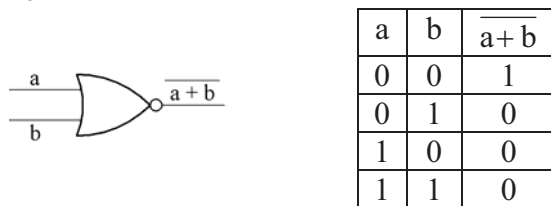
Bramka NAND jest funkcjonalnie pełna, ponieważ realizuje operację iloczynu AND i operację negacji NOT.



a) b)

Rys. 5. a) Realizacja negacji za pomocą bramki NAND. [2, s.35] b) Realizacja iloczynu logicznego za pomocą bramki NAND. [2, s.36]

Bramka NOR (NIE LUB) może mieć dwa lub więcej wejść, realizuje funkcję sumy logicznej $\overline{a+b}$

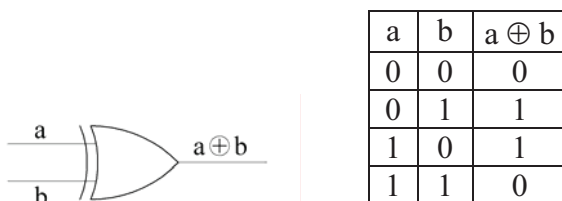


a) b)

Rys. 6. Dwuwejściowa bramka NOR a) symbole graficzne; b) tablica prawdy. [2, s.36]

Bramka NOR jest funkcjonalnie pełna.

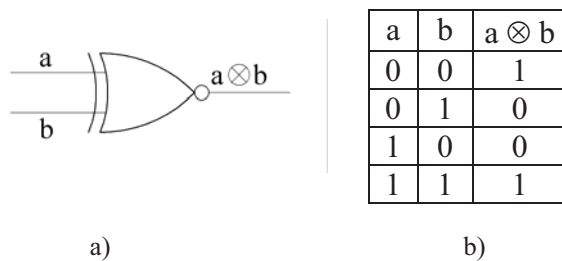
Bramka Ex-OR (XOR, ALBO) realizuje funkcję sumy modulo dwa $a \oplus b = a\bar{b} + \bar{a}b$



a) b)

Rys. 7. Dwuwejściowa bramka Ex-OR a) symbol graficzny; b) tablica prawdy [2, s.36]

Bramka Ex-NOR (NIE ALBO) realizuje funkcję równoważności $a \otimes b = a \equiv b = ab + \bar{a} \bar{b}$



Rys. 8. Dwuwejściowa bramka Ex-NOR a) symbol graficzny; b) tablica prawdy. [2, s.36]

4.4.2. Pytania sprawdzające

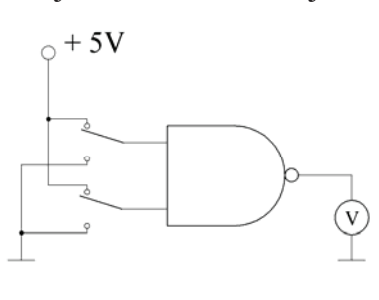
Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Czym jest zmienna logiczna w układach cyfrowych?
2. Jakie stany przyjmują zmienne logiczne?
3. Jaką funkcję realizuje bramka OR?
4. Jaka bramka realizuje funkcję iloczynu logicznego?
5. Która z bramek posiada tylko jedno wejście?
6. Jakie bramki są funkcjonalnie pełne?
7. Jaką funkcję realizuje bramka Ex-NOR?
8. Jaka bramka realizuje funkcję sumy modulo dwa?

4.4.3. Ćwiczenia

Ćwiczenie 1

Zbadaj właściwości funkcjonalne bramki NAND.



Układ do badania właściwości funkcjonalnych bramki NAND.

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) połączyć układ według schematu,
- 3) zbadać działanie bramki (mierząc napięcie na jej wyjściu) podając na jej wejścia sygnał zgodnie z tablicą prawdy dla bramki NAND, przyjmując +5V jako stan 1 logicznej, 0V jako stan 0 logicznego,
- 4) sformułować wnioski,
- 5) sporządzić sprawozdanie z wykonanego ćwiczenia.

- Wyposażenie stanowiska pracy:
- bramka NAND (z układu UCY 7400),
 - zasilacz +5V,
 - dwa przełączniki dwupozycyjne,
 - multimetr uniwersalny,
 - stanowisko do łączenia układów,
 - literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Wykaż, że bramka NOR jest funkcjonalnie pełna.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

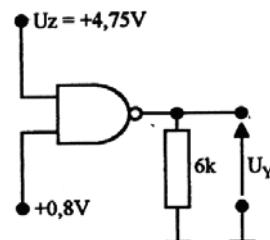
- 1) przeanalizować treść zadania,
- 2) narysować układ zbudowany z bramek NOR realizujący funkcję negacji,
- 3) narysować układ zbudowany z bramek NOR realizujący funkcję sumy logicznej,
- 4) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zbadaj działanie bramki NAND jako negatora.



Schemat układu realizującego negację na bramce NAND [5, s. 31].

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 1) narysować schemat układu pomiarowego na podstawie rysunku ,
- 2) skompletować potrzebne elementy i aparaturę pomiarową,
- 2) zbudować układ pomiarowy według zaproponowanego schematu,
- 3) zmierzyć napięcie U_Y na wyjściu układu,
- 4) oszacować dokładność pomiaru,
- 5) porównać otrzymaną wartość z danymi katalogowymi,
- 6) sformułować wnioski i sporządzić sprawozdanie z wykonanego ćwiczenia.

Wyposażenie stanowiska pracy:

- bramka NAND (z układu UCY 7400N),

- dwa zasilacze regulowane napięcia stałego $0V \div 10V$,
- multimetr cyfrowy,
- 4 rezystory $R= 100\Omega$, rezystor $R=5,6k\Omega$,
- stanowisko do łączenia układów,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

4.4.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) opisać działanie podstawowych bramek logicznych?	<input type="checkbox"/>	<input type="checkbox"/>
2) narysować symbole bramek logicznych?	<input type="checkbox"/>	<input type="checkbox"/>
3) wykazać, że bramka NOR jest funkcjonalnie pełna?	<input type="checkbox"/>	<input type="checkbox"/>
4) rozpoznać typ bramki logicznej na podstawie tablicy prawdy?	<input type="checkbox"/>	<input type="checkbox"/>
5) zrealizować negację za pomocą bramek NAND?	<input type="checkbox"/>	<input type="checkbox"/>
6) zrealizować iloczyn logiczny za pomocą bramek NAND?	<input type="checkbox"/>	<input type="checkbox"/>

4.5. Cyfrowe układy scalone

4.5.1. Materiał nauczania

Techniki wykonania cyfrowych układów scalonych

Cyfrowe układy scalone dzielimy ze względu na liczbę bramek elementarnych tworzących dany układ na:

- małej skali integracji SSI do 10 bramek,
- średniej skali integracji MSI od 10 do 100 bramek,
- dużej skali integracji LSI od 100 do 10000 bramek.

Ze względu na technologię wytwarzania rozróżniamy układy:

- DTL – obecnie niestosowane z powodu małej obciążalności i odporności na zakłócenia,
- TTL – najpopularniejsze obecnie, jest to zmodyfikowana technika DTL, gdzie elementy diodowe zastąpiono bipolarnymi tranzystorami wieloemiterowymi,
- MOS – wykorzystujące tranzystory unipolarne z kanałem typu P (PMOS) i z kanałem typu N (NMOS). W technice PMOS i NMOS wykonuje się układy MSI i LSI zawierające całe bloki funkcyjne,
- CMOS – wykorzystujące unipolarne tranzystory komplementarne, są obecnie równie popularne jak TTL,
- ECL – najszybsze układy wykorzystujące tranzystory bipolarne,
- I²L – wykorzystujące tranzystory bipolarne, charakteryzują się dużą gęstością upakowania w strukturze scalonej,
- CTD – oparte na technologii MOS, wykorzystują zjawisko magazynowania i transportu ładunku, reprezentującego informację, stosowane do budowy pamięci półprzewodnikowych.

Najczęściej w systemach cyfrowych stosuje się układy wykonane w technologiach TTL oraz CMOS.

Oznaczenia cyfrowych układów scalonych

Producenci cyfrowych układów scalonych stosują zazwyczaj własne oznaczenia. Polskie firmy stosują symbolikę zgodną z poniższymi zasadami:

- pierwszy znak – litera - określa wykonanie: U - układ scalony półprzewodnikowy monolityczny wykonany w technologii bipolarnej, M - układ scalony półprzewodnikowy monolityczny wykonany w technologii unipolarnej;
- drugi znak – litera – określa spełnianą funkcję: C – układy cyfrowe, L – układy analogowe;
- trzeci znak – litera – określa zastosowanie: X – prototyp, Y – do sprzętu profesjonalnego, A – do zastosowań specjalnych,
- czwarty znak - cyfra – określa numer serii (różny dla TTL i CMOS), dodatkowo mogą wystąpić jedna lub dwie litery określające rodzaj serii,
- kolejne znaki – dwie lub trzy cyfry określają rodzaj elementu,
- ostatnia może wystąpić litera określająca rodzaj obudowy.

Podstawowe parametry układów scalonych

Parametry układów scalonych dzielimy na statyczne i dynamiczne.

Podstawowe parametry statyczne to :

- straty mocy P_s (moc pobierana przez układ, moc rozpraszana) jest to moc tracona w układzie przy przełączaniu go przebiegiem prostokątnym o wypełnieniu $\frac{1}{2}$ i częstotliwości 100kHz, określa zapotrzebowanie układu na prąd,

- margines zakłóceń ΔU – jest to maksymalna wartość amplitudy impulsu zakłócającego, która dodana do sygnału wejściowego elementu nie powoduje przekroczenia przez sygnał wyjściowy dopuszczalnych granic,
- obciążalność N wyjścia układu określa dopuszczalną liczbę wejść innych elementów, które mogą być z tego wyjścia prawidłowo sterowane,

W katalogach podanych jest jeszcze szereg innych parametrów statycznych takich jak napięcie zasilania, prąd zasilania w stanie wysokim i niskim, napięcia i prądy wejściowe w stanie wysokim i niskim, napięcia i prądy wyjściowe w stanie wysokim i niskim.

Podstawowym parametrem dynamicznym jest czas propagacji t_p , czyli czas upływający między wystąpieniem sygnału na wejściu i na wyjściu układu. Zazwyczaj określa czasy propagacji przy zmianie sygnału logicznego na wyjściu z wysokiego na niski i z niskiego na wysoki. Czas propagacji t_p jest najczęściej średnią arytmetyczną tych czasów i określa szybkość działania układu.

Dla porównania różnych serii układów cyfrowych stosuje się współczynnik dobroci D będący iloczynem strat mocy i czasu propagacji.

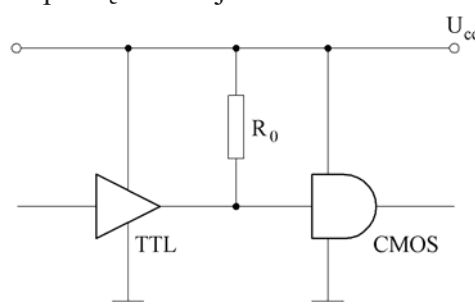
Istotnym parametrem bramek logicznych są charakterystyki przejściowe określające między innymi zależność napięcia wyjściowego od napięcia wejściowego oraz zależność prądu obciążenia źródła zasilania od napięcia wejściowego.

Najczęściej w systemach cyfrowych stosuje się układy wykonane w technologiach TTL oraz CMOS. Układy TTL są szybsze niż CMOS mają też większy margines zakłóceń. Natomiast układy CMOS charakteryzują mniejsze straty mocy.

Współpraca układów TTL i CMOS

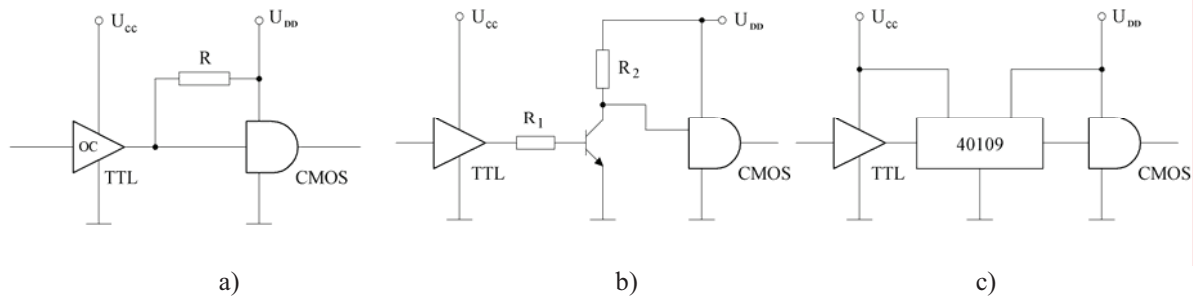
W systemach cyfrowych z elementami wykonanymi w różnych technologiach zachodzi konieczność osiągnięcia zgodności takich parametrów jak napięcia zasilania, poziomy sygnałów logicznych, prądy wejściowe i wyjściowe, a co za tym idzie również obciążalność. Poprawność współpracy zapewniają wówczas specjalne układy zwane translatorami lub konwerterami.

W przypadku sprzężenia układów TTL z układami CMOS problemem jest niedopasowanie poziomów napięciowych. Stosuje się wówczas rezystor podciągający R_0 (o wartości z zakresu $1,5 \div 4,7 \text{ k}\Omega$), którego zadaniem jest zwiększenia napięcia wyjściowego z układów TTL w stanie wysokim. Zaleca się też do wyjścia bramki TTL sterującej układami CMOS nie podłączać wejść TTL.



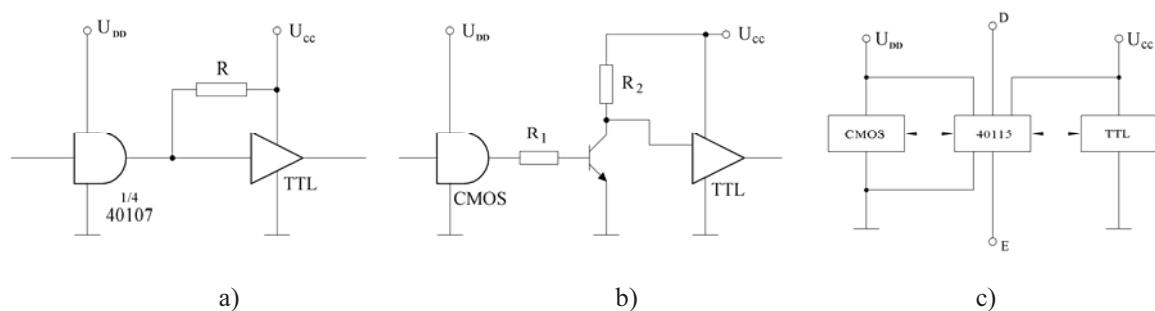
Rys. 9. Sprzężenie TTL z CMOS z użyciem rezystora podciągającego. [2,s.174]

Aby wykorzystać krótszy czas propagacji i większy margines zakłóceń układów CMOS należy zasilić je napięciem większym niż 5V. Sterowanie układów CMOS realizuje się wówczas za pomocą układów transformujących poziomy logiczne TTL na odpowiednio wyższe poziomy CMOS. Mogą być to specjalizowane układy sprzęgają (translatorzy) np. 40109B lub układy wykorzystujące bramki TTL z otwartym obwodem kolektora (OC) czy proste klucze tranzystorowe.



Rys. 10. Sprzężenie TTL z CMOS z użyciem a) bramki TTL typu OC, c) klucza tranzystorowego, d) translatora 40109B. [2,s.176]

Układy CMOS i TTL zasilane z jednego źródła napięcia mogą być łączone bezpośrednio choć przeciwskazaniem może być mała wydajność prądowa CMOS. Zwiększenie napięcia zasilania układów CMOS zwiększa ich wydajność prądową ale równocześnie likwiduje dopasowanie prądowe. Sprzężenia układów CMOS z TTL jest wówczas analogiczne jak w przypadku sprzężenia TTL z CMOS.



Rys. 11. Sprzężenie CMOS z TTL z użyciem a) bramki CMOS typu OD, b) klucza tranzystorowego, c) translatora 40115B[2,s.176]

4.5.2. Pytania sprawdzające

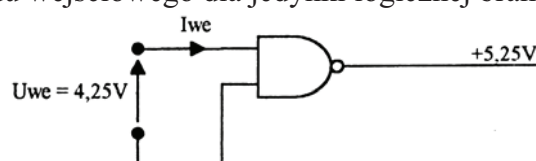
Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie znasz technologie wykonania cyfrowych układów scalonych?
2. Które z technologii produkcji układów cyfrowych są najczęściej wykorzystywane?
3. Jakie są zasady budowy oznaczeń scalonych układów cyfrowych polskiej produkcji?
4. Jakie znasz podstawowe parametry układów cyfrowych?
5. Co określają straty mocy w układach scalonych?
6. Jak definiowany jest czas propagacji dla cyfrowych układów scalonych?
7. Jakie są zasady sprzęgania układów TTL z układami CMOS?
8. Jakie są zasady sprzęgania układów TTL z układami CMOS?

4.5.3. Ćwiczenia

Ćwiczenie 1

Wykonaj pomiar prądu wejściowego dla jedynek logicznych bramki NAND TTL.



Schemat bramki NAND w stanie jedynek logicznych [5, s. 31]

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 3) przeanalizować zadanie,
- 4) narysować schemat układu pomiarowego na podstawie rysunku ,
- 5) skompletować potrzebne elementy i aparaturę pomiarową,
- 6) zbudować układ pomiarowy według zaproponowanego schematu,
- 7) dokonać pomiaru prądu wejściowego dla jedynki logicznej bramki NAND TTL.
- 8) oszacować dokładność pomiarów,
- 9) sformułować wnioski i sporządzić sprawozdanie.

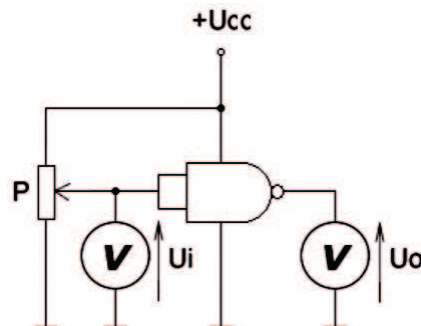
Wyposażenie stanowiska pracy:

- ¼ układu UCY 7400N,
- dwa zasilacze regulowane napięcia stałego 0V ÷ 10V,
- multimetr cyfrowy,
- stanowisko do łączenia układów pomiarowych,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

Ćwiczenie 2

Wyznacz charakterystykę przejściową bramki NAND TTL.

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.



Schemat układu do wyznaczania charakterystyki przejściowej $U_o = f(U_i)$ bramki NAND TTL

Tabela wyników pomiarów

U_i [V]													
U_o [V]													

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować zadanie,
- 2) skompletować potrzebne elementy i aparaturę pomiarową,
- 3) zbudować układ pomiarowy według zaproponowanego schematu,

- 4) dokonać pomiarów napięcia wejściowego U_i i wyjściowego U_o logicznej bramki NAND TTL, zmieniając wartość napięcia wejściowego w zakresie od 0V do 5V,
- 5) sporządzić charakterystykę przejściową $U_o = f(U_i)$ bramki NAND TTL,
- 6) oszacować dokładność pomiarów,
- 7) sformułować wnioski i sporządzić sprawozdanie.

Wyposażenie stanowiska pracy:

- ¼ układu UCY 7400N,
- zasilacz regulowany napięcia stałego 0V ÷ 10V,
- potencjometr 1kΩ,
- 2 multimetry cyfrowe,
- stanowisko do łączenia układów pomiarowych,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

Ćwiczenie 3

Wyjaśnij znaczenie następujących symboli: UCY84LS00N, UCY74LH00N.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) wyjaśnić znaczenia powyższych symboli,
- 3) sprawdzić w katalogu elementów elektronicznych poprawność analizy znaczeń,
- 4) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- katalogi elementów elektronicznych,
- literatura uzupełniająca zgodna z punktem 6.

4.5.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) scharakteryzować podstawowe technologie wykonywania scalonych układów cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
2) wymienić podstawowe parametry scalonych układów cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
3) wyjaśnić, co określa czas propagacji w scalonych układach cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
4) wyjaśnić, co określają straty mocy w scalonych układach cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
5) wyjaśnić, co określa margines zakłóceń w scalonych układach cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
6) wyjaśnić oznaczenia scalonych układów cyfrowych?	<input type="checkbox"/>	<input type="checkbox"/>
7) określić zasady sprzęgania układów TTL z układami CMOS?	<input type="checkbox"/>	<input type="checkbox"/>
8) określić zasady sprzęgania układów CMOS z układami TTL?	<input type="checkbox"/>	<input type="checkbox"/>

4.6. Układy kombinacyjne

4.6.1. Materiał nauczania

Układy kombinacyjne to układy cyfrowe, w których każda kombinacja sygnałów wejściowych określa jednoznacznie kombinację sygnałów wyjściowych.

Kombinacja sygnałów wejściowych jest nazywana stanem wejść układu lub słowem wejściowym. Kombinacja sygnałów wyjściowych jest nazywana stanem wyjść układu lub słowem wyjściowym.

Schemat logiczny układu kombinacyjnego można jednoznacznie opisać przy pomocy funkcji logicznych (funkcji przełączających, funkcji bulowskich). Układy kombinacyjne buduje się z bramek logicznych.

Działanie układów kombinacyjnych przedstawia się za pomocą:

- opisu słownego,
- tablicy prawdy,
- postaci kanonicznej.

Opis słowny jest pierwotną informacją o funkcjonowaniu układu, nie zawsze jest on jednoznaczny. Projektowanie układów kombinacyjnych na jego podstawie wymaga uściśleń.

Tablica prawdy jest wykorzystywana przy opisie działania bramek logicznych. Wiersze tablicy zawierają wszystkie kombinacje sygnałów wejściowych układu oraz odpowiadające im stany wyjść układu. Jeśli ze względu na fizyczne działanie urządzenia pewne stany wejściowe nie mogą zaistnieć, w tablicy prawdy wpisuje się „-”, który traktuje się przy analizie i minimalizacji funkcji zależnie od potrzeb jako 1 lub 0. Mówimy, że w tym punkcie funkcja jest nieokreślona.

Tablica prawdy to zwięzły i jednoznaczny opis funkcjonowania układu, bardzo przydatny przy jego projektowaniu.

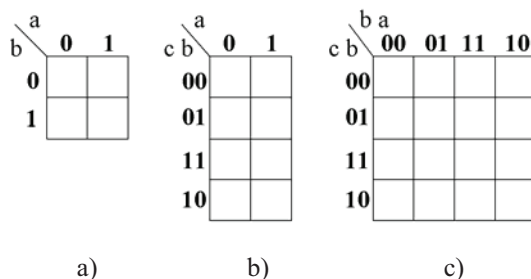
Postać kanoniczna jest umownym sposobem opisu obiektów matematycznych. W postaci tej stosuje się pełne iloczyny czyli iloczyn wszystkich argumentów funkcji i pełne sumy czyli sumy wszystkich argumentów funkcji.

Wyróżniamy kanoniczną postać sumy będącą sumą pełnych iloczynów funkcji dla których przyjmuje ona wartość 1 oraz kanoniczną postać sumy będącą sumą pełnych iloczynów dla których funkcja przyjmuje wartość 1 oraz kanoniczną postać iloczynu będącą iloczynem pełnych sum dla których funkcja przyjmuje wartość 0. Na podstawie postaci kanonicznej można budować układy kombinacyjne przy użyciu bramek, które jednak ze względu na rozbudowaną strukturę są drogie i zawodne.

Minimalizacja funkcji logicznych

W celu zaprojektowania ekonomicznego i niezawodnego układu kombinacyjnego, czyli zawierającego jak najmniejszą liczbę elementów i połączeń, stosuje się minimalizację funkcji opisującej jego działanie.

Funkcje logiczne można minimalizować stosując prawa algebry Boole'a. Jest to jednak sposób bardzo pracochłonny i mało efektywny. Istnieją uproszczone sposoby minimalizacji funkcji logicznej spośród których najpopularniejszą jest metoda graficzna – tablic Karnaugh. Stosuje się ją do minimalizacji funkcji maksymalnie 6 zmiennych. Wiersze i kolumny tablicy opisane są zmiennymi wejściowymi funkcji zakodowanymi w kodzie Graya. Każde pole tablicy odpowiada jednej kombinacji zmiennych wejściowych i zawiera wartość jaką przyjmuje funkcja dla tej kombinacji.



Rys.12. Tablice Karnauga a) dwóch zmiennych, b) trzech zmiennych, c) czterech zmiennych.

Minimalizacja funkcji metodą tablic Karnauga przebiega w trzech etapach:

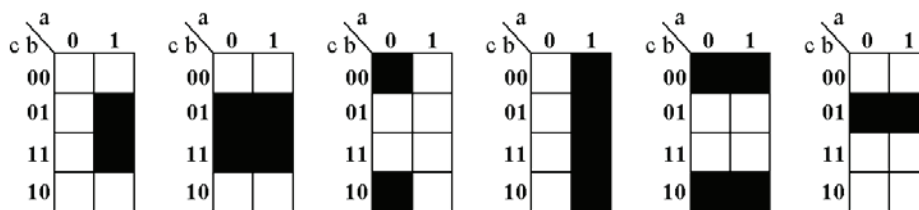
1. przygotowanie tablicy dla danej liczby zmiennych i wpisanie w jej pola wartości funkcji, często na tym etapie bardzo pomocna jest tablica prawdy, której wiersze odpowiadają odpowiednio opisanym polom tablicy Karnauga,
2. połączenie w grupy możliwie największych obszarów obejmujących wyłącznie jedynki lub wyłącznie zera logiczne, jeżeli sąsiadujące pola tablicy zawierające te same wartości (0 lub 1) to odpowiadające tym jedynkom (zerom) pełne iloczyny (pełne sumy) można skleić – co odpowiada usunięciu litery, która w ramach sklejonej grupy zmienia swoją wartość,
3. zapisanie funkcji
 - dla grup jedynek w postaci sumy iloczynów zmiennych wejściowych (jeden iloczyn odpowiada jednej grupie),
 - dla grup zer w postaci iloczynu sum zmiennych wejściowych (jedna suma odpowiada jednej grupie).

Zasady zakreślania grup w tablicy Karnauga:

1. liczba pól elementarnych łączonych ze sobą musi być potęgą liczby 2,
2. łączone ze sobą pola muszą być polami sąsiadującymi ze sobą, tzn. linią poziomą, pionową lub krawędziami tablicy,
3. połączone pola muszą mieć kształt symetryczny względem swych osi (kwadraty, prostokąty),
4. dla tablic 5 zmiennych obowiązuje zasada: jeśli zakreślone pola znajdują się w obu połówkach tablicy, to w wyniku złożenia tej tablicy względem osi dzielącej ją na dwie symetryczne części zakreślony obszar powinien się dwukrotnie zmniejszyć i spełniać zasadę określoną w punkcie 3,
5. jeśli w tablicy znajduje się „-” (funkcja jest nieokreślona), to pola takie można łączyć z jedynkami bądź z zerami.

Grupa 1 ba b $\bar{d} \bar{a}$ a \bar{d} $\bar{c} b$

Grupa 0 $\bar{d} + \bar{a}$ \bar{d} a + b \bar{a} b c + \bar{d}



Rys. 13. Przykłady sklejania w tablicy trzech zmiennych. [2, s. 49]

Projektowanie układów kombinacyjnych

Układ kombinacyjny powinien być projektowany tak, by zawierał jak najmniejszą liczbę elementów i połączeń, co zapewnia większą niezawodność i niskie koszty wykonania.

W celu zaprojektowania układu kombinacyjnego należy zrealizować kolejno trzy etapy:

1. określić funkcję logiczną rozpatrywanego problemu, może to być postać kanoniczna lub tablicy prawdy,
2. zminimalizować określoną funkcję logiczną wykorzystując np. tablice Karnaugh,
3. sporządzić schemat połączeń elementów logicznych układu.

Zjawiska szkodliwe w układach kombinacyjnych

Układy kombinacyjne posiadają określone właściwości przełączające i transmisyjne, które w stanach przejściowych mogą być przyczyną pojawiania się błędnych stanów na wyjściach. Zjawisko to nosi nazwę hazardu. Hazardy mogą być statyczne, gdy ich źródłem są nieidealne właściwości przełączające układu, lub dynamiczne, gdy ich źródłem są nieidealne właściwości transmisyjne.

Hazard statyczny występuje zawsze wtedy, gdy w tablicy Karnaugh funkcji, którą minimalizujemy, występują sąsiadujące ze sobą grupy. W celu jego wyeliminowania wprowadza się dodatkowe grupy w tablicy Karnaugh, zawierające pola wchodzące w skład sąsiadujących ze sobą grup. Taki zabieg likwiduje jednocześnie hazard dynamiczny.

		b a			
		00	01	11	10
c	0	1	1	0	0
	1	1	0	0	1

		b a			
		00	01	11	10
c	0	1	1	0	0
	1	1	0	0	1

Rys.14. Likwidacja zjawiska hazardu statycznego poprzez wprowadzenie dodatkowych grup w tablicy Karnaugh.

4.6.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Czym charakteryzuje się układ kombinacyjny?
2. Z jakich elementów zbudowane są układy kombinacyjne?
3. Jakie znasz sposoby opisu działania układów kombinacyjnych?
4. Jak zbudowana jest tablica prawdy?
5. Jakie znasz postacie kanoniczne funkcji?
6. Jakie są zasady minimalizacji funkcji za pomocą tablicy Karnaugh?
7. Jakie są etapy projektowania układów kombinacyjnych?
8. Jakie właściwości układów kombinacyjnych są źródłem hazardów?

4.6.3. Ćwiczenia

Ćwiczenie 1

Zminimalizuj za pomocą tablicy Karnaugh funkcję opisaną tablicą prawdy:

c	b	a	y
0	0	0	0
0	0	1	1
0	1	0	-
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) zbudować tablicę Karnaugh'a trzech zmiennych,
- 3) wypełnić jej pola zgodnie z tablicą prawdy,
- 4) zaznaczyć grupy zer lub jedynek,
- 5) zapisać postać funkcji y wynikający ze sklejania grup,
- 6) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Zaprojektuj układ kombinacyjny realizujący funkcję logiczną
 $y = abc + \bar{a}bc + a\bar{b}c + a\bar{c}\bar{b} + ab\bar{c}$

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

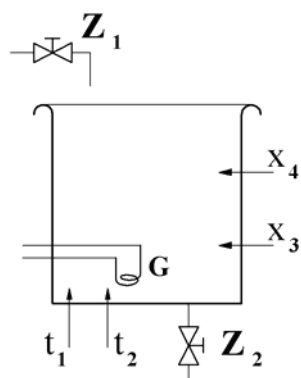
- 1) przeanalizować treść zadania,
- 2) stworzyć tablicę prawdy dla funkcji y ,
- 3) zbudować tablicę Karnaugh'a trzech zmiennych,
- 4) wypełnić jej pola zgodnie z tablicą prawdy,
- 5) zaznaczyć grupy zer lub jedynek,
- 6) zapisać postać funkcji y wynikający ze sklejania grup,
- 7) narysować schemat układu kombinacyjnego realizującego funkcję y ,
- 8) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zaprojektuj układ kombinacyjny realizujący sterowanie dwóch zaworów dwustanowych Z_1 i Z_2 oraz grzałki G zbiornika w którym podgrzewana jest woda. Do zbiornika doprowadzana jest woda zaworem Z_1 , poziom wody sygnalizowany jest za pomocą dwóch czujników x_3 dla poziomu minimalnego i x_4 dla poziomu maksymalnego. Temperaturę wody sygnalizują natomiast czujniki t_1 - temperatura minimalna i t_2 - temperatura maksymalna. Grzałka G może być załączona, gdy w zbiorniku jest minimalny poziom wody. Zawór Z_2 otwierany jest wówczas, gdy poziom wody jest większy od minimalnego a jej temperatura T spełnia zależność $t_1 < T < t_2$.



Schemat poglądowy zbiornika

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) stworzyć tablicę prawdy opisującą działanie układu,
- 3) zbudować oddzielne tablice Karnaugh dla Z_1 i Z_2 oraz G ,
- 4) zminimalizować funkcje Z_1 i Z_2 oraz G ,
- 5) narysować schematy układów kombinacyjnych realizujących funkcje Z_1 i Z_2 oraz G ,
- 6) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

4.6.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) stworzyć tablicę prawdy na podstawie opisu słownego działania układu kombinacyjnego?	<input type="checkbox"/>	<input type="checkbox"/>
2) zapisać postać kanoniczną sumy dla danej tablicy prawdy?	<input type="checkbox"/>	<input type="checkbox"/>
3) zapisać postać kanoniczną iloczynu dla danej tablicy prawdy?	<input type="checkbox"/>	<input type="checkbox"/>
4) stworzyć tablicę Karnaugh dla dowolnej ilości zmiennych?	<input type="checkbox"/>	<input type="checkbox"/>
5) sklejać grupy tablicy Karnaugh?	<input type="checkbox"/>	<input type="checkbox"/>
6) zapisać postać funkcji wynikającą ze sklejenia grup?	<input type="checkbox"/>	<input type="checkbox"/>
7) zlikwidować zjawisko hazardu w układach kombinacyjnych?	<input type="checkbox"/>	<input type="checkbox"/>

4.7. Przerzutniki. Układy sekwencyjne

4.7.1. Materiał nauczania

Przerzutniki

Przerzutniki to układy służące do zapamiętywania wartości wybranych zmiennych logicznych. Przerzutnik posiada co najmniej dwa wejścia i zazwyczaj dwa wyjścia Q i \bar{Q} . Wejścia mogą być:

- zegarowe zwane również synchronizującymi lub wyzwalającymi, oznaczone C (używane są też oznaczenia: CK , CL , CP , T , CLK),
- informacyjne,
- programujące: ustawiające w stan wysoki (oznaczone S lub PR), ustawiające w stan niski nazywane zerującym (oznaczone R lub CLR).

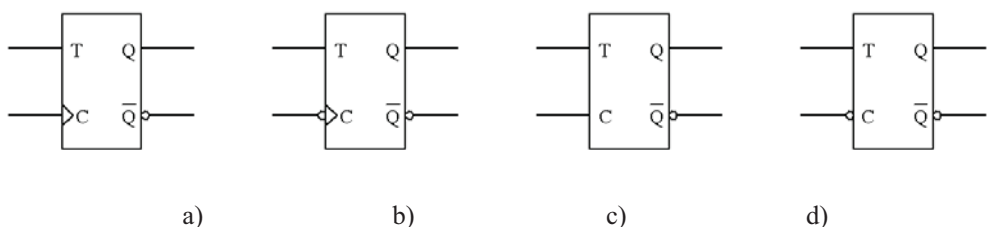
Działanie przerzutnika opisuje się za pomocą:

- tablicy przejść, która określa kolejny stan przerzutnika w zależności od aktualnego stanu jego wejść i wyjść;
- tablicy charakterystycznej, będącej bardziej skomplikowaną postacią tablicy przejść;
- tablicy wzbudzeń, która określa, jaki powinien być stan wejść informacyjnych, aby przerzutnik przeszedł z jednego stanu w drugi;
- wykresu czasowego.

Przerzutniki dzielimy na :

- asynchroniczne, które nie posiadają wejścia zegarowego,
- synchroniczne, które reagują na informacje podaną na wejścia informacyjne tylko w obecności aktywnego sygnału zegarowego (stan wejść informacyjnych powinien być wówczas ustalony).

Przerzutniki synchroniczne mogą być wyzwalone poziomem niskim lub wysokim sygnału zegarowego, albo zboczem jego opadającym bądź narastającym.



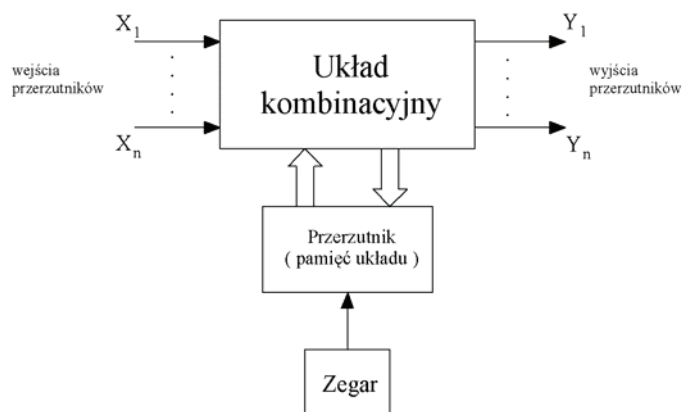
Rys. 15. Symbole przerzutników typu T wyzwanych a) zboczem narastającym, b) zboczem opadającym, c) poziomem wysokim, d) poziomem niskim. [2, s.123]

Przerzutniki służą do budowy układów sekwencyjnych między innymi liczników i rejestrów.

Układy sekwencyjne

Układy sekwencyjne są układami kombinacyjnymi wyposażonymi w pamięć. Stan wyjść w układzie sekwencyjnym zależy od słowa wejściowego oraz od poprzednich stanów wyjściowych. Elementem pamiętającym w układzie sekwencyjnym jest przerzutnik. Jednocześnie przerzutnik jest najprostszym układem sekwencyjnym.

Układy sekwencyjne dzielimy na synchroniczne i asynchroniczne. Stan układu synchronicznego zmienia się przy aktywnym sygnale synchronizującym, natomiast stan układu asynchronicznego pozostaje tak długo niezmienny jak długo nie zmienia się stan jego wejść.



Rys. 16. Schemat poglądowy sekwencyjnego układu synchronicznego. [2, s. 33]

Przerzutniki asynchroniczne

Przerzutnik $r\ s$

Przerzutnik $r\ s$ zbudowany jest z dwóch bramek NOR, posiada dwa wejścia s – ustawiające oraz r – zerujące. Aktywnym stanem logicznym tego przerzutnika jest jedynka. Wymuszenie jedynki na obu wejściach jest stanem zabronionym natomiast podanie zera na oba wejścia powoduje zapamiętanie informacji wpisanej do przerzutnika.

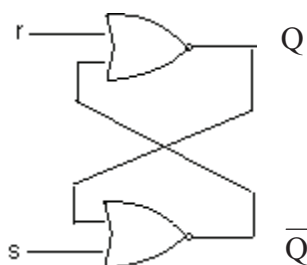


Tabela 1 stany stabilne przerzutnika $r\ s$

r	s	Q	Q̄	Stan stabilny
1	1	0	0	Stan wejść logicznie zabroniony
1	0	0	1	Przerzutnik ustawiony w stan 0
0	1	1	0	Przerzutnik ustawiony w stan 1
0	0	0	1	Pamiętanie stanu 0
0	0	1	0	Pamiętanie stanu 1

Rys. 17. Schemat logiczny przerzutnika $r\ s$. [2, s. 117]

Przerzutnik $\bar{r}\ \bar{s}$

Przerzutnik $\bar{r}\ \bar{s}$ zbudowany jest z dwóch bramek NAND, posiada dwa wejścia \bar{s} – ustawiające oraz \bar{r} – zerujące. Aktywnym stanem logicznym tego przerzutnika jest zero. Wymuszenie zera na obu wejściach jest stanem zabronionym natomiast podanie jedynki na oba wejścia powoduje zapamiętanie informacji wpisanej do przerzutnika.

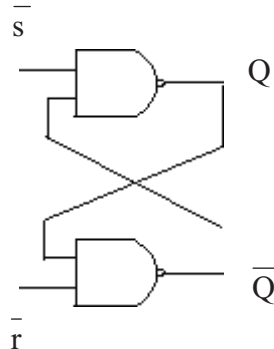


Tabela 2 stany stabilne przerzutnika $\bar{r}\ \bar{s}$

\bar{r}	\bar{s}	Q	Q̄	Stan stabilny
0	0	1	1	Stan wejść logicznie zabroniony
0	1	0	1	Przerzutnik ustawiony w stan 0
1	0	1	0	Przerzutnik ustawiony w stan 1
1	1	0	1	Pamiętanie stanu 0
1	1	1	0	Pamiętanie stanu 1

Rys. 18. Schemat logiczny przerzutnika $\bar{r}\ \bar{s}$ [2, s. 119]

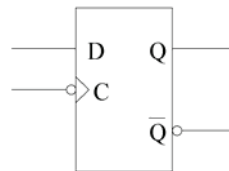
Przerzutniki synchroniczne

Przerzutnik synchroniczny typu D

Przerzutnik synchroniczny typu D posiada jedno wejście informacyjne D oraz wejście zegarowe. Przy aktywnym sygnale zegarowym informacja z wejścia D zostaje przepisana na wyjście Q.

Q aktualny stan wyjścia przerzutnika
 Q^+ następny stan wyjścia przerzutnika

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1



a)

b)

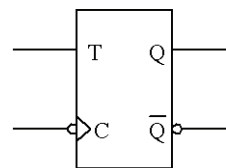
Rys. 19. Przerzutnik D wyzwalany zboczem opadającym a) tablica wzbudzeń, b) symbol graficzny. [2, s. 122]

Przerzutnik synchroniczny typu T

Przerzutnik synchroniczny typu T posiada jedno wejście informacyjne T oraz wejście zegarowe C. Zmiana stanu wyjścia przerzutnika na przeciwny następuje przy aktywnym sygnale zegarowym i jedynce logicznej na wejściu T.

Q aktualny stan wyjścia przerzutnika
 Q^+ następny stan wyjścia przerzutnika

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0



a)

b)

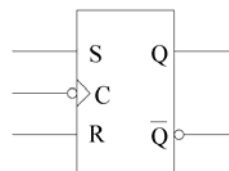
Rys. 20. Przerzutnik T wyzwalany zboczem opadającym a) tablica wzbudzeń, b) symbol graficzny. [3, s. 123]

Przerzutnik synchroniczny typu RS

Przerzutnik synchroniczny typu RS posiada dwa wejścia informacyjne S oraz R i wejście zegarowe C. Stan 11 na wejściach informacyjnych jest stanem zabronionym.

Q aktualny stan wyjścia przerzutnika
 Q^+ następny stan wyjścia przerzutnika

Q	Q^+	S R
0	0	0 -
0	1	1 0
1	0	0 1
1	1	- 0



a)

b)

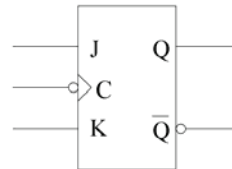
Rys. 21. Przerzutnik RS wyzwalany zboczem opadającym a) tablica wzbudzeń, b) symbol graficzny. [3, s. 123]

Przerzutnik synchroniczny typu JK

Przerzutnik synchroniczny typu JK posiada dwa wejścia informacyjne J oraz K i wejście zegarowe C. Wejście J = 1 ustawia przerzutnik w stan 1, natomiast wejście K = 1 ustawia przerzutnik w stan 0. Stan 11 na wejściach informacyjnych powoduje zmianę stanu wyjścia przerzutnika na przeciwny.

Q aktualny stan wyjścia przerzutnika
 Q⁺ następny stan wyjścia przerzutnika

Q	Q ⁺	JK
0	0	0 -
0	1	1 -
1	0	- 1
1	1	- 0



a)

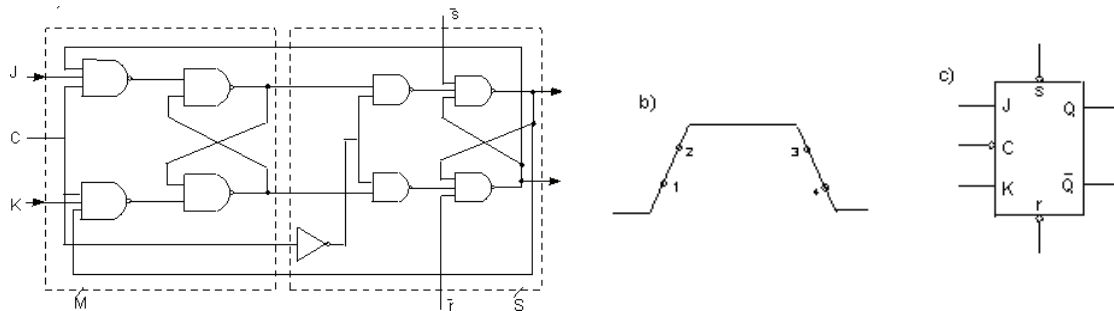
b)

Rys. 22. Przerzutnik JK wyzwalany zboczem opadającym a) tablica wzbudzeń, b) symbol graficzny. [2, s. 123]

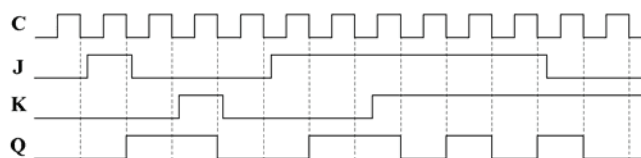
Przerzutnik dwutaktowy typu JK

Do ustawienia stanu przerzutnik dwutaktowego wymagane są dwa kolejne zbocza impulsu zegarowego. Przerzutnik JK zwany popularnie przerzutnikiem Master-Slave; składa się z dwóch przerzutników połączonych kaskadowo, pierwszy przerzutnik układu to część Master, przerzutnik drugi to część Slave. Działanie rzeczywistego przerzutnika Master-Slave można rozdzielić na cztery fazy. W punkcie 1 impulsu zegarowego następuje przerwanie połączenia między częścią Master i częścią Slave, z kolei w punkcie 2 zostają otwarte wejścia bramek dla sygnałów J i K; następuje zapamiętanie wejść części Master, wreszcie w punkcie 4 ma miejsce przepisanie wartości zmiennych wyjściowych przerzutnika Master do przerzutnika Slave, tzn. na wyjściu układu.

a)



Rys. 23. Przerzutnik JK Master –Slave a) schemat logiczny b) fazy przełączania c) symbol graficzny. [2, s. 131]



Rys. 24. Przykładowe przebiegi czasowe przerzutnika JK, Master-Slave.

Konwersja przerzutników

Każdy przerzutnik synchroniczny możemy zbudować z przerzutnika synchronicznego innego typu.

Algorytm konwersji danego przerzutnika na inny typ:

1. zapisanie tablicy przejść przerzutnika jaki chcemy uzyskać,
2. przekształcenie zapisanej tablicy przejść w tablicę Karnaugh'a,
3. wpisanie w pola tablicy Karnaugh'a (w miejsce wartości logicznych wejścia informacyjnego projektowanego przerzutnika) wartości logicznych podawanych na wejścia informacyjne przerzutnika jakim dysponujemy zapewniających określoną zmianę stanu wyjścia,
4. minimalizacja powstałej tablicy Karnaugh'a.

Na podstawie wyniku minimalizacji tablicy Karnaugh'a można narysować układ logiczny realizujący konwersję przerzutnika.

4.7.2. Pytania sprawdzające

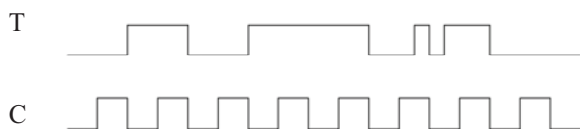
Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. W jakich układach wykorzystywane są przerzutniki?
2. Jakie wejścia posiadają przerzutniki?
3. Jak dzielimy przerzutniki?
4. Czym charakteryzują się układy sekwencyjne?
5. Jak działa przerzutnik asynchroniczny sr?
6. Jak działa przerzutnik synchroniczny D?
7. Jak działa przerzutnik synchroniczny T?
8. Co oznacza nazwa przerzutnik dwutaktowy?

4.7.3. Ćwiczenia

Ćwiczenie 1

Narysuj przebieg na wyjściu przerzutnika synchronicznego T wyzwalanego zboczem opadającym, na podstawie przebiegu sygnału na wejściu informacyjnym T oraz wejściu zegarowym C.



Przebiegi sygnałów na wejściu informacyjnym T oraz wejściu zegarowym C przerzutnika synchronicznego T.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) napisać tablicę wzbudzeń przerzutnika T,
- 3) narysować przebieg wyjściowy przerzutnika, na podstawie przebiegu sygnału na wejściu informacyjnym T oraz wejściu zegarowym C i tablicy wzbudzeń,
- 4) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,

- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Dokonaj konwersji przerzutnika synchronicznego typu JK w przerzutnik typu D.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) zapisać tablicy wzbudzeń przerzutnika typu D,
- 3) przekształcić tablicę wzbudzeń przerzutnika D w tablicę Karnaugh,
- 4) wpisać w pola tablicy Karnaugh wartości logicznych podawanych na wejścia informacyjne przerzutnika JK zapewniające określoną zmianę stanu wyjścia,
- 5) zminimalizować tablicę Karnaugh,
- 6) narysować układ logiczny realizujący konwersję przerzutnika JK w przerzutnik D, na podstawie wyniku minimalizacji tablicy Karnaugh,
- 7) zeprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zbadaj działanie przerzutnika asynchronicznego $\bar{r} \bar{s}$.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) narysować schemat logiczny przerzutnika asynchronicznego $\bar{r} \bar{s}$,
- 3) zapisać tablicę wzbudzeń przerzutnika asynchronicznego $\bar{r} \bar{s}$,
- 4) narysować schemat układu do badania działania przerzutnika asynchronicznego $\bar{r} \bar{s}$,
- 5) zmontować układ przerzutnika asynchronicznego $\bar{r} \bar{s}$,
- 6) zmontować układ do badania przerzutnika asynchronicznego $\bar{r} \bar{s}$ według zaproponowanego schematu,
- 7) zmierzyć napięcie na wyjściach przerzutnika zmieniając na wejściu stany logiczne zgodnie z tablicą wzbudzeń,
- 8) sformułować wnioski i sporządzić sprawozdanie z ćwiczenia.

Wyposażenie stanowiska pracy:

- 2 bramki NAND (z układu UCY 7400N),
- zasilacz napięcia stałego +5V,
- 2 multimetry cyfrowe,
- stanowisko do łączenia układów,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

4.7.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) narysować schemat logiczny przerzutnika asynchronicznego rs?	<input type="checkbox"/>	<input type="checkbox"/>
2) opisać działanie przerzutnika asynchronicznego rs?	<input type="checkbox"/>	<input type="checkbox"/>
3) rozróżnić typ przerzutnika na podstawie symbolu graficznego?	<input type="checkbox"/>	<input type="checkbox"/>
4) napisać tablicę przejść lub wzbudzeń dla dowolnego przerzutnika asynchronicznego?	<input type="checkbox"/>	<input type="checkbox"/>
5) narysować przebieg wyjściowy przerzutnika na podstawie przebiegów na jego wejściach informacyjnych i zegarowych?	<input type="checkbox"/>	<input type="checkbox"/>
6) wykonać konwersję dowolnego typu przerzutnika w inny typ?	<input type="checkbox"/>	<input type="checkbox"/>

4.8. Układy komutacyjne i arytmetyczne

4.8.1. Materiał nauczania

Układy komutacyjne

Układy komutacyjne to układy kombinacyjne umożliwiające przełączanie sygnałów cyfrowych. Do układów komutacyjnych zaliczamy multipleksery, demultipleksery oraz niektóre przetworniki kodów.

Multiplekser służy do wyboru jednego z sygnałów wejściowych i przekazania go na wyjście układu. Multiplekser posiada :

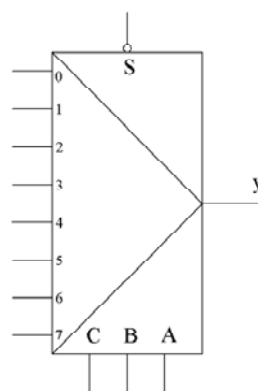
- N wejść informacyjnych (danych), na które podawane są sygnały, które należy przełączać;
- n wejść adresowych (sterujących), sygnały podane na te wejścia określają numer wejścia z którego sygnał przeniesiony zostanie na wyjście;
- wejście zezwalające (strobujące), podanie aktywnego sygnału na nie, umożliwia działanie multipleksera;
- jedno wyjście.

Adresowanie wejść informacyjnych realizowane jest w naturalnym kodzie binarnym zatem ilość N wejść informacyjnych multipleksera powiązana jest z ilością n wejść adresowych zależnością:

$$N = 2^n$$

wejścia adresowe			nr wejścia danych, z którego sygnał pojawi się na wyjściu
C	B	A	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

a)



b)

Rys. 25. Multiplekser ośmiowyjściowy a) tablica działania, b) symbol graficzny multipleksera. [2, s. 201]

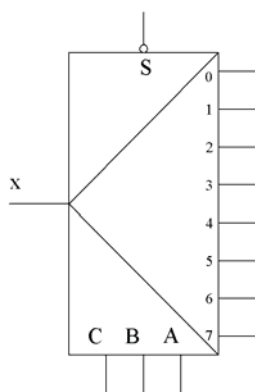
Demultiplekser umożliwia przekazanie sygnału z wejścia informacyjnego do jednego z jego wyjść. Demultiplekser posiada :

- jedno wejście informacyjne;
- N wyjść;
- n wejść adresowych (sterujących), sygnały podane na te wejścia określają numer wyjścia na które sygnał zostanie przeniesiony sygnał z wejścia;
- wejście zezwalające (strobujące), podanie aktywnego sygnału na nie, umożliwia działanie demultipleksera.

Adresowanie wyjść realizowane jest w naturalnym kodzie binarnym, zatem ilość N wyjść demultipleksera powiązana jest z ilością n wejść adresowych zależnością:

$$N = 2^n$$

wejścia adresowe			nr wyjścia							
C	B	A	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



a)

b)

Rys. 26. Demultiplexer ośmiowejściowy a) tablica działania, b) symbol graficzny. [2, s. 203]

Multiplexery i demultiplexery służą do realizacji funkcji logicznych, adresowania pamięci i innych układów w systemach cyfrowych.

Przetworniki kodów inaczej zwane konwerterami kodów dokonują zamiany kodów.

Układ realizujący zamianę informacji w kodzie 1 z n na kod wewnętrzny urządzenia nazywa się koderem (enkoderem).

Układu zamieniający dowolny kod na kod 1 z n nazywa się dekoderem.

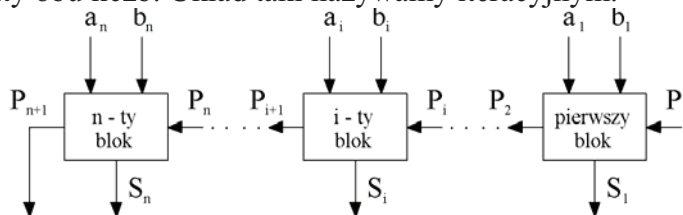
Układu zamieniający jeden kod wewnętrzny urządzenia na inny (żaden z nich nie jest kodem 1 z n) nazywa się transkoderem.

Układy arytmetyczne

Układy cyfrowe umożliwiające realizację podstawowych działań arytmetycznych nazywamy układami arytmetycznymi.

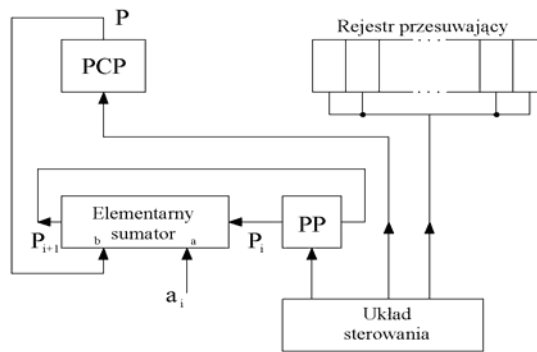
Sumatory to podstawowe układy arytmetyczne, które prócz dodawania mogą po zastosowaniu dodatkowych przekształceń realizować mnożenie, odejmowanie i dzielenie. Rozróżniamy sumatory równoległe i szeregowe.

Sumatory równoległe umożliwiają dodawanie dwóch liczb n-bitowych, ponieważ zbudowane są z n elementarnych sumatorów połączonych kaskadowo, z których każdy dodaje dwa odpowiednie bity obu liczb. Układ taki nazywamy iteracyjnym.



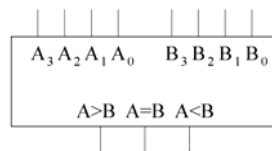
Rys. 27. Schemat blokowy sumatora równoległego. [2, s. 61]

W sumatorze szeregowym dodawane składniki są podawane na wejście kolejno. Układ ten posiada pamięć oraz układ sterowania. Podstawowy cykl pracy sumatora szeregowego polega na dodaniu dwóch bitów i określeniu wyniku sumowania i bitu przeniesienia. Powtarzany n-razy umożliwia dodawanie n-bitowych liczb. Pamięć układu zrealizowana z rejestrów przesuwających powinna być tak rozbudowana, aby pomieściła kolejne wyniki kolejnych operacji dodawania. Sumator szeregowy może dodawać ciągle nowe składniki (gromadzić – akumulować), dlatego nazywany jest akumulatorem.



Rys. 28. Schemat funkcjonalny sumatora elementarnego w układzie akumulatora PCP - pamięć cyklu, podstawowego PP – pamięć przeniesienia [2, s. 123]

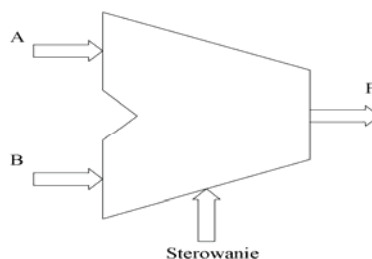
Komparator służy do porównania dwóch liczb. Układ posiada dwa wejścia n-bitowe, na które podawane są porównywane liczby A i B oraz trzy wyjścia ($A=B$), ($A>B$), ($A<B$) gdzie sygnalizowany jest wynik porównania. Komparatory realizuje się jako układy iteracyjne.



Rys. 29. Symbol graficzny komparatora czterobitowego. [2, s. 233]

Jednostka arytmetyczno-logiczna ALU jest blokiem funkcjonalnym przeznaczonym do wykonywania operacji arytmetycznych i logicznych na liczbach n-bitowych. Operacje logiczne są realizowane przez ALU na odpowiadających sobie parach bitów słów wejściowych. Układ jednostki arytmetyczno-logicznej posiada:

- wejścia wyboru funkcji – ustawione na nich słowo umożliwia zrealizowanie odpowiadającej mu funkcji,
- wejście rodzaju funkcji – wartości logiczne na tym wejściu decydują, czy będą realizowane funkcje logiczne czy mieszane,
- wejścia danych dla dwóch liczb n-bitowych,
- wyjścia wyniku, gdzie generowany jest wynik operacji,
- wyjście przeniesienia używane przy realizacji funkcji arytmetycznych,
- wyjście komparatora sygnalizujące równość liczb podanych na wejścia danych,
- wyjścia wykorzystywane do przyspieszania pracy – działa przy kaskadowym połączeniu kilku układów ALU.



Rys. 30. Symbol graficzny ALU [2, s. 233]

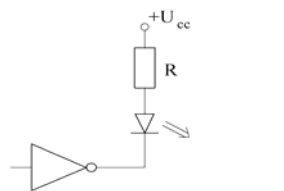
Multiplikatory to układy arytmetyczne realizujące mnożenie, natomiast komparatory porównują liczby dwójkowe.

Układy wyjściowe

Informacja wyjściowa układów cyfrowych jest przetwarzana i służy doysterowania pewnych obiektów, bardzo często jest wizualizowana za pomocą układów wyświetlania zbudowanych z diod świecących lub wskaźników ciekłokrystalicznych. W przypadku diod

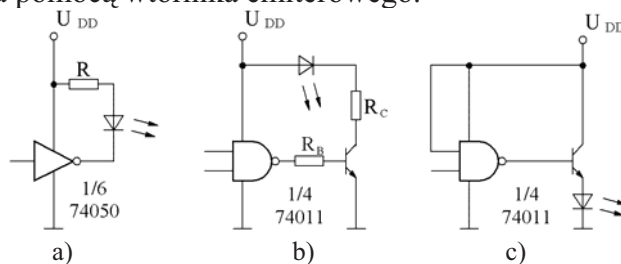
LED moc sygnałów wyjściowych z elementów TTL, czy MOS jest zbyt mała by jeysterować.

Wskaźniki elektroluminescencyjne to najczęściej pojedyncze diody LED lub zbudowane z nich wskaźniki alfanumeryczne. Do sterowania diod świecących można użyć dowolnych bramek TTL z wyjściem przeciwsobnym lub otwartym kolektorem. Należy pamiętać, że sterując diodą LED ze źródła napięcia stałego należy włączyć szeregowo z nią rezystor ograniczający prąd. Wartość rezystora należy dobrać tak, aby prąd płynący przez diodę nie przekraczał wartości jej prądu przewodzenia i dopuszczalnej wartości prądu wyjściowego bramki.



Rys. 31. Układ sterowania diody LED z bramek TTL z wyjściem przeciwsobnym lub otwartym kolektorem. [2, s.184]

W przypadku układów CMOS charakteryzujących się niskim prądem wyjściowym, wymaga się zastosowania bufora np. układu MCY74050. W układzie tym również stosuje się rezystor ograniczający prąd płynący przez diodę. Elementy wskaźników siedmiosegmentowych ze wspólną anodą mogą być sterowane za pomocą wzmacniacza tranzystorowego, a ze wspólną katodą za pomocą wtórnika emiterowego.

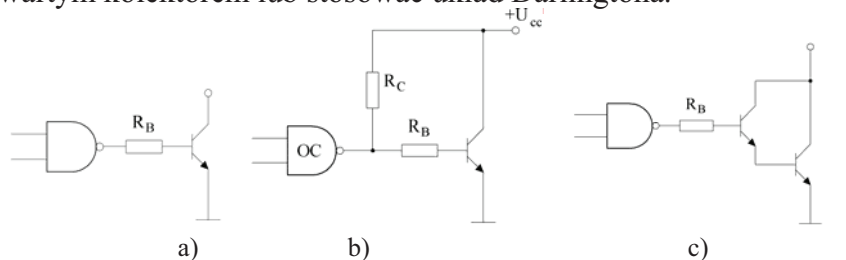


Rys. 32. Układ sterowania diody LED z układu CMOS za pomocą a) bufora b) wzmacniacza tranzystorowego, c) wtórnika emiterowego.

Wskaźniki ciekłokrystaliczne wymagają znikomo małych mocy sygnałów sterujących w porównaniu z mocą sygnałów wyjściowych elementów TTL czy CMOS.

Sterowanie tranzystora bipolarnego z układów TTL i MOS

Ze względu na małą moc sygnałów wyjściowych elementów TTL i CMOS układy cyfrowe nie są przystosowane do sterowania odbiorników dużej mocy np. grzałek czy silników. Dlatego bezpośrednio z wyjść układów cyfrowych sterowane są układy wzmacniające takie jak np. tranzystory. W obwodzie bazy umieszcza się rezystor ustalający prąd bazy. Dla uzyskania większych prądów bazy, a zatem i większych prądów wyjściowych umożliwiających sterowanie urządzeniami większej mocy można stosować bramki z otwartym kolektorem lub stosować układ Darlingtona.



Rys. 33. Sterowanie tranzystora bipolarnego z bramki TTL (MOS) a) z wyjściem przeciwsobnym, b) typu OC, c) sterowanie wzmacniacza tranzystorowego w układzie Darlingtona. [2, s.186]

Sterowanie przełącznika półprzewodnikowego z układów TTL i CMOS

Przełączniki półprzewodnikowe mogą być sterowane bezpośrednio z układów TTL i CMOS ze względu na niski poziom wymaganego prądu. Zapewniają one separację galwaniczną układu sterującego i sterowanego.

4.8.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Do czego służą układy komutacyjne?
2. Jak działa multiplekser?
3. Jak działa demultiplekser?
4. Czym różni się koder od transkodera?
5. Co to znaczy, że układ jest iteracyjnym?
6. Czym różni się sumator szeregowy od równoległego?
7. Jaką operację realizuje komparator?
8. Jakie operacje może realizować jednostka arytmetyczno-logiczna?

4.8.3. Ćwiczenia

Ćwiczenie 1

Zaprojektuj, wykorzystując multiplekser 16-wejściowy, układ kombinacyjny realizujący następującą funkcję logiczną $f(d,c,b,a) = \Sigma(1,4,6,9,11,13,15)$.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) napisać tablicę prawdy funkcji $f(d,c,b,a)$,
- 3) narysować układ logiczny z wykorzystaniem multipleksera realizujący tę funkcję,
- 4) zaprezentować wyniki pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Zaprojektuj sumator dwóch liczb dwubitowych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) napisać tablicę prawdy funkcji realizującej dodawanie dwóch liczb dwubitowych,
- 3) przekształcić ją w tablice Karnaugh'a,
- 4) zminimalizować uzyskane tablice Karnaugh'a,
- 5) narysować układ logiczny realizujący uzyskaną funkcję,
- 6) zaprezentować wyniki pracy.

- Wyposażenie stanowiska pracy:
- zeszyt,
 - literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zaprojektuj, a następnie zbadaj działanie prostego kodera 2-bitowego kodu binarnego na kod 1 z 4, używając dwuwejściowych bramek NAND.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinienes:

- 1) przeanalizować treść zadania,
- 2) zapisać tablicę działania kodera 2-bitowego kodu binarnego na kod 1 z 4,
- 3) narysować schemat ideowy kodera,
- 4) zmontować układ kodera z bramek dwuwejściowych NAND,
- 5) zbadać działanie kodera próbnikiem stanów logicznych, podając na wejścia sygnały zgodnie z tablicą działania kodera,
- 6) sformułować wnioski i sporządzić sprawozdanie z ćwiczenia.

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- 3 układy UCY 7400N,
- zasilacz napięcia stałego +5V,
- próbnik stanów logicznych,
- stanowisko do łączenia układów,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

4.8.4. Sprawdzian postępów

Czy potrafisz:	Tak	Nie
1) zapisać tablice działania multiplexera i demultiplexera?	<input type="checkbox"/>	<input type="checkbox"/>
2) narysować symbole graficzne multiplexera i demultiplexera?	<input type="checkbox"/>	<input type="checkbox"/>
3) opisać działanie kodera?	<input type="checkbox"/>	<input type="checkbox"/>
4) wyjaśnić zasadę działania sumatora równoległego?	<input type="checkbox"/>	<input type="checkbox"/>
5) opisać działanie jednostki arytmetyczno-logicznej?	<input type="checkbox"/>	<input type="checkbox"/>
6) narysować układ sterowania diody LED sygnałem wyjściowym z bramki TTL?	<input type="checkbox"/>	<input type="checkbox"/>
7) narysować układ sterowania diody LED sygnałem wyjściowym z bramki CMOS?	<input type="checkbox"/>	<input type="checkbox"/>
8) narysować układ sterowania elementem wzmacniającym sygnałem wyjściowym z bramki TTL?	<input type="checkbox"/>	<input type="checkbox"/>
9) wyjaśnić dlaczego wyświetlacze ciekłokrystaliczne mogą być sterowane bezpośrednio z wyjścia układu cyfrowego?	<input type="checkbox"/>	<input type="checkbox"/>

4.9. Liczniki i rejestry. Sterowanie układami cyfrowymi

4.9.1. Materiał nauczania

Liczniki

Licznik jest sekwencyjnym układem czasowym, służącym do zliczania i zapamiętywania liczby impulsów podawanych na jego wejście zliczające. Podstawowym parametrem licznika jest pojemność określająca ilość impulsów N jakie może on zliczyć.

$$N \leq 2^n$$

n określa ilość przerzutników użytych do budowy licznika a tym samym ilość jego wyjść i nazywana jest długością licznika. Liczniki o różnych długościach można łączyć kaskadowo, wówczas otrzymuje się licznik, którego pojemność jest iloczynem pojemności połączonych liczników. Stan licznika, od którego rozpoczyna się zliczanie, nazywamy stanem początkowym licznika.

Licznik który, przechodzi wszystkie stany cyklicznie nazywamy modulo N , natomiast taki, który przechodzi przez wszystkie stany jednokrotnie i pozostaje w stanie ostatnim nazywamy licznikiem do N .

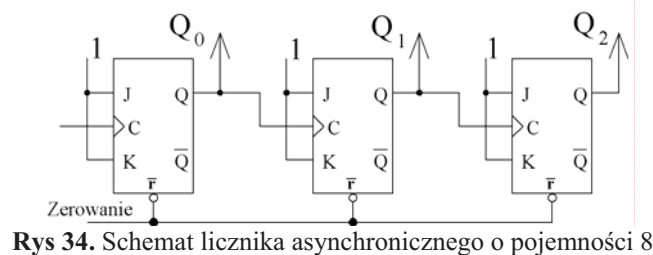
Rozróżniamy liczniki:

- dodające, czyli liczące w przód, które po każdym impulsie zwiększają zapamiętaną liczbę o jeden,
- odejmujące czyli liczące w tył, które po każdym impulsie zmniejszają zapamiętaną liczbę o jeden,
- dwukierunkowe, czyli liczące w przód i w tył, zwane rewersyjnymi.

Ze względu na sposób wprowadzania impulsów zliczanych liczniki dzielimy na:

- szeregowe, czyli asynchroniczne,
- równoległe, czyli synchroniczne.

Liczniki asynchroniczne zbudowane są z przerzutników synchronicznych połączonych kaskadowo. Impulsy zliczane podawane są na wejście zegarowe pierwszego przerzutnika, a poszczególne przerzutniki zmieniają swoje stany kolejno (wejście zegarowe kolejnego przerzutnika jest wysterowane wyjściem przerzutnika poprzedniego).



Rys 34. Schemat licznika asynchronicznego o pojemności 8

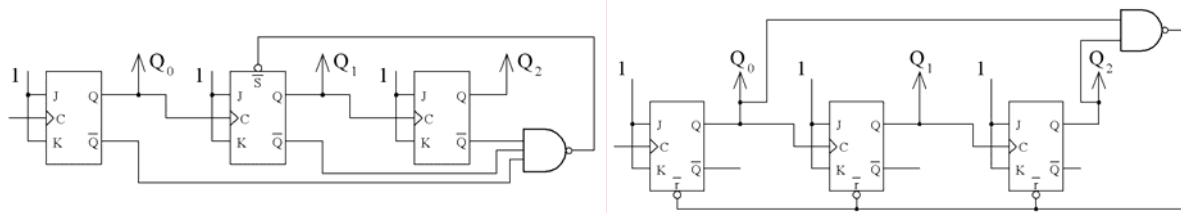
W celu uzyskania licznika o pojemności $N < 2^n$ można skrócić cykl liczenia, ustawić warunek początkowy różny od 0 (maksymalnej ilości impulsów dla licznika liczącego w tył) lub opuścić wybrane stany licznika. Dwa pierwsze sposoby są najpopularniejszymi rozwiązaniami stosowanymi w licznikach szeregowych.

Aby skrócić cykl liczenia należy:

- dla zadanej pojemności N połączyć w kaskadę n przerzutników,
- z ostatniej sekwencji stanów odpowiadającej zadanej pojemności N , utworzyć sygnał sprzężenia zwrotnego zerujący licznik (podawany na wejścia zerujące wszystkich przerzutników).

Aby ustawić warunek początkowy różny od 0 (maksymalnej ilości impulsów dla licznika liczącego w tył) należy:

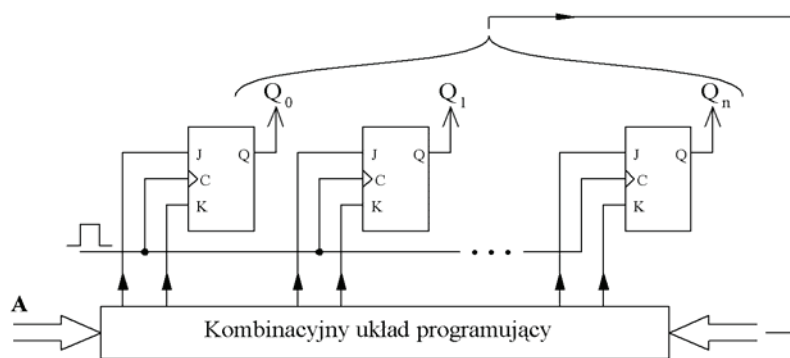
- dla zadanej pojemności N połączyć w kaskadę n przerzutników,
- z ostatniej sekwencji stanów odpowiadającej utworzyć sygnał sprzężenia zwrotnego ustawiający stan początkowy licznika (podawany na wejścia ustawiające określonych przerzutników).



Rys 35. Schemat licznika asynchronicznego liczącego w przód, [5, s. 117], a) z niezerowym warunkiem początkowym b) ze skróconym cyklem liczenia o pojemności. [5, s. 116]

Liczniki szeregowe są proste do zaprojektowania natomiast ich podstawową wadą jest długi czas propagacji sygnału.

Liczniki synchroniczne zbudowane są z przerzutników synchronicznych i kombinacyjnego układu programującego, określającego funkcje informacyjnych zmiennych wejściowych. Zmiana stanów poszczególnych przerzutników odbywa się jednocześnie w takt impulsów wprowadzanych równoległe do wejścia zegarowego każdego przerzutnika.



Rys 36. Schemat blokowy licznika synchronicznego. [5, s.120]

Projektowanie liczników synchronicznych składa się z następujących etapów:

- stworzenia tablicy wzbudzeń licznika,
- stworzenia tablicy Karnaugh dla wejścia informacyjnego każdego przerzutnika,
- minimalizacji tablic Karnaugh i określeniu na tej podstawie funkcji logicznej określającej wejścia informacyjne,
- narysowaniu schematu licznika (wraz z kombinacyjnym układem programującym).

Rejestry

Rejestry to układy składające się z zespołu przerzutników służące do przechowywania informacji w systemie cyfrowym. Jeżeli rejestr zbudowany jest z przerzutników synchronicznych to nazywamy go synchronicznym, natomiast rejestry zbudowane z przerzutników asynchronicznych noszą nazwę asynchronicznych.

Podstawowe parametry rejestrów to :

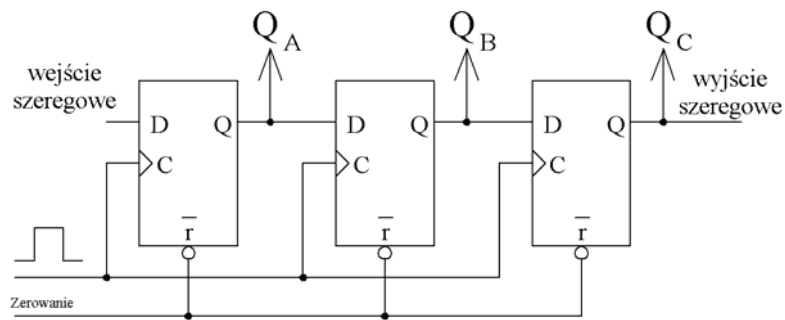
- długość rejestru określająca liczbę bitów, jaka może być przechowywana w rejestrze - jest ona zawsze równa liczbie przerzutników, z których zbudowany jest rejestr,
- szybkość pracy rejestru.

Ze względu na sposób wprowadzania i wyprowadzania informacji rejestry dzielimy na:

- równoległe (PIPO), w których wprowadzanie i wyprowadzanie informacji odbywa się równoległe,

- równoległo-szeregowe (PISO), w których wprowadzanie informacji odbywa się równoległe a wyprowadzanie szeregowo,
- szeregowo-równoległe (SIPO), w których wprowadzanie informacji odbywa się szeregowo a wyprowadzanie równoległe,
- szeregowe (SISO), w których wprowadzanie i wyprowadzanie informacji odbywa się szeregowo.

W rejestrach, w których informacja wprowadzana jest lub wyprowadzana szeregowo, zachodzi konieczność jej przesuwania w prawo lub w lewo są to rejestry przesuwające. Rejestry, które umożliwiają przesuwanie informacji w obu kierunkach to rejestry rewersyjne.



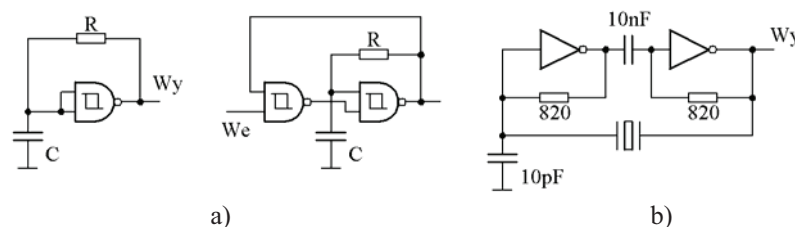
Rys. 37. Rejestr zbudowany z trzech przerzutników D z wejściem szeregowym oraz wyjściami równoległymi i szeregowym. [5, s.254]

Sterowanie układów cyfrowych

Układy czasowe służące do sterowania układami cyfrowymi to przerzutniki monostabilne i astabilne zwane też multiwibratorami. Generują one pojedynczy impuls lub falę prostokątną. Szerokość generowanych impulsów można regulować dołączając do układu zewnętrzne elementy RC. Układy czasowe wykonywane są w postaci układów scalonych. Przerzutniki monostabilne charakteryzują się jednym stanem równowagi stałej co oznacza, że wytwarzają pojedynczy impuls i wracają do poprzedniego stanu, natomiast przerzutniki astabilne generują falę prostokątną. Powszechnie stosowane są programowalne układy czasowe zawierające generatory, liczniki, multipleksery i układy sterujące, które odpowiednio zaprogramowane pracują jako multiwibratory monostabilne i astabilne.

Generatory fali prostokątnej są często zbudowane na jednym lub kilku przerzutnikach monostabilnych i innych elementach logicznych. Proste generatory zrealizowane na bramkach z układem Schmitta umożliwiają generowanie fali prostokątnej o częstotliwościach od 1 do 10 MHz.

Jeżeli generowany przebieg musi cechować się dużą stabilnością częstotliwości stosuje się generatory z oscylatorami kwarcowymi.

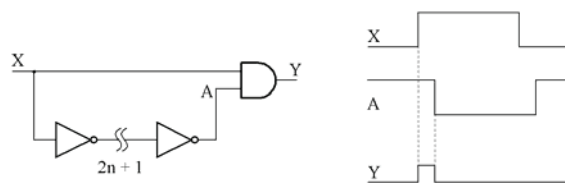


Rys. 38. Układy generatorów fali prostokątnej a) zbudowane z bramek Schmitta, b) z rezonatorem kwarcowym.

Często stosowane są też generatory o programowalnej częstotliwości wyposażone w programowalny dzielnik częstotliwości (będący programowalnym licznikiem).

Układy wyzwajające generują krótkie impulsy przy zmianie sygnału wejściowego i stosowane są w układach cyfrowych do zerowania lub ustawiania stanu początkowego.

Ze względu na charakter generowanego impulsu układy te nazywane są różniczkującymi. Układy takie mogą być zbudowane z samych bramek lub bramek i przerzutników.



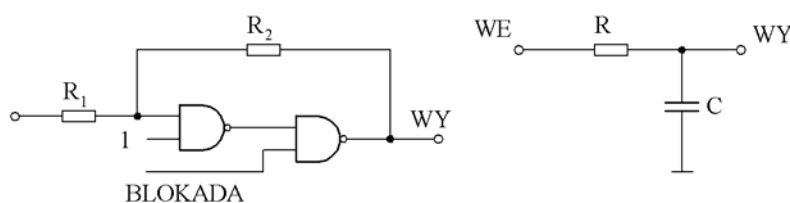
Rys. 39. Układ różniczkujący a) schemat logiczny, b) przebiegi czasowe. [2, s.159]

Układy cyfrowe często przetwarzają sygnały, których źródłem są zestyki przełączników, przekaźników, czujników z analogowymi sygnałami wyjściowymi itp.

Układy wejściowe służą dostosowaniu parametrów sygnałów wejściowych do wymagań układów cyfrowych. Podstawowe układy wejściowe to układy formowania i regeneracji sygnałów stosowane w celu:

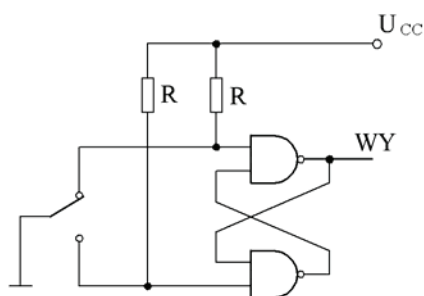
- kształtowania przebiegów prostokątnych,
- tłumienia zakłóceń.

Układy filtrujące zakłócenia to detektory sygnałów użytecznych zawierające układy dyskryminatora amplitudy oraz obwody całkujące. Dyskryminator amplitudy filtruje zakłócenia o amplitudzie mniejszej niż określony poziom, natomiast układ całkujący tłumi sygnały krótkotrwałe. Układy formowania i regeneracji sygnałów najczęściej realizuje się z wykorzystaniem przerzutnika Schmitta lub standardowych bramek logicznych.



Rys. 40. Schematy a) dyskryminatora, b) układu całkującego. [2, s.178]

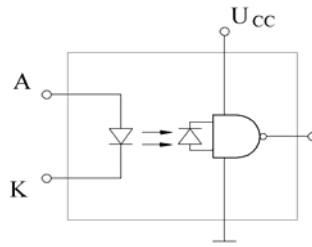
W przypadku współpracy układu cyfrowego z zestykami przekaźników istnieje problem nieprawidłowych stanów na wyjściach tych układów spowodowanych drganiami zestyków podczas przełączania. W celu wyeliminowania wpływu tego zjawiska na układ cyfrowy stosuje układy dyskryminatora i filtry dolnoprzepustowe. Układem współpracującym z zestykiem przełączającym może być też przerzutnik asynchroniczny.



Rys. 41. Schemat układu współpracującego z zestykiem przełączającym z przerzutnikiem \bar{r} s. [2, s.180]

W celu rozdzielania poszczególnych części układu cyfrowego stosuje się galwaniczną separację oznaczającą brak metalicznego połączenia między nimi, a zatem eliminuje zakłócenia powstające we wspólnych układach zasilania. Separację galwaniczną realizują przekaźniki, transformatory, a w układach cyfrowych przede wszystkim transoptory. Są to układy scalone zawierające sprzężony optycznie i odizolowany galwanicznie układ

składający się z diody LED, detektora promieniowania (fotodiody lub fotorezystora) i ośrodka przewodzącego promieniowanie.



Rys. 42. Schemat transoptora typu LED-bramka logiczna. [2, s.182]

Transmisja sygnałów cyfrowych

Sygnaly cyfrowe mogą być przesyłane :

- jedнопроводово np. ścieżką obwodu drukowanego,
- skręcona parą przewodów,
- kablem koncentrycznym.

Podstawowym problemem transmisji sygnałów cyfrowych występującym, w tak zwanych połączeniach długich, jest dłuższy czas propagacji sygnału niż czas jego przełączania. Ze względu na ten parametr dobiera się dopuszczalne długości połączeń.

W połączeniach długich występują też oscylacje, spowodowane odbiciami od końców przewodu (przy braku dopasowania). Oscylacje te zakłócają działanie układu cyfrowego i ograniczają jego szybkość działania.

Powszechnie stosowane są dwa typy transmisji sygnałów cyfrowych:

- za pośrednictwem linii niesymetrycznych (kable koncentryczne) – na niewielkie odległości ze względu na wrażliwość na zakłócenia,
- za pośrednictwem linii symetrycznych (para przewodów) – na duże odległości, gdy system nadawczy i odbiorczy mają potencjały odniesienia, ten typ transmisji jest odporny na zakłócenia, lecz kosztowny.

4.9.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jakie są podstawowe parametry liczników?
2. Czym różnią się liczniki asynchroniczne od synchronicznych?
3. Jakie jest przeznaczenie rejestrów?
4. Jak klasyfikujemy rejestry?
5. Czym charakteryzuje się przerzutnik monostabilny?
6. Czym charakteryzuje się przerzutnik astabilny?
7. Co to są układy wyzwalające?
8. Jakie są podstawowe zadania układów wejściowych w systemach cyfrowych?

4.9.3. Ćwiczenia

Ćwiczenie 1

Zaprojektuj licznik asynchroniczny o pojemności 8 na przerzutnikach JK. Zmontuj zaproponowany układ i zaobserwuj na oscyloskopie przebiegi na jego wyjściach Q_0 , Q_1 , Q_2 .

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinienes:

- 1) przeanalizować treść zadania,
- 2) narysować schemat ideowy licznika asynchronicznego o pojemności 8, zbudowany z przerzutników JK,
- 3) narysować schemat układu do badania licznika,
- 4) zmontować układ według zaproponowanego schematu,
- 5) zaobserwować przebiegi na wyjściach licznika Q_0 , Q_1 , Q_2 , podając na wejście zegarowe przebieg prostokątny o napięciu 5V i częstotliwości 10kHz,
- 6) sformułować wnioski i sporządzić sprawozdanie z ćwiczenia.

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.

Wyposażenie stanowiska pracy:

- dwa układy 74109,
- zasilacz napięcia stałego +5V,
- generator funkcyjny,
- oscyloskop dwukanałowy,
- stanowisko do łączenia układów,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

Ćwiczenie 2

Zaprojektuj rejestr synchroniczny, równoległy czterobitowy na przerzutnikach typu D.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinienes:

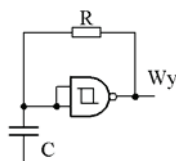
- 1) przeanalizować treść zadania,
- 2) narysować schemat ideowy rejestru synchronicznego, równoległego, czterobitowego na przerzutnikach typu D,
- 3) dobrać z katalogu elementów elektronicznych układy scalone, które można wykorzystać do budowy rejestru,
- 4) uzasadnić przyjęte rozwiązanie,
- 5) zaprezentować wyniki swojej pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

Ćwiczenie 3

Zbuduj generator fali prostokątnej o częstotliwości 1 MHz na bramce Schmitta i zbadaj jego działanie.



Układ generatora fali prostokątnej zbudowany z bramki Schmitta.

Uwaga ! Przed załączeniem napięcia zasilania układ musi sprawdzić nauczyciel.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinienes:

- 1) przeanalizować treść zadania,
- 2) obliczyć wartość kondensatora C, przy wartości rezystora 330Ω dla częstotliwości 1MHz,
- 3) narysować schemat układu do badania generatora fali prostokątnej na bramce Schmitta,
- 4) zmontować układ według zaproponowanego schematu,
- 5) zaobserwować przebiegi na wyjściach generatora,
- 6) sformułować wnioski i sporządzić sprawozdanie z ćwiczenia.

Wyposażenie stanowiska pracy:

- układy UCY 74132,
- zasilacz napięcia stałego +5V,
- rezystor 330Ω ,
- kondensatory 680pF, 4,7nF, 10pF, 3,3nF.
- oscyloskop,
- stanowisko do łączenia układów,
- literatura uzupełniająca zgodna z punktem 6,
- katalogi elementów elektronicznych.

4.9.4. Sprawdzian postępów

Czy potrafisz:

- 1) zaprojektować licznik asynchroniczny o dowolnej pojemności?
- 2) rozpoznać schemat logiczny licznika synchronicznego?
- 3) rozpoznać na podstawie schematu logicznego rodzaj rejestru?
- 4) narysować schemat logiczny dowolnego rejestru?
- 5) zaprojektować rejestr o określonych parametrach?
- 6) opisać budowę generatorów fali prostokątnej?
- 7) opisać działanie układów wyzwalających?
- 8) wyjaśnić zadania układów wejściowych w systemach cyfrowych?
- 9) opisać sposoby transmisji sygnałów cyfrowych?

Tak	Nie
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

4.10. Pamięci. Programowalne struktury logiczne.

4.10.1. Materiał nauczania

Pamięci

Pamięci są układami służącymi do przechowywania informacji w postaci ciągu słów bitowych. Wykonuje się jako układy o bardzo dużym stopniu scalenia w technice TTL, ECL, CMOS i NMOS. Pamięć podzielona jest na komórki (rejstry), każda z nich posiada swój adres umożliwiający dostęp do zawartej w niej informacji. Liczba bitów w komórce określa organizację pamięci, czyli sposób dostępu do informacji. Zazwyczaj komórki pamięci zawierają 8 bitów (choć mogą mieć 4 bity), wówczas mówimy o organizacji słowowej typu $N \cdot 8$ ($N \cdot 4$), gdzie N oznacza ilość komórek. Jeśli pamięć ma komórki jednobitowe mówimy, że jest to pamięć bitowa.

Organizacja pamięci wiąże się z podstawowym parametrem pamięci jakim jest pojemność P , określająca jak wiele informacji można w niej przechowywać. Pojemność pamięci określa się w bitach [b], ze względu na bardzo szybki postęp technologiczny umożliwiający wykonywanie pamięci o dużych pojemnościach, powszechnie używa się pamięci o pojemnościach wyrażanych w kilobitach [Kb] ($1\text{Kb} = 2^{10} \text{ b} = 1024 \text{ b}$) lub megabitach [Mb] ($1\text{Mb} = 1\text{K} \cdot \text{K b} = 2^{20} \text{ b} = 1048576 \text{ b}$).

Przykład

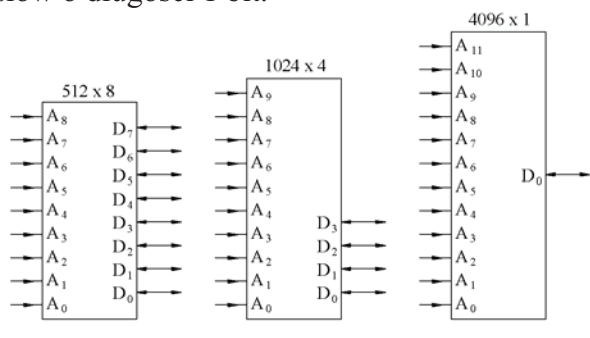
Pamięć o 10 wejściach adresowych i komórkach 8-bitowych, czyli długości słowa 1 bajt (1B) ma pojemność $P = 2^{10} \text{ B} = 1\text{KB}$.

Pamięć tej samej pojemności może być zorganizowana na kilka sposobów zależnie od ilości słów i ich długości.

Przykład

Pamięć o pojemności 4Kb może posiadać organizację:

- $512 \cdot 8$, czyli 512 słów o długości 8 bitów,
- $1024 \cdot 4$, czyli 1024 słowa o długości 4 bity,
- $4096 \cdot 1$, czyli 4096 słów o długości 1 bit.



Rys.43. Przykłady organizacji pamięci 4Kb.

Układy pamięci mogą posiadać następujące wejścia i wyjścia:

- wejścia adresowe umożliwiające dostęp do określonych komórek,
- wejścia sterujące między innymi są to: wejście uaktywniające pamięć CS (CE), wejście zezwalające na zapis WR (WE), wejście zezwalające na odczyt RD (OE), wejście strobowe adresów ALE (RAS lub CAS),
- wejścia /wyjścia danych (informacyjne D).

Pamięci dzielimy na:

- pamięci odczyt – zapis, popularnie zwane RAM,
- pamięci stałe ROM - tylko do odczytu.

Parametry dynamiczne pamięci

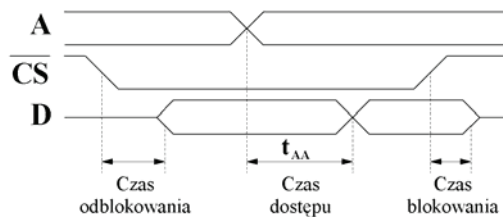
Podstawowe parametry dynamiczne pamięci to czasy: dostępu, cyklu i blokowania.

Czas dostępu t_{AA} jest to czas liczony od wystąpienia nowego adresu do pojawienia się na wyjściach układu zawartości komórki pamięci o tym adresie.

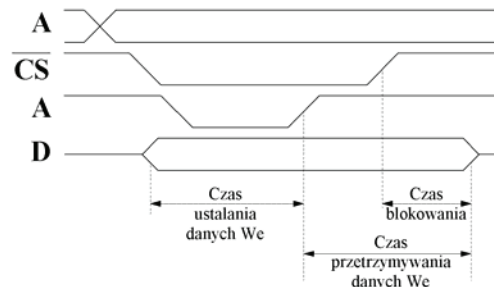
Czas cyklu to minimalny odstęp czasu między kolejnymi prawidłowymi zapisami i/lub odczytami pamięci. Rozróżnia się czasami czas cyklu zapisu, cyklu odczytu i cyklu odczyt-zapis.

Czas blokowania to odstęp czasu liczony od zmiany stanu wejścia wybierającego \overline{CS} do chwili przejścia wyjść/wejść danych do stanu wielkiej impedancji. Czas odblokowania jest liczony od zmiany stanu wejścia wybierającego \overline{CS} do chwili przejścia wejść/ wyjść danych do trybu pracy dwustanowej.

Dla bezkolizyjnej współpracy kilku bloków pamięci trójstanowych o połączonych wyjściach czas blokowania tych układów powinien być krótszy niż czas odblokowania.



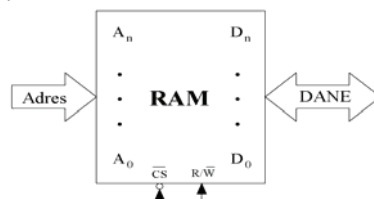
Rys. 44. Przebiegi czasowe sygnałów przy odczycie informacji z pamięci



Rys. 45. Przebiegi czasowe sygnałów przy zapisie informacji do pamięci

Pamięć RAM

Pamięć RAM jest zespołem rejestrów równoległych. Informacja może być zapisywana w dowolnym rejestrze lub odczytywana z dowolnego rejestru, dlatego pamięć ta nazywa się pamięcią o dostępie bezpośrednim.



Rys.46. Symbol graficzny pamięci RAM.[2, s.264]

Pamięć RAM posiada:

- wejścia adresowe $A_0 \dots A_n$,
- wejście uaktywniające \overline{CS} ,
- wejście wyboru trybu pracy R/\overline{W} ,
- wejścia / wyjścia danych $D_0 \dots D_m$.

Pamięć RAM wykonuje się jako układy:

- bipolarne TTL i ECL charakteryzujące się małymi pojemnościami, dużymi prądami zasilającymi i małą szybkością działania (porównywalnie szybkie z CMOS są pamięci ECL)

– unipolarne CMOS i NMOS: statyczne SRAM i dynamiczne DRAM (obecnie DDRAM). Pamięci dynamiczne DRAM wymagają okresowego odświeżania zawartych informacji, mogą one posiadać wewnętrzny układ odświeżania. Obecnie powszechnie stosowane są pamięci DDRAM.

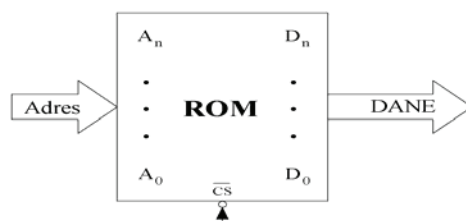
Pamięć ROM

Pamięć ROM to pamięć stała, której zawartość podczas eksploatacji jest niezmienna, raz zapisana informacja jest trwale przechowywana i może być wielokrotnie odczytywana. Pamięć ROM jest cyfrowym układem kombinacyjnym.

Pamięci stałe mogą być programowane:

- przez wytwórcę podczas produkcji - pamięci MROM,
- przez użytkownika za pomocą specjalnych programatorów w sposób trwały, bez możliwości wykasowania zapisanej informacji - pamięci PROM (wykonywane są zazwyczaj jako układy bipolarnie),
- przez użytkownika w sposób prawie trwały, z możliwością wykasowania zapisanej informacji - pamięci EPROM (wykonywane są zazwyczaj jako układy unipolarne).

Pamięci EPROM mogą być zapisywane metodą elektryczną, a kasowane promieniami X lub promieniami ultrafioletowymi, albo zapisywane i kasowane elektrycznie – wtedy noszą nazwę EEPROM.

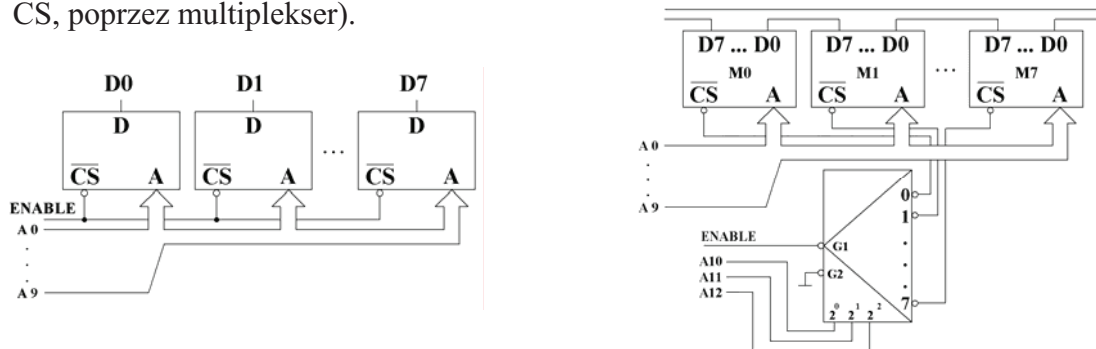


Rys.47. Symbol graficzny pamięci ROM. [2, s.264]

Zwiększanie pojemności pamięci

Układy pamięci można ze sobą łączyć otrzymując w ten sposób bloki pamięci o większej pojemności. Zwiększenie pojemności uzyskuje się:

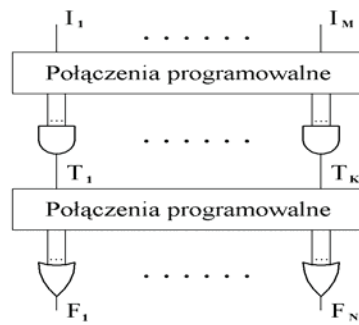
- zwiększając długość słowa,
- zwiększając ilość pamiętanych słów (uaktywniając poszczególne układy pamięci - wejście CS, poprzez multiplexer).



Rys. 48. Powiększenie pojemności pamięci przez a) zwiększenie długości słowa [2, s.272], b) zwiększenie liczby pamiętanych słów [2, s.273]

Programowalne struktury logiczne PLD

Programowalne moduły logiczne PLD są układami o standardowej strukturze, które można dostosować do potrzeb użytkownika poprzez ingerencję w ich standardową strukturę. Układy te wykonywane są w technice TTL i CMOS z tranzystorów, tworzących matrycę bramek AND i matrycę bramek OR. Obie matryce umożliwiają realizację wszystkich funkcji logicznych.



Rys. 49. Struktura układów PLD. [2, s.274]

Moduły PLD mogą być dodatkowo wyposażone w przerzutniki, układy wejściowo-wyjściowe, bufory, sprzężenia zwrotne, wyjścia trójstanowe.

Układy PLD typu matrycowego wykonywane są jako układy PAL, PLA, PLE.

Układy PAL wykonywane są w technice bipolarnej i unipolarnej jako:

- układy EPLD – kasowalne promieniami ultrafioletowymi (wykonane w technice CMOS),
- układy GAL – kasowalne elektrycznie (wykonane w technice EECMOS), charakteryzują się niskimi kosztami wykonania, niewielkim poborem prądu, dużą szybkością działania.

Układy PLA prócz matryc bramek AND i OR posiadają programowalny układ polaryzacji wyjść i sterowany z zewnątrz trójstanowy bufor wyjściowy. Dodatkowo mogą posiadać przerzutniki zwane są wówczas układami PLS lub sekwenserami.

Układy PLE są nową generacją szybkich pamięci PROM, realizującą układy logiczne. Mogą być dodatkowo wyposażone w buforowe rejestry wyjściowe.

4.10.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

1. Jak dzielimy pamięci półprzewodnikowe?
2. Czym różni się pamięć SRAM od pamięci DRAM?
3. Co oznacza organizacja pamięci?
4. Jaki jest związek między pojemnością a organizacja pamięci?
5. Jak dzielimy pamięci ROM ze względu na sposób programowania?
6. Co określają parametry dynamiczne pamięci?
7. W jaki sposób można zwiększyć pojemność pamięci?
8. Do czego służą układy PLD?

4.10.3. Ćwiczenia

Ćwiczenie 1

Narysuj symbole graficzne pamięci ROM o pojemności 8Kb i organizacji: 1024 · 8, 2048 · 4, 8192 · 1.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinienes:

- 1) przeanalizować treść zadania,
- 2) narysować symbole graficzne pamięci ROM o określonych pojemnościach,
- 3) zaprezentować wyniki swojej pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 2

Zaprojektuj pamięci RAM o pojemności 8Kb i organizacji 1024·8, korzystając z układów pamięci 1 Kb i organizacji 1024 ·1.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) narysować schemat bloku pamięci,
- 3) zaprezentować wyniki swojej pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

Ćwiczenie 3

Zaprojektuj pamięci ROM o pojemności 8KB i organizacji 8192·8, korzystając z układów pamięci 8 Kb i organizacji 1024 ·8 .

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) przeanalizować treść zadania,
- 2) narysować schemat bloku pamięci,
- 3) zaprezentować wyniki swojej pracy.

Wyposażenie stanowiska pracy:

- zeszyt,
- literatura uzupełniająca zgodna z punktem 6.

4.10.4. Sprawdzian postępów

Czy potrafisz:

	Tak	Nie
1) wyjaśnić pojęcie organizacji pamięci?	<input type="checkbox"/>	<input type="checkbox"/>
2) wyjaśnić, co to znaczy, że pamięć ma dostęp bezpośredni?	<input type="checkbox"/>	<input type="checkbox"/>
3) wyjaśnić różnicę między pamięcią RAM i ROM?	<input type="checkbox"/>	<input type="checkbox"/>
4) opisać podstawowe parametry dynamiczne pamięci?	<input type="checkbox"/>	<input type="checkbox"/>
5) opisać sposoby programowania pamięci ROM?	<input type="checkbox"/>	<input type="checkbox"/>
6) projektować bloki pamięci o dużej pojemności wykorzystując mniejsze układy pamięci?	<input type="checkbox"/>	<input type="checkbox"/>
7) opisać budowę programowalnych struktur PLD?	<input type="checkbox"/>	<input type="checkbox"/>
8) sklasyfikować programowalne struktury PLD?	<input type="checkbox"/>	<input type="checkbox"/>

6. LITERATURA

1. Chwaleba A., Moeschke B., Płoszajski G.: Elektronika. WSiP, Warszawa 1999
2. Głocki W.: Układy cyfrowe. WSiP, Warszawa 2000
3. Płoszajski G.: Automatyka. WSiP, Warszawa 1995,
4. Praktyczna elektronika. REA, Warszawa 2003
5. Piecha J.: Elementy i układy cyfrowe PWN Warszawa 1990.