

ĆWICZENIE nr 1

PODSTAWOWE FUNKCJE LOGICZNE

Dokument służy wyłącznie do celów indywidualnego kształcenia.
Zabrania się utrwalania, przekazywania osobom trzecim oraz rozpowszechniania.

1.1 Cel ćwiczenia:

Celem ćwiczenia jest zapoznanie się z podstawowymi bramkami logicznymi. W ćwiczeniu należy wyznaczyć tablice przejść wszystkich badanych bramek logicznych.

Druga część ćwiczenia polega na złożeniu z dostępnych bramek prostego układu logicznego i wyznaczenie dla niego tablicy przejść.

1.2 Wprowadzenie teoretyczne:

1.2.1 Poziomy logiczne.

Wszystkie układy standardu TTL zasilane są napięciem o wartości +5V z tolerancją $\pm 0,25$ V. Przekroczenie podanego zakresu może spowodować uszkodzenie układu - za duża wartość napięcia, lub jego błędne działanie - za niska wartość napięcia.

Sygnały w technice cyfrowej przybierają jedną z dwóch dozwolonych wartości napięcia: 0 V (logiczne zero) lub +5 V (logiczna jedynka). Niewielkie odchylenia napięć nie powodują błędów. W praktyce określa się dwa przedziały, w których mogą znajdować się wartości napięć odpowiadające poziomom logicznym 0 i 1. Dla układów scalonych serii UCY74 przedziały te są następujące:

- wartość logiczna 0 - napięcia z zakresu $-0,5$ V ÷ $+0,8$ V,
- wartość logiczna 1 - napięcia z zakresu $+2$ V ÷ $+5,5$ V.

Doprowadzenie do wejść układów napięć innych niż podane wyżej powoduje błędne działanie lub uszkodzenie układu. Praktycznie w układach pojawiają się napięcia w zakresach:

- wartość logiczna 0 - napięcia z zakresu 0 V ÷ $+0,4$ V,
- wartość logiczna 1 - napięcia z zakresu $+2,4$ V ÷ $+5$ V.

Dzięki temu uzyskuje się większą odporność układów na zakłócenia i szумы.

1.2.2 Rodzaje bramek. Parametry elektryczne.

Głównym przeznaczeniem bramek logicznych jest realizacja układów obliczających funkcje logiczne. Do podstawowych bramek logicznych należą trzy bramki AND, OR i NOT. Za pomocą tych trzech bramek można zbudować pozostałe bramki pochodne oraz dowolny układ logiczny. Mimo tego produkuje się znacznie więcej rodzajów bramek. Różnią się one między sobą liczbą wejść, realizowaną funkcją i parametrami elektrycznymi.

Jednym z parametrów elektrycznych bramek jest obciążalność. Parametr ten mówi nam o tym ile wejść może być wysterowanych przez jedno wyjście. Liczba ta wynika z obciążalności prądowej wyjścia i prądów wejściowych. Typowa obciążalność jest równa 10.

Innym parametrem bramek jest czas propagacji bramki określający szybkość działania bramki. Typowy czas opóźnienia zbocza opadającego (przejście z 1 na 0) wynosi 7 ns, zaś zbocza narastającego (przejście z 0 na 1) - 11 ns. Wpływ szybkości narastania i opadania zboczy sygnału sterującego na pracę bramki występuje dlatego, że przez pewien czas napięcie na wejściu ma nieokreślony poziom pośredni między 0 i 1. W tym czasie na wyjściu pojawi się również poziom nieokreślony, a nawet mogą

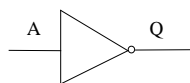
wystąpić oscylacje. Dlatego też zaleca się, aby czasy narastania i opadania sygnałów sterujących wejścia trwały krócej niż 1 μ s.

Zasady łączenia wejść i wyjść:

- wejścia układów można łączyć bezpośrednio z wyjściami innych, przy czym do jednego wyjścia można przyłączyć nie więcej jak 10 wejść,
- wejścia układów można zwierać do masy i do +5V,
- wejścia układów można łączyć ze sobą,
- nie wolno łączyć wyjść układów z +5V i masą,
- nie wolno łączyć wyjść układów ze sobą, chyba, że wyjścia są typu otwarty kolektor lub trójstanowe.
- wolne wejścia należy łączyć z masą lub +5V, tak aby nie zakłóciło to pracy układu (nie wolno pozostawiać ich „w powietrzu” ze względu na wrażliwość na zakłócenia).

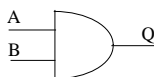
1.2.3 Opisy poszczególnych bramek logicznych.

Inwerter - bramka ta odwraca sygnał podany na jej wejście. Symbol inwertera i tablicę przejść pokazano poniżej. Jak można zauważyć, poziomy napięcie na wyjściu i na wejściu są zawsze odwrotne.

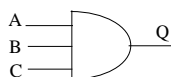


| A | Q |
|---|---|
| 1 | 0 |
| 0 | 1 |

AND - jest to bramka, w której na jej wyjściu pojawia się logiczna 1 tylko wtedy, gdy na wszystkich jej wejściach występują poziomy logiczne 1. Bramki wejściowe AND mogą mieć dwa, trzy lub więcej wejść, zależnie od tego ile zmiennych wejściowych ma być ze sobą skojarzonych przez tzw. iloczyn logiczny. Poniżej przedstawiono symbole i tabele przejść dla dwuwejściowej i trójwejściowej bramki AND.

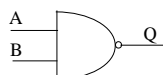


| A | B | Q |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |



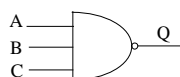
| A | B | C | Q |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

NAND - bramka o funkcji odwrotnej niż bramka AND. Bramkę tą można uważać za szeregowo połączenie bramki AND i Inwertera. Logiczna jedynka pojawia się na wyjściu zawsze wtedy, gdy na którymkolwiek z wejść występuje logiczne zero. Natomiast logiczne zero pojawi się na wyjściu tylko wtedy, gdy na wszystkich wejściach panuje logiczna jedynka. Poniżej przedstawiono symbole i tabele przejść dla dwuwejściowej i trójwejściowej bramki NAND.



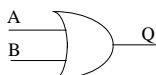
| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

| A | B | C | Q |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 |

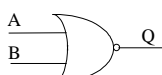


OR - jest to bramka sumy logicznej. Na jej wyjściu jedynka pojawia się wtedy, gdy przynajmniej na jednym z wejść występuje logiczna jedynka. Zero na wyjściu pojawi się tylko w przypadku, gdy na wszystkich wejściach występuje zero. Symbol bramki i tablicę przejść pokazano poniżej.

| A | B | Q |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

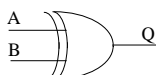


NOR - stanowi ona połączenie bramki NOT z bramką OR. Na wyjściu tej bramki logiczna jedynka pojawi się tylko wówczas, gdy na wszystkich wejściach będą występować logiczne zera. W każdym innym przypadku na wyjściu tej bramki będzie występować logiczne zero. Symbol bramki i tablicę przejść pokazano poniżej.



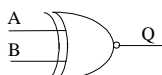
| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

EX-OR - Exclusive-OR. Bramka ta wykazuje nierówność stanów logicznych podanych na jej wejścia. Gdy na wejściach tej bramki panują różne stany logiczne (0 i 1, 1 i 0) to na jej wyjściu występuje logiczna jedynka.



| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

EX-NOR - Exclusive-NOR. Bramka ta wykazuje równość stanów logicznych podanych na jej wejścia. Gdy na wejściach tej bramki panują jednakowe stany logiczne (0 i 0, 1 i 1) to na jej wyjściu występuje logiczna jedynka.



| A | B | Q |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

1.3 Podstawowe prawa algebry Boole'a

Spośród wielu praw algebry Boole'a podstawowe znaczenie w zastosowaniu do teorii układów cyfrowych mają następujące cztery prawa:

- przemienności
- łączności
- rozdzielczości
- De Morgana

Prawa te i odpowiadające im wyrażenia zestawiono w poniższej tabelicy.

| | iloczyn logiczny | suma logiczna |
|-----------------------|--|---|
| prawo przemienności | $A*B = B*A$ | $A+B = B+A$ |
| prawo łączności | $A*(B*C) = (A*B)*C$ | $A+(B+C) = (A+B)+C$ |
| prawo rozdzielczości | $A*(B+C) = A*B+A*C$ | $A+B*C = (A+B)*(A+C)$ |
| prawo De Morgana | $\overline{A*B*...} = \overline{A} + \overline{B} + ...$ | $\overline{A+B+...} = \overline{A} * \overline{B} * ...$ |
| Tożsamości podstawowe | $A*0 = 0$ $A*1 = A$ $A*A = A$ $A*\overline{A} = 0$ | $A+1 = 1$ $A+0 = A$ $A+A = A$ $A+\overline{A} = 1$ |
| Tożsamości dodatkowe | $A*(A+B) = A$ $A+\overline{A}*B = A+B$ $(A+B)*(\overline{A}+\overline{B}) = B$ | $A+A*B = A$ $A*(\overline{A}+B) = A*B$ $A*B+\overline{A}*B = B$ |

Prawo przemienności i prawo łączności, a także prawo rozdzielczości mnożenia względem dodawania są takie same jak w zwykłej algebrze. Natomiast prawo rozdzielczości dodawania względem mnożenia i prawo De Morgana są specyficznymi prawami dwuelementowej algebry Boole'a.

Porównując wzory z pierwszej i drugiej kolumny powyższej tabelicy można zauważyć charakterystyczną dwoistość polegającą na tym, że każdemu prawu odnoszemu się do działania dodawania odpowiada analogiczne prawo odnoszące się do działania mnożenia. Z powyższych zależności korzysta się przy przekształcaniu wyrażeń opisujących złożone funkcje o wielu zmiennych w celu otrzymania ich możliwie najprostszej postaci końcowej, a co za tym idzie, prostszej realizacji układowej. Proces ten jest określany jako **minimalizacja** funkcji logicznej.

1.4 Proces minimalizacji funkcji logicznej

Minimalizacja funkcji logicznej polega na takim przekształceniu postaci kanonicznej funkcji logicznej, zgodnie z zasadami algebry Boole'a, aby uzyskać możliwie najprostszy jej zapis. Im bardziej złożona jest funkcja logiczna, tym bardziej rozbudowany jest system cyfrowy potrzebny do realizacji tej funkcji. Zatem każde uproszczenie wyrażenia logicznego umożliwia łatwiejszą realizację układową funkcji przy użyciu mniejszej liczby elementarnych bramek logicznych. Metody minimalizacji funkcji logicznych można podzielić ogólnie na **algebraiczne i graficzne**.

Stosowanie metod algebraicznych z wykorzystaniem praw i tożsamości algebry Boole'a ilustrują następujące, proste przykłady:

$$1. \quad F = \overline{A}BC + A\overline{B}\overline{C} = \overline{A}B(C + \overline{C}) = \overline{A}B$$

$$2. \quad F = AB + B\bar{C} + \bar{A}\bar{C} = AB + B\bar{C}(A + \bar{A}) + \bar{A}\bar{C} = AB + AB\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{C} = \\ = AB(1 + \bar{C}) + \bar{A}\bar{C}(1 + B) = AB + \bar{A}\bar{C}$$

Pierwszy przykład jest bardzo prosty. Natomiast w drugim przypadku dostrzeżenie, że $AB + B\bar{C} + \bar{A}\bar{C} = AB + \bar{A}\bar{C}$ nie jest takie łatwe. W przypadku złożonych funkcji wielu zmiennych metoda kolejnych przekształceń algebraicznych wyrażeń logicznych przy bezpośrednim wykorzystaniu praw algebry Boole'a staje się bardzo uciążliwa i nie zawsze w praktyce prowadzi do osiągnięcia zamierzonego celu. Prostota końcowej postaci otrzymanych funkcji zależy w dużej mierze od intuicji i umiejętności projektanta, dlatego też jest stosowana rzadko i tylko dla prostych funkcji. Efektywniejszą metodą minimalizacji jest jedna z metod graficznych - **metoda Karnaugh'a**.

Tablica (mapa) Karnaugh'a jest uporządkowaną w specyficzny sposób postacią zapisu tablicy wartości funkcji logicznej. Korzysta się z niej w procesie minimalizacji funkcji logicznych. Tablica ta ma strukturę prostokątną, złożoną z elementarnych pól. Każde pole reprezentuje iloczyn pełny w odniesieniu do zmiennych wejściowych, czyli zmiennych niezależnych danej funkcji. Zatem tablica ta obejmuje wszystkie możliwe kombinacje wartości argumentów. Na marginesach tablicy wpisuje się w określonym porządku (wg kodu Gray'a) wartości argumentów. Przy parzystej liczbie argumentów połowa z nich umieszczona jest na marginesie poziomym, a druga połowa - na marginesie pionowym.

Przy nieparzystej liczbie argumentów wpisuje się na jednym marginesie o jeden argument więcej niż na drugim. Ułożenie tablicy Karnaugh'a polega na takim zgrupowaniu wszystkich kombinacji wartości argumentów, aby zawsze przy przejściu z danego pola do pola sąsiedniego zmieniała się wartość tylko jednego argumentu. Zasada sąsiedztwa obowiązuje również dla pól leżących przy krawędzi tablicy.

Poniżej przedstawione są tablice dla funkcji dwóch, trzech i czterech zmiennych wejściowych. Wartości argumentów zanegowanych są opisane cyfrą 0, a niezanegowanych - cyfrą 1.

Tabela Karnaugh'a funkcji dwóch zmiennych

| | | |
|----------|------------------|------------|
| B | 0 | 1 |
| A | | |
| 0 | $\bar{A}\bar{B}$ | $\bar{A}B$ |
| 1 | $A\bar{B}$ | AB |

Tabela Karnaugh'a funkcji trzech zmiennych

| | | |
|-----------|-------------------------|-------------------|
| C | 0 | 1 |
| AB | | |
| 00 | $\bar{A}\bar{B}\bar{C}$ | $\bar{A}\bar{B}C$ |
| 01 | $\bar{A}B\bar{C}$ | $\bar{A}BC$ |
| 11 | $AB\bar{C}$ | ABC |
| 10 | $A\bar{B}\bar{C}$ | $A\bar{B}C$ |

Tabela Karnaugh'a funkcji czterech zmiennych

| CD AB | 00 | 01 | 11 | 10 |
|----------|--|---|------------------------------|---|
| 00 | $\overline{A}\overline{B}\overline{C}\overline{D}$ | $\overline{A}\overline{B}\overline{C}D$ | $\overline{A}\overline{B}CD$ | $\overline{A}B\overline{C}\overline{D}$ |
| 01 | $\overline{A}\overline{B}C\overline{D}$ | $\overline{A}\overline{B}CD$ | $\overline{A}BCD$ | $\overline{A}B\overline{C}\overline{D}$ |
| 11 | $A\overline{B}\overline{C}\overline{D}$ | $A\overline{B}\overline{C}D$ | $A\overline{B}CD$ | $A\overline{B}\overline{C}\overline{D}$ |
| 10 | $A\overline{B}C\overline{D}$ | $A\overline{B}CD$ | $AB\overline{C}\overline{D}$ | $AB\overline{C}D$ |

Następny rysunek ilustruje prosty przykład stosowania tablicy Karnaugh'a do minimalizacji funkcji opisanej wyrażeniem:

$$F = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D}$$

Funkcję logiczną będącą sumą iloczynów jej argumentów (z negacją lub bez) oznacza się przypisując cyfrę 1 każdemu polu, w którym występuje składnik analizowanej funkcji. Pola nieopisane pozostawia się puste lub oznacza cyfrą 0.

Przykład zastosowania tablicy Karnaugh'a do funkcji czterech zmiennych

| CD AB | 00 | 01 | 11 | 10 |
|----------|----|----|----|----|
| 00 | | 1 | 1 | |
| 01 | | 1 | 1 | |
| 11 | | | | |
| 10 | 1 | | | 1 |

Minimalizacja funkcji logicznej polega na łączeniu sąsiednich pól oznaczonych cyfrą 1 w odpowiednie grupy złożone z dwóch, czterech, ośmiu itd. pól, które wyróżnia się obwiednią. Należy przy tym pamiętać, że pola na brzegach tablicy również sąsiadują ze sobą. Istnienie sąsiadujących pól oznaczonych 1 wskazuje możliwość wyeliminowania niektórych zmiennych. Na przykład zmienna C może zostać wyeliminowana w grupie:

$$\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} = \overline{A}\overline{B}D(C + \overline{C}) = \overline{A}\overline{B}D$$

Postępując w podobny sposób ze składnikami grupy czteropolowej

$$\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD + \overline{A}\overline{B}\overline{C}\overline{D} = \overline{A}\overline{B}D$$

podane wyrażenie funkcyjne można ostatecznie sprowadzić do prostej postaci:

$$F = \overline{A}\overline{B}D + \overline{A}D$$

W niektórych przypadkach proces minimalizacji funkcji przebiega łatwiej, gdy grupuje się zera, czyli określa funkcję będącą dopełnieniem wyrażenia reprezentowanego przez jedynki. Gdy liczba zmiennych przewyższa pięć, metoda Karnaugh'a staje się uciążliwa i wówczas niekiedy dogodniej stosować inne metody minimalizacyjne, np. Quine'a-Mc Cluskey'a, lub o wiele wydajniejsze metody numeryczne poszukiwania rozwiązań minimalnych za pomocą komputera.

1.5 Pytania sprawdzające:

- 1) Podać wartości poziomów logicznych stosowanych w technice cyfrowej. Podać przedziały w jakich zawierają się poziomy logiczne 1 i 0.
- 2) Wyjaśnić pojęcie obciążalności wyjścia bramki.
- 3) Podać podstawowe zasady łączenia wejść i wyjść bramek.
- 4) Wymienić poznane bramki i podać ich tablice przejść.

- 5) Podać podstawowe prawa logiki stosowane przy projektowaniu układów kombinacyjnych.
- 6) Podać wzory De Morgan'a.
- 7) Czym jest proces minimalizacji funkcji logicznej? Podać cel i sposoby.
- 8) Do czego służą siatki Karnaugh'a? Omówić sposób ich wykorzystywania przy minimalizacji funkcji na konkretnym przykładzie podanym przez prowadzącego.

1.6 Przebieg ćwiczenia:

W trakcie ćwiczenia należy:

- 1) zbadać wybrane bramki logiczne podając na ich wejścia wszystkie możliwe kombinacje stanów logicznych, obserwując jednocześnie stany wyjść. Wyniki należy wpisać do podanych poniżej tabel. Następnie należy porównać otrzymane tablice przejść poszczególnych bramek z podanymi w części teoretycznej.

inwerter NOT

| A | Q |
|---|---|
| 1 | |
| 0 | |

bramki dwuwejściowe AND, NAND, NOR, OR, XNOR.

| A | B | Q |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

bramka trójwejściowa NAND

| A | B | C | Q |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

- 2) Przeprowadzić symulację, układu kombinacyjnego realizującego funkcję logiczną podaną przez prowadzącego.

W czasie wykonywania ćwiczenia należy, podając na wejścia układu wszystkie możliwe kombinacje stanów logicznych, zbadać odpowiadające im stany wyjść wpisując wyniki do tabeli. W kolejnym etapie dokonać minimalizacji funkcji oraz symulacji uzyskanego układu zapisując wyniki w tabeli. Tabele będą służyć do porównania funkcji logicznych podanej w postaci nie zminimalizowanej z postacią zminimalizowaną wyprowadzoną przez ćwiczącego.

- 3) Zaproponować układ realizujący funkcję NAND, NOR, EXOR złożony z bramek podstawowych AND, OR i NOT.