

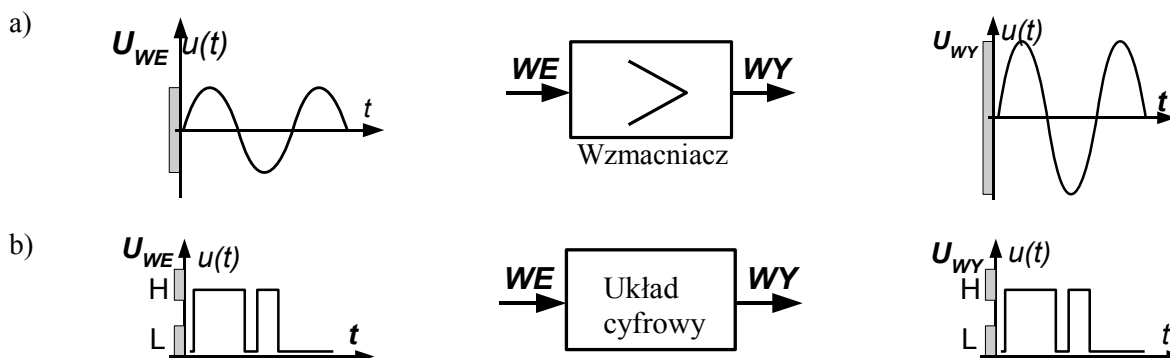
Ćwiczenie nr 14. Podstawowe funktry logiczne

Cel ćwiczenia:

Zapoznanie studentów z elementarnymi funkcjami logicznymi i układami realizującymi te funkcje, powszechnie stosowanymi w technice cyfrowej.

Wprowadzenie

Analogowe układy elektroniczne przeznaczone są do przetwarzania sygnałów analogowych, które mogą przyjmować dowolną wartość z pewnego określonego przedziału np. napięcia, natężenia prądu [3].



Rys.1. Porównanie układów: a) analogowego i b) cyfrowego

Natomiast układy cyfrowe stosowane są do odczytu, przetwarzania i zapisu informacji zakodowanej w postaci dwóch przedziałów wartości napięć (lub natężeń prądów): wysokiego **H** (High) i niskiego **L** (Low). Taki dwuwartościowy sygnał nazywany jest binarnym (dwójkowym).

Sygnał dwójkowy w określonej chwili może przyjmować jedną z dwóch wartości napięcia oznaczonych umownie symbolami **0** (zera) i **1** (jedyńki) logicznej.

Sygnały cyfrowe charakteryzują się dużą odpornością na zakłócenia i zniekształcenia. Przy sygnałach napięciowych, w których wyższemu poziomowi napięcia **H** (bardziej dodatni) przyporządkowuje się jedyńkę logiczną, a niższemu **L** (mniej dodatni) – zero logiczne, mówi się o logice dodatniej. Konwencja przeciwna nazywana jest logiką ujemną. Poziomy napięć odpowiadające stanom zera i jedyńki są związane z technologią realizacji układów logicznych. Najszerzej znane to układy realizowane w technologii TTL (Transistor Transistor Logic) i CMOS (Complementary Metal Oxide Semiconductor) oraz ich modyfikacje związane z rozwojem technologii.

Sygnał cyfrowy np. dwójkowy, służy do przedstawienia wartości wielkości nieciągłych (ziarnistych). Wartości wielkości ziarnistej można zapisać za pomocą kombinacji cyfr **1** i **0**, czyli w postaci kodu.

Najmniejszą jednostką informacji sygnału cyfrowego jest bit, który oznacza jeden z dwóch możliwych stanów: 0 lub 1 – prawda lub fałsz. Grupa bitów tworzy słowo, a liczba bitów słowa określa jego długość. Słowo złożone z 8 bitów stanowi 1 bajt – można w nim zapisać 1 z 256 różnych stanów. Pierwszy bit z lewej strony słowa cyfrowego nazywa się bitem najbardziej znaczącym – MSB (Most Significant Bit), a pierwszy bit z prawej strony nazywany jest bitem najmniej znaczącym – LSB (Least Significant Bit) [1]. Najczęściej stosowane kody to: naturalny kod dwójkowy, kod Greya, kod dwójkowo-dziesiętny – BCD (Binary Coded Decimal). Kod BCD jest podzbiorem naturalnego 4-bitowego kodu dwójkowego, który określa liczbę w systemie dziesiętnym na podstawie zależności:

$$n = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

Współczynniki b_3 , b_2 , b_1 , b_0 mogą przyjmować wartości tylko **1** lub **0**. Wartości tych współczynników odpowiadające liczbom dziesiętnym w zakresie od zera do piętnastu przedstawiono w tab. 1a.

Strukturę kodu Greya (4-bitowego) przedstawiono w tab. 1c. Kod Greya charakteryzuje się tym, że tylko jeden bit zmienia wartość przy kolejnym zliczeniu. Kod Greya, w przeciwieństwie do naturalnego kodu dwójkowego, nie jest kodem pozycyjnym.

Układy elektroniczne służące do przetwarzania sygnałów cyfrowych noszą nazwę przetworników cyfrowo-cyfrowych lub układów logicznych.

Ćwiczenie nr 14. Podstawowe funkcje logiczne

Tabela 1. Wartości współczynników kodu naturalnego, BCD i kodu Greya

a) kod naturalny (8421)					b) kod BCD					c) kod Greya				
<i>n</i>	<i>b₃</i>	<i>b₂</i>	<i>b₁</i>	<i>b₀</i>	<i>n</i>	<i>b₃</i>	<i>b₂</i>	<i>b₁</i>	<i>b₀</i>	<i>n</i>	<i>b₃</i>	<i>b₂</i>	<i>b₁</i>	<i>b₀</i>
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	1	0	0	0	1
2	0	0	1	0	2	0	0	1	0	2	0	0	1	1
3	0	0	1	1	3	0	0	1	1	3	0	0	1	0
4	0	1	0	0	4	0	1	0	0	4	0	1	1	0
5	0	1	0	1	5	0	1	0	1	5	0	1	1	1
6	0	1	1	0	6	0	1	1	0	6	0	1	0	1
7	0	1	1	1	7	0	1	1	1	7	0	1	0	0
8	1	0	0	0	8	1	0	0	0	8	1	1	0	0
9	1	0	0	1	9	1	0	0	1	9	1	1	0	1
10	1	0	1	0						10	1	1	1	1
11	1	0	1	1						11	1	1	1	0
12	1	1	0	0						12	1	0	1	0
13	1	1	0	1						13	1	0	1	1
14	1	1	1	0						14	1	0	0	1
15	1	1	1	1						15	1	0	0	0

1. Układy logiczne

Układy logiczne dzieli się na układy kombinacyjne i sekwencyjne. Układy logiczne kombinacyjne to takie, w których stan wyjść zależy tylko od aktualnego stanu na wejściach. W układach logicznych sekwencyjnych sygnały wyjściowe zależą nie tylko od stanów na wejściach, lecz także od stanów wewnętrznych układu tzn. od jego „historii”. Najprostszymi układami logicznymi kombinacyjnymi są bramki logiczne – funkcje. Podstawowymi układami sekwencyjnymi są przerzutniki.

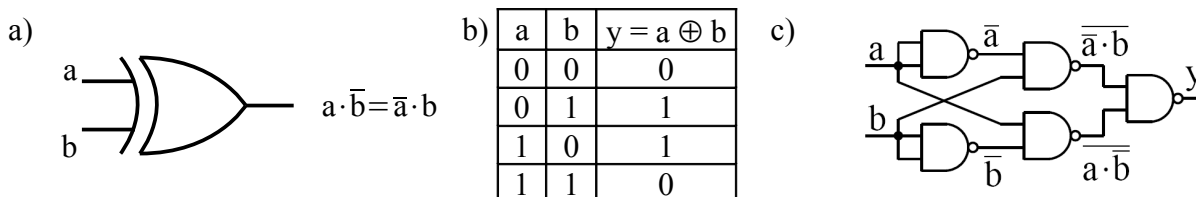
Przy opisie układów logicznych korzysta się z algebry logiki, zwanej też algebrą Boole'a. W algebrze logiki dowolne zmienne mogą osiągać tylko dwa stany: „tak” lub „nie” przybierające wartości: „1” i „0”. Funkcję jednej lub wielu zmiennych, które są zmiennymi binarnymi nazywa się funkcją przełączającą.

Trzy podstawowe funkcje przełączające używane w algebrze to: przeczenie, suma logiczna i iloczyn logiczny, zwane również: negacją, alternatywą i koniunkcją (ang. NOT, OR, AND), a ich właściwości definiują tablice stanów (prawdy) - tab. 2.

Tabela 2. Tablice stanów funkcji: a) AND, b) OR, c) NOT, d) NOR, e) NAND

a) Negacja NOT	b) Suma logiczna OR	c) Iloczyn logiczny AND	d) Zanegowana suma NOR	e) Zanegowany iloczyn NAND																																																																		
<table border="1"> <tr><td>a</td><td>$v = \bar{a}$</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	a	$v = \bar{a}$	0	1	1	0	<table border="1"> <tr><td>a</td><td>b</td><td>$a \vee b$</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	$a \vee b$	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td>a</td><td>b</td><td>$a \wedge b$</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	$a \wedge b$	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><td>a</td><td>b</td><td>$\overline{a \vee b}$</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	$\overline{a \vee b}$	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <tr><td>a</td><td>b</td><td>$\overline{a \wedge b}$</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	$\overline{a \wedge b}$	0	0	1	0	1	1	1	0	1	1	1	0
a	$v = \bar{a}$																																																																					
0	1																																																																					
1	0																																																																					
a	b	$a \vee b$																																																																				
0	0	0																																																																				
0	1	1																																																																				
1	0	1																																																																				
1	1	1																																																																				
a	b	$a \wedge b$																																																																				
0	0	0																																																																				
0	1	0																																																																				
1	0	0																																																																				
1	1	1																																																																				
a	b	$\overline{a \vee b}$																																																																				
0	0	1																																																																				
0	1	0																																																																				
1	0	0																																																																				
1	1	0																																																																				
a	b	$\overline{a \wedge b}$																																																																				
0	0	1																																																																				
0	1	1																																																																				
1	0	1																																																																				
1	1	0																																																																				

Elementy fizyczne realizujące podstawowe funkcje przełączające nazywa się elementami kombinacyjnymi, funktoami lub bramkami logicznymi – tab. 3 [1]. Powszechnie stosuje się też bramkę Exclusive OR, zwaną również symetryczną. Jej symbol, tablice stanów i schemat przedstawiono na rys.2.







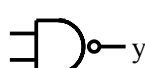
Rys.2. Bramka EXOR: a) symbol graficzny, b) tablica stanów, c) realizacja bramki EXOR za pomocą bramek NAND

Dowolną funkcję logiczną można przedstawić jako kombinację zdefiniowanych w tab. 3 działań i zrealizować przy użyciu ww. funktoów. Podstawowe działania logiczne przedstawiono w tab. 4.

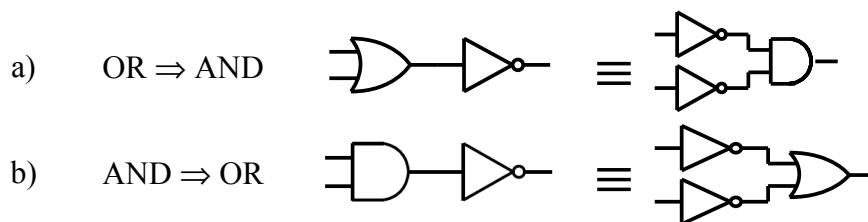
Ćwiczenie nr 14. Podstawowe funktry logiczne

Za pomocą tzw. przekształceń równoważnych można dokonać zamiany elementów NOR w NAND i odwrotnie. Realizacje równoważności zachodzących między funkcjami NOR, NAND, AND i OR są opisane prawami de Morgana - tab.4.

Tabela 3. Oznaczenia kombinacyjnych układów logicznych

Układ	Symbol graficzny	Funkcja
inwerter NOT (NIE)		negacja $y = \bar{a}$
bramka sumacyjna OR (LUB)		suma logiczna $a \vee b \equiv a + b$
bramka iloczynowa AND (I)		iloczyn logiczny $a \wedge b \equiv a \cdot b$
NOR (NIE-LUB)		zanegowana suma $\overline{a \vee b} \equiv \bar{a} + \bar{b}$
NAND (NIE-I)		zanegowany iloczyn $\overline{a \wedge b} \equiv \bar{a} \cdot \bar{b}$

Przykładowe równoważności między układami zawierającymi elementy OR i AND pokazano na rys.3.



Rys.3. Równoważności między układami zawierającymi elementy OR i AND na przykładzie realizacji bramek: a) NOR, b) NOT

Prawa de Morgana mają istotne znaczenie przy projektowaniu układów logicznych. Ponieważ wszystkie wartości liczbowe można wyrazić w postaci kombinacji dwóch stanów logicznych, a operacje algebraiczne w postaci operacji na tych stanach, więc algebra logiki jest bardzo ważna w elektronice. Jednymi z najbardziej rozpowszechnionych układów logicznych są układy TTL, pracujące w logice dodatniej. Dążąc do poprawy parametrów układów TTL, wprowadzono serie z tranzystorami Schottky'ego: zwykła (74S) oraz szybka małej mocy (74LS). Ich wspólne cechy to: napięcie zasilania +5 V ($\pm 5\%$) oraz poziomy wyjściowych sygnałów logicznych: wysoki 3,5 V (ale nie mniej niż 2,4 V), niski 0,2 V (ale nie więcej niż 0,4 V), obciążalność 10 – do jednego wyjścia można dołączyć 10 standardowych wejść.

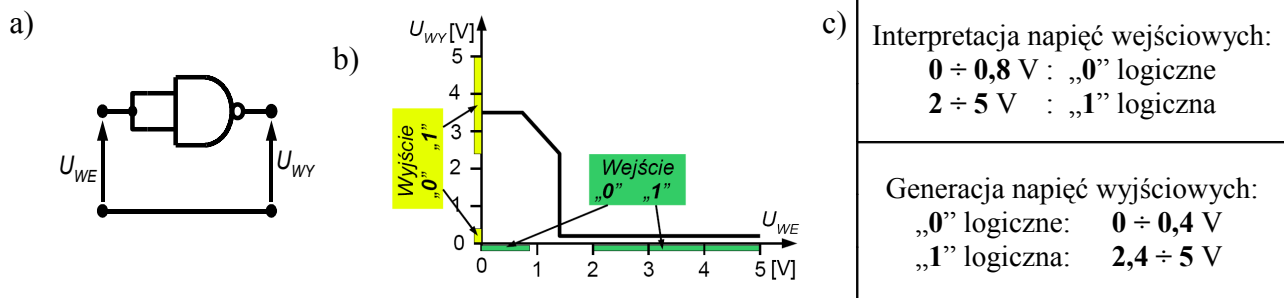
Typową charakterystykę przejściową tych układów przedstawiono na rys. 4. Wymagany zakres interpretacji napięć wejściowych to: dla 0 $\leq 0,8$ V i dla 1 logicznej ≥ 2 V. Zwiększone w stosunku do wyjścia przedziały wejściowe o 0,4 V stanowią margines bezpieczeństwa, z uwagi na możliwość wystąpienia zakłóceń. Uproszczoną charakterystykę przejściową inwertera, typową dla układów TTL, przedstawiono na rys.4b wraz z przedziałami napięć wejściowych i wyjściowych.

Układy logiczne wykorzystuje się w elektronice m.in. w prostych urządzeniach automatyki analizują stany czujników, jak również do komputerów sterujących eksperymentem.

Tabela 4. Podstawowe działania logiczne i ich właściwości

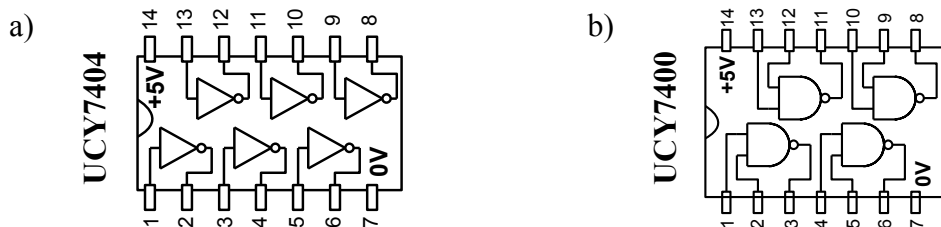
Lp.	Działanie	Lp.	Działanie	Lp.	Prawa de Morgana
1	$a + 0 = a$	5	$a + b \cdot a = a + b$	9	$\overline{a \cdot b} = \bar{a} + \bar{b}$
2	$a \cdot 1 = a$	6	$a \cdot b \cdot c = a \cdot (b \cdot c) = (a \cdot b) \cdot c$	10	$\overline{a + b} = \bar{a} \cdot \bar{b}$
3	$a \cdot b = b \cdot a$	7	$a \cdot (b + c) = a \cdot b + a \cdot c$	11	$\overline{a \cdot b} = \bar{a} + \bar{b}$
4	$a + b = b + a$	8	$a + b + c = a + (b + c) = (a + b) + c$	12	$a + b = \overline{\bar{a} \cdot \bar{b}}$

Ćwiczenie nr 14. Podstawowe funktry logiczne



Rys.4. Inwerter (negator) wykonany z bramek NAND: a) schemat połączeń, b) charakterystyka przejściowa dla układów TTL, c) wymagane przedziały napięć dla układów TTL

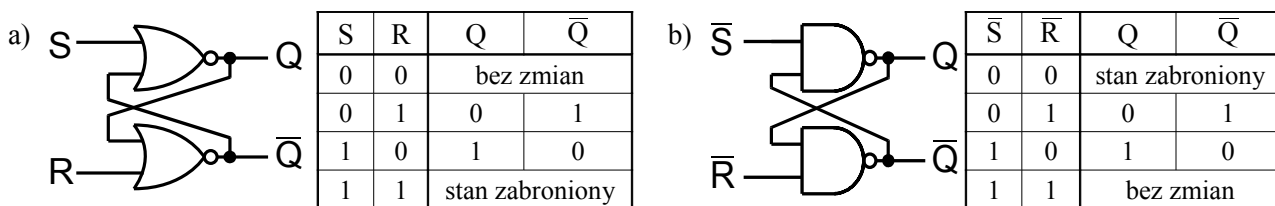
W zestawie elementów logicznych **UNILOG 2** będącym na wyposażeniu stanowiska laboratoryjnego [2], do budowy modułów logicznych wykorzystano scalone układy TTL. W module logicznym 7400 zamontowano układ scalony **UCY7400** zawierający 4 dwuwejściowe bramki NAND. Układ scalony ma prostokątną obudowę i 14 wyprowadzeń. Dwa z nich oznaczone +5V i 0 V służą do zasilania układu napięciem 5 V, do pozostałych dołączono wejścia i wyjścia bramek. W module logicznym 7404 zamontowano układ scalony **UCY7404**^{*)} zawierający 6 bramek NOT (negatorów) – rys. 5.



Rys.5. Rozkład wyprowadzeń układów scalonych: a) UCY7404, b) UCY7400

2. Przerzutniki

Prostymi elementami pamięciowymi są przerzutniki (rys. 6). Są to układy dwustanowe. Wyjście przerzutnika pod wpływem wymaganej kombinacji sygnałów na jego wejściach może zmienić swój stan z 0 na 1 lub z 1 na 0. Wyróżnia się dwa rodzaje przerzutników: monostabilne i bistabilne. Przerzutnik bistabilny może w każdym z dwóch stanów pozostawać przez nieograniczenie długi czas. Natomiast przerzutnik monostabilny tylko pewien zaprogramowany czas, po którym samoistnie powraca do stanu spoczynkowego.



Rys.6. Przerzutnik RS i jego tablica prawdy zrealizowany na: a) bramkach NOR, b) bramkach NAND

Przerzutniki monostabilne mają zdolność zapamiętania pojedynczego zdarzenia w postaci jednego z dwóch stanów: 0 lub 1 logicznej – 1 bitu informacji. Przykładem jest przerzutnik **RS** – rys.6a. Składa się z dwóch bramek typu NOR. Posiada dwa wejścia informacyjne: **S** (Set) i **R** (Reset) oraz dwa wyjścia: **Q** i \bar{Q} . Wejścia **R** i **S** są wejściami asynchronicznymi, oznacza to, że zmiana stanu na wyjściu następuje bezpośrednio po wystąpieniu aktywnego stanu na odpowiednim wejściu – tu 1 logicznej. Podobnie działa przerzutnik złożony z dwóch bramek NAND – rys. 6b. W tym przypadku stanem aktywnym jest 0 logiczne – stąd spotykane oznaczenie przerzutnika \bar{R}, \bar{S} .

Przerzutniki synchroniczne oprócz wejść informacyjnych posiadają wejścia synchronizujące nazywane zegarowymi (*ang. Clock*) lub taktującymi **T**. Wykorzystuje się je tam, gdzie zachodzi potrzeba zapamiętania

*) Standardowe oznaczenia układów TTL: pierwsze litery oznaczają producenta: np. UCY – CEMI, SN – Texas Instruments, następne dwie cyfry (74) wykonanie komercyjne, opcjonalnie następne litery to technologia np. LS oraz ostatnie dwie lub trzy następne cyfry oznaczają funkcję realizowaną przez układ – np. 00 cztery bramki NAND

Ćwiczenie nr 14. Podstawowe funkcory logiczne

żądaną informacji w wybranej chwili czasu, czyli w momencie wystąpienia sygnału zegarowego. Są to przerzutniki typu D, JK.

Przerzutniki JK najczęściej występują w wersji M-S (Master-Slave). Taki przerzutnik tworzą dwa synchroniczne przerzutniki połączone kaskadowo. Przerzutniki typu D, JK są podstawowymi elementami bardziej złożonych układów takich jak: liczniki^{*)}, rejestry przesuwne, pamięci^{**)}. W praktyce nie buduje się liczników z pojedynczych przerzutników lecz korzysta się z gotowych układów liczących.

3. Zadania pomiarowe – do wyboru przez prowadzącego



1. Zmierzyć napięcie na wyjściu bramki (NOT, NAND lub NOR) odpowiadające stanom **0** i **1** logicznej. Porównać z wymaganymi przedziałami – dane na rys.4.
2. Wyznaczyć doświadczalnie tablicę prawdy dla funkcorów: NOT, NAND i NOR.
3. Zbudować bramkę AND i NOR używając bramek NAND. Wyznaczyć tablicę stanów.
4. Sprawdzić doświadczalnie prawa de Morgana - skorzystać z równoważności między funkcjami.
5. Zrealizować za pomocą funkcorów i wyznaczyć tablicę stanów dla następujących funkcji:

$$Y1 = \overline{(A+B)} \cdot C, \quad Y2 = \overline{(A+B)} \cdot \overline{C}, \quad Y3 = A \cdot \overline{B} + C, \quad Y4 = \overline{(A+B)} \cdot \overline{C}$$

Uwaga 1.

1. Wszystkie połączenia na makiety są realizowane przy użyciu specjalnych kabelków. Kabelek tworzy giętki pojedynczy przewód w izolacji zakończony na obu końcach miniaturowymi gniazdami umieszczonymi w izolacyjnej koszulce.
2. W celu utworzenia połączenia należy delikatnie nałożyć gniazdo przewodu na końcówkę wyprowadzenia, która ma postać kilkumilimetrowej szpilki tak, aby nie złamać tej szpilki.
3. Rozłączając połączenia, należy uchwycić łączący przewód za gniazdo tak, aby nie uszkodzić połączenia przewodu z gniazdem.

Uwaga 2.

1. W dolnej części makiety znajduje się 8 układów generujących stany logiczne 0 lub 1 (SWITCH REGISTER). Każdy układ posiada przełącznik dwustanowy i związane z nim dwa komplementarne wyjścia oznaczone symbolami: , .
2. Nad wyjściami umieszczono dwa rzędy wskaźników stanów logicznych (DISPLAY REGISTER) – po 8 sztuk w rzędzie. Wskaźnik emituje światło, gdy na jego wejściu jest **1** logiczna. W celu utworzenia połączenia należy delikatnie nałożyć gniazdo przewodu na końcówkę wyprowadzenia, która ma postać kilkumilimetrowej szpilki tak, aby nie złamać tej szpilki.
3. Do sprawdzania poprawności działania budowanych układów należy wykorzystać ww. wskaźniki stanów dołączając je do wejść i wyjść badanych układów.

4. Pytania kontrolne

1. Opisać funkcje logiczne podstawowych funkcorów – symbol graficzny, tablica stanów.
2. Zbudować bramkę AND i NOR używając tylko bramek NAND.
3. Uprościć funkcje logiczne: a) $\overline{A} \cdot \overline{B} + \overline{A} + B$, b) $(A + A \cdot B) + \overline{A} \cdot B$, c) $[(A \cdot B + A \cdot \overline{B}) + A \cdot B] + \overline{A} \cdot B$
4. Wymienić i opisać wymagania dotyczące układów logicznych serii TTL – napięcia zasilające, poziomy napięć dla zera i jedynki logicznej, obciążalność.

5. Literatura

- [1] Chwaleba A, Poniński M, *Metrologia elektryczna*, WNT, Warszawa 2007
- [2] Krasiński W, *Doświadczenia z podstaw techniki cyfrowej. Instrukcja dydaktyczna modułowego zestawu elementów logicznych UNILOG-2*, Warszawa 1986
- [3] Stacewicz T, Kotliński W, *Elektronika w laboratorium naukowym*, PWN, Warszawa 1994

6. Wyposażenie stanowiska laboratoryjnego

Zestaw elementów logicznych UNILOG-2
 Woltomierz napięcia stałego,
 Oscyloskop dwukanałowy

Opracowała: mgr inż. Beata Krzywaźnia

Instytut Inżynierii Biomedycznej i Pomiarowej Wydziału PPT Politechniki Wrocławskiej

*) Na N przerzutnikach można zbudować licznik binarny o pojemności do 2^N , liczba zliczanych impulsów zawiera się od 0 do $2^N - 1$.

**) Do zapisania jednego 8 bitowego słowa – 1 bajtu potrzeba 8 przerzutników. Stąd liczba wymaganych przerzutników do zapisania informacji równej: 1 kB = $1024 \cdot 8 = 8192$, 1 MB = $1024 \cdot (1024 \cdot 8) = 8\,388\,608$ przerzutników.