

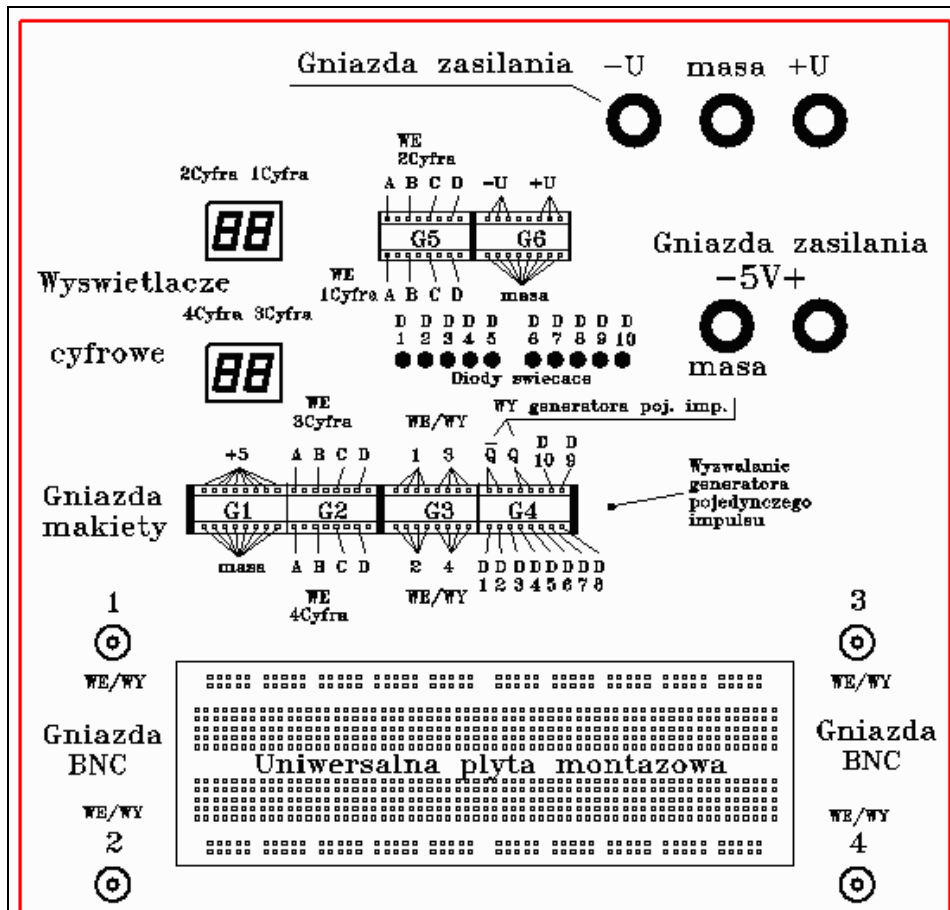
## INSTRUKCJA WYKONAWCZA

do ćwiczenia

# „Cyfrowe układy scalone”

## Wstęp

Celem ćwiczenia jest zaznajomienie studentów z podstawami techniki cyfrowej. Ćwiczenie wykonuje się na uniwersalnej mackiecie pozwalającej montować układy elektroniczne bez użycia połączeń lutowanych.



Rys. 1 Płyta czołowa uniwersalnej mackiety

Mackieta działa prawidłowo po doprowadzeniu do jej zacisków, oznaczonych symbolami (+ 5V -) napięcia ( $5 \pm 0,25$ ) V z zewnętrznego zasilacza. Przed podłączeniem mackiety do zasilacza należy za pomocą woltomierza sprawdzić wartość napięcia na zaciskach zasilacza. Prawidłowe zasilanie mackiety sygnalizuje świecenie diod  $D_1 \dots D_{10}$ .

Umieszczone na makiecie montażowej gniazda  $G_1$ -  $G_4$  spełniają następującą rolę:

- Gniazdo  $G_1$  podaje napięcie zasilające (+5V) i masę układu.
- Gniazdo  $G_2$  i  $G_5$  dekodera BCD obsługuje dwa wyświetlacze cyfrowe. Odpowiednie cyfry dziesiętne w kodzie BCD (Binary Coded Decimal 8421) wyświetlane są na wyświetlaczu cyfrowym po podaniu na jego wejścia słowa czterobitowego (wejścia A, B, C, D).
- Gniazdo  $G_3$  obsługuje cztery koncentryczne gniazda wejścia / wyjścia typu BNC.
- Gniazdo  $G_4$  służy do detekcji stanów logicznych. Stany te po doprowadzeniu do wejść diod świecących  $D_1$  -  $D_{10}$  wyświetlane są na makiecie zgodnie z konwencją logiczną: „0” logiczne = dioda nie świeci, „1” logiczna = dioda świeci. Należy jednak zauważyć, że zgodnie ze standardem TTL, niepodłączone wejście układu cyfrowego jest rozumiane jako logiczna „1” i powoduje także świecenie diody.
- Wyjścia Q oraz  $\sim Q$  (zaprzeczenie Q) gniazda  $G_4$  są wyjściami ręcznie wyzwalanego generatora impulsów pojedynczych. Po naciśnięciu przycisku "Wyzwalanie generatora pojedynczego impulsu", na wyjściu Q pojawi się pojedynczy przebieg w standardzie TTL o postaci: "  $\square$  " , a jednocześnie na wyjściu  $\sim Q$  impuls "  $\square$  " .
- Dodatkowe trzy gniazda zasilające (-U, +U, masa) służą do doprowadzenia napięć poza standardem TTL.

Montażu układów elektronicznych dokonuje się na uniwersalnej płytce montażowej, na której umieszcza się układy scalone, oporniki, kondensatory, diody itp. Łączenie elementów elektronicznych wykonuje się za pomocą zewnętrznych kabelków zakończonych cienkimi końcówkami lub wykorzystując już przygotowane wewnętrzne połączenia w makiecie. Za pomocą kabelków doprowadza się do układów napięcie zasilające z gniazd makiety oraz podłącza odpowiednie wejścia / wyjścia (np. generator, oscyloskop).

## **Część wykonawcza**

Zasadniczym zadaniem ćwiczenia jest zbudowanie jednego z urządzeń elektronicznych zaproponowanych w punkcie XI części wykonawczej tej instrukcji. Studenci mogą także wykonywać układy zaproponowane i opracowane przez siebie, o ile zostaną one zaakceptowane przez asystenta. Z tej części ćwiczenia należy wykonać sprawozdanie.

Układy w ćwiczeniach należy projektować tak, by do realizacji funkcji wykorzystać możliwie najmniejszą liczbę układów scalonych – czyli doprowadzić funkcję do postaci minimalnej. Można uzyskać to metodą przekształceń algebraicznych lub o wiele prościej, metodą graficzną tablic Karnaugh. W projekcie można posługiwać się układami scalonymi o różnej liczbie wejść, ale najczęściej będą to podstawowe bramki typu NAND, NOR, inwertory NOT oraz bramki EXCLUSIVE OR.

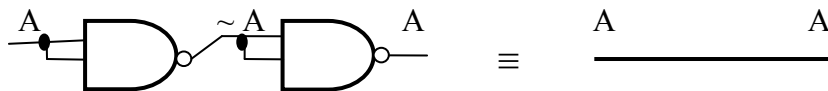
Projekt rozpoczynamy od narysowania schematu blokowego i napisania tabelki prawdy dla poszczególnych bloków. Następnie rozwijamy schemat blokowy na schemat logiczny realizowany na elementach NAND (lub innych podanych w założeniach zadania) z zastosowaniem zasad minimalizacji. Zaczynamy uruchamianie zbudowanego układu wykorzystując diody LED z makiety jako pomocnicze wyświetlacze stanów binarnych. W początkowej fazie uruchamiania wykorzystujemy generator pojedynczych impulsów z

makiety. Następnie podłączamy generator zewnętrzny i obserwujemy przebiegi na oscyloskopie.

Przed przystąpieniem do budowania układu należy narysować projekt, z naniesieniem pozycji układów scalonych na makiiecie i podaniem konkretnych numerów „nózek” bramek wykorzystanych w połączeniach.

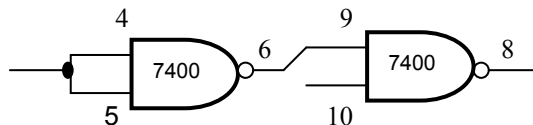
Skomplikowany projekt uruchamiamy metodą podziału na bloki - uruchamiamy na przykład blok licznika, blok dekodera, potem łączymy bloki i uruchamiamy całość. Na koniec dobudowujemy dodatkowe funkcje pozwalające na wstępne ustawienie zadanych wartości (SET) lub wyzerowanie układu (RESET).

Staramy się jeszcze „optycznie” zminimalizować powstałe po połączeniu układy pamiętając, że negacja negacji zmiennej daje zmienną pierwotną.



Do badania stanów wybranych punktów w układzie, przydatna może okazać się „sonda” – długi kabelek podłączony do wejścia kanału oscyloskopu (lub diody świecącej) ustawionego w modzie DC, którym dotykamy wybrane wejścia i wyjścia bramek.

Przykład : Podłączyć generator impulsów do nóżki 4 pierwszej bramki NAND, dotknąć sondą nóżki 5, 6 oraz 9, 10, 8 drugiej bramki NAND i oglądać na oscyloskopie, czy sygnały są zgodne z oczekiwaniami.



## ĆWICZENIA

**I. Zbadać tabelę prawdy dla dwu-wejściowych bramek NAND oraz NOR**  
( NAND = Not AND - UCY 7400; NOR = Not OR - UCY 7402 ).

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	
0	1	
1	0	
1	1	

Ręcznie podawać sygnały "0" lub "1" na wejścia X<sub>1</sub> X<sub>2</sub> i obserwować diodę świecąca podłączoną do wyjścia bramki.

Następnie podłączyć na jedno z wejść periodyczny sygnał prostokątny z generatora a na drugie wejście ręcznie podawać "0" lub "1". Zaobserwować istotę działania „bramki” przepuszczającej lub blokującej impulsy.

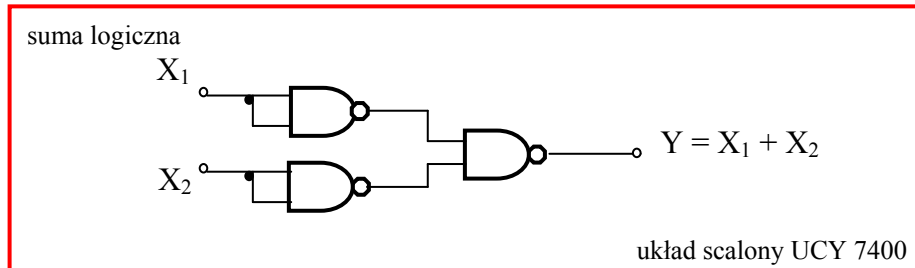
**UWAGA** : sygnał fali prostokątnej z generatora powinien zawierać się w granicach "0" = (0 - 0.2 - 0.4) V, "1" = (2.4 - 3.4 - 5) V ; wartości typowe wytłuszczono.

**II. Zbudować z bramek NAND poniższe funkcje logiczne i zbadać tabelki prawdy.**  
Na schemat wpisać odpowiednie numery nóżek bramek.

**Uwaga:** Dwie pierwsze funkcje zostały przykładowo wykonane.

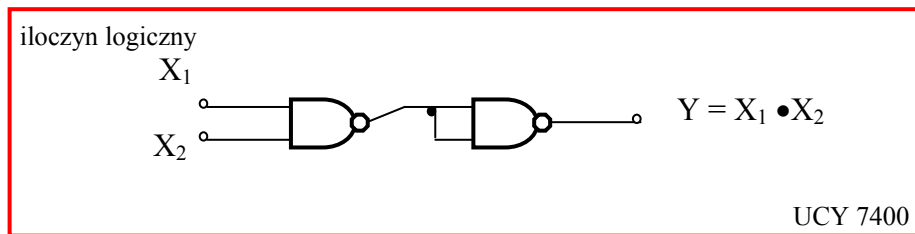
**1) Suma logiczna**

$$Y = X_1 + X_2$$



**2) Iloczyn logiczny**

$$Y = X_1 \cdot X_2$$



**3) Implikacja [  $X_2 \rightarrow X_1$  ]**

$$Y = X_1 + \sim X_2$$

**4) Zakaz  $X_2$  przez  $X_1$**

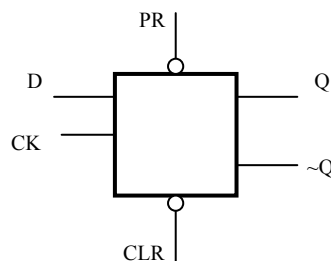
$$Y = \sim X_1 \cdot X_2$$

**5) Suma *Exclusive Or***

$$Y = X_1 \cdot \sim X_2 + \sim X_1 \cdot X_2$$

Uwaga: Funkcja ta realizowana jest bezpośrednio na układzie scalonym UCY 7486

**III. Wypełnić tabelkę przejść dla przerzutnika typu „D” (układ scalony UCY 7474).**



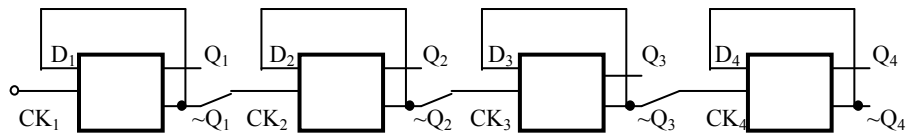
D	$Q^t$	$Q^{t+1}$
0	0	
0	1	
1	0	
1	1	

Wprowadzając dodatkowe sprzężenia zmienić przerzutnik „D” w przerzutnik „T”.

W przerzutniku „T” podłączyć do wejścia zegarowego (CK) impulsy periodyczne prostokątne z generatora i zaobserwować na oscyloskopie, które zbocze impulsu zegarowego jest aktywne i powoduje zmianę stanu wyjścia.

#### IV. Zbudować 4 bitowy licznik szeregowy na przerzutnikach typu „D” (UCY 7474).

Wpisać numery nóżek na schemacie. Na wejście podłączyć generator pojedynczych impulsów z makiety. Wyjścia Q z licznika podłączyć do wyświetlaczy na diodach.



Nr impulsu	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
0	0	0	0	0

Następnie podłączyć na wejście zewnętrzny generator i zaobserwować przebiegi Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub>, Q<sub>4</sub> za pomocą oscyloskopu. Dobudować dodatkowe zerowanie licznika wykorzystując generator pojedynczych impulsów i asynchroniczne wejścia CLR (Clear) przerzutników.

Przerobić licznik na liczący „do tyłu” łącząc Q-CK (zamiast ~Q-CK) i zaobserwować zmianę stanów.

Powrócić do poprzedniego układu licznika.

**UWAGA : Nie rozłączać licznika.** Wykorzystać wyjścia licznika Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub>, Q<sub>4</sub> jako zmienne wejściowe w następnych punktach ćwiczenia.

#### V. Zbudować na 2-wejściowych bramkach NAND cztero bitowy dekodер liczby równej "numerowi stanowiska + 7".

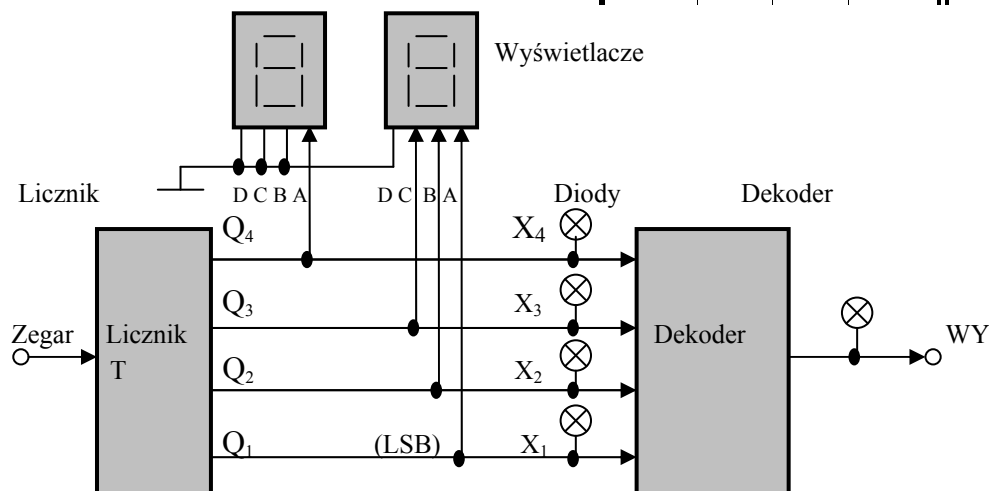
Jako wejście na dekodер zastosować 4 bitowy licznik wykonany w poprzednim ćwiczeniu. Stany wejściowe na dekodер (z licznika) wyświetlić na diodach oraz podłączyć je do wyświetlacza BCD, zgodnie z podanym rysunkiem. Do wyjścia z dekodera podłączyć diodę świecącą.

Sprawdzić i zanotować wskazania w pełnym cyklu licznika na diodach i wyświetlaczu.

Zapisać numer swojego "stanowiska +7" binarnie i oktalnie.

Najpierw ułożyć tabelkę prawdy

X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	WY



Wskazówka 1

Najpierw uruchomić licznik. Przy podłączeniach wyjść licznika na wyświetlacze zwrócić uwagę na kolejność podłączanych bitów – od najmniej znaczącej pozycji *LSB* (Least Significant Bit) do najbardziej znaczącej pozycji *MSB* (Most SB).

Wskazówka 2

Do uruchamiania układu, jako wejście zegarowe licznika zastosować generator pojedynczego impulsu z makiety. Następnie podłączyć w to miejsce sygnał z generatora zewnętrznego o częstotliwości nie przekraczającej 10 Hz.

Wskazówka 3

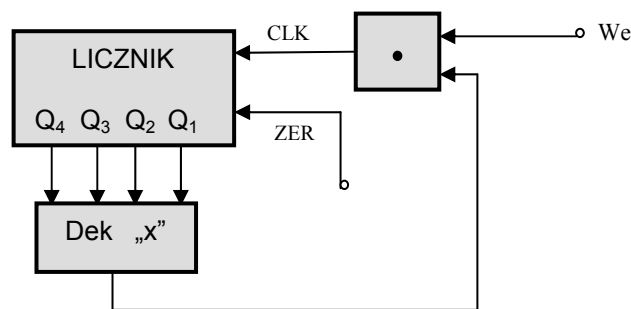
Wyświetlacz BCD w tym podłączeniu wyświetla liczbę binarną w systemie oktalnym. Grupując bity w triady ( $2^3 = 8$ ) i odczytując liczby w każdej triadzie otrzymamy liczbę oktalnie.

binarnie : 101 111 001

oktalnie : 5 7 1

czyli  $101111001_{(2)} = 571_{(8)}$

## VI. Zbudować licznik liczący do liczby równej "numerowi stanowiska + 7" i automatycznie zatrzymujący się.



**UWAGA :** Zmodyfikować poprzednie zadanie zgodnie z powyższym schematem.

Wskazówka 1

Automatycznie zatrzymujący się - czyli blokujący dalsze impulsy zegara.

Wskazówka 2

Do ponownego uruchomienia zablokowanego układu dobudować sygnał ZER powodujący zerowania licznika z generatora jednego impulsu, używając właściwego wyjścia generatora (o odpowiedniej fazie sygnału).

## VII. Zbudować licznik modulo "numer stanowiska + 7" czyli liczący „w kółko”.

**UWAGA :** Zmodyfikować poprzednie zadanie.

Wskazówka 1

Modulo 10 oznacza liczenie 0, 1, 2 ... 8, 9 → 0, 1, ....

Wskazówka 2

Wyjście z dekodera podłączyć do wejścia zerowania licznika.

**VIII. Zbudować jeden z poniższych układów zaproponowany przez asystenta.**

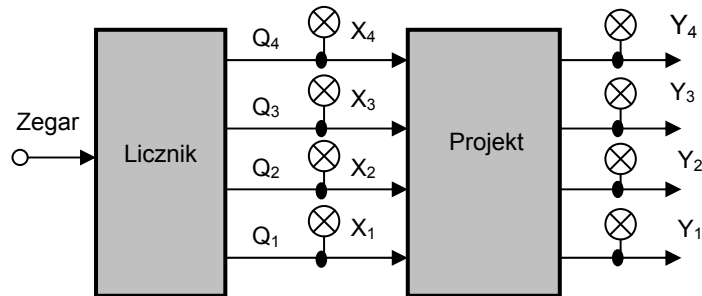
Rozpocząć budowę od napisania tabelki prawdy. Zminimalizować postać funkcji. Zastosować wcześniej zbudowany licznik (wyjścia  $Q_1, Q_2, Q_3, Q_4$ ) jako wejście układów.

- a) Zbudować układ wykrywający liczbę binarną, gdy jest ona parzysta lub podzielna przez 3 dla trzech zmiennych wejściowych (liczba trzy bitowa).

$X_3$	$X_2$	$X_1$	Y

Dla liczby binarnej (000) przyjąć również wartość funkcji  $Y = 1$

- b) Zbudować układ kombinatoryczny, zmieniający wyjście 4 bitowego dodającego licznika (0, 1, 2, 3, 4, 5 itd.) na liczenie parzystości (0, 2, 4, 6, 8, itd.) lub nieparzystości (1, 3, 5, 7, itd.) w zależności od tego, czy numer stanowiska jest parzysty czy nieparzysty.



Tabelka prawdy przyjmie postać:

$X_4$	$X_3$	$X_2$	$X_1$	$Y_4$	$Y_3$	$Y_2$	$Y_1$

- c) Zaprojektować na elementach NAND układ kombinatoryczny, zamieniający naturalny kod binarny na kod Grey'a, czyli taki, w którym każda następna kombinacja różni się od poprzedniej zmianą na jednym tylko bicie. Warunki te spełnia na przykład ciąg 000, 001, 011, 010, 110, 111, 101, 100. Jako wejście zastosować 3 bitowy licznik.

Najpierw ułożyć tabelkę prawdy

$X_3$	$X_2$	$X_1$	$Y_3$	$Y_2$	$Y_1$

**IX. Zbadać tabelkę przejść przerzutnika „JK” (UCY 7473 → UCY 7493).**

J	K	$Q^t$	$Q^{t+1}$

Dokonać modyfikacji zamieniających ten przerzutnik w przerzutnik typu „D” oraz „T”.

Następnie wykorzystując układ UCY 7493 połączyć 4-bitowy licznik szeregowy. Podłączyć na wejście zegarowe zewnętrzny generator i zaobserwować zależności między przebiegami  $Q_1, Q_2, Q_3, Q_4$  na oscyloskopie.

Wskazówka 1

Dla poprawnej pracy 4-bitowego licznika (7493) należy wprowadzić modyfikację łączącą dwa niezależne bloki ( $Q_A - B$ ) oraz zablokować zerowanie (uziemić np. wejście 2 –  $R_{0(1)}$ )

**X. Na podsumowanie ćwiczeń z przerzutnikami wypełnić pozostałe pola tablicy przerzutników zgodnie z przykładem:**

$Q^t \rightarrow Q^{t+1}$	D	T	R S	J K
0 0	0	0	- 0	0 -
0 1				
1 0				
1 1				

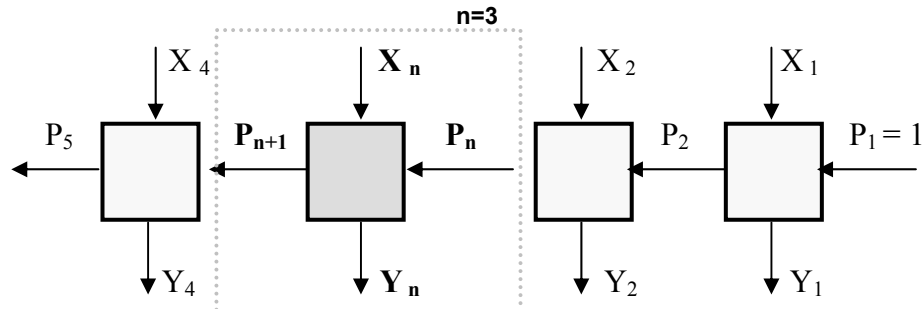
**UWAGA :** "-" oznacza stan dowolny na danym wejściu, 0 lub 1



**XI. Zaprojektować, rozrysować na elementach logicznych i wykonać jeden z projektów proponowanych poniżej wykorzystując układy cyfrowe dostępne w pracowni. (spis podany na końcu instrukcji)**

**1. Układ czterobitowy następnika liczby binarnej** (dający na wyjściu liczbę o 1 większą) składający się z czterech bloków iteracyjnych.

Na wejście podłączyć licznik czterobitowy, wyświetlać stany wejściowe i wyjściowe na diodach.



$X_n$  - wejściowy n-ty bit liczby ;  $P_n$  - przeniesienie z poprzedniego bitu liczby  
 $Y_n$  - wyjściowy n-ty bit liczby ;  $P_{n+1}$  - przeniesienie do następnego bitu liczby

Zauważyć zasadę działania następnika :

<b>X</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	- do pierwszego napotkanego od prawej strony zera włącznie neguje bity na przeciwne
<b>Y</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	

Uprościć pierwszy blok podstawiając warunki początkowe przeniesienia  $P_1 = 1$  (czyli tak jakby liczba miała jeszcze jedynki z prawej strony) do funkcji  $P_{n+1}(X_n, P_n)$  i  $Y_n(X_n, P_n)$ . Najpierw, po zdefiniowaniu logiki działania bloku, ułożyć tabelkę prawdy.

$X_n$	$P_n$	$Y_n$	$P_{n+1}$

Wskazówka 1

Rozpocząć budowę od uruchomienia licznika. Przy podłączeniach wyjść licznika na wyświetlacze zwrócić uwagę na kolejność podłączanych bitów – od najmniej znaczącej pozycji *LSB* (Least Significant Bit) do najbardziej znaczącej pozycji *MSB* (Most SB)

Wskazówka 2

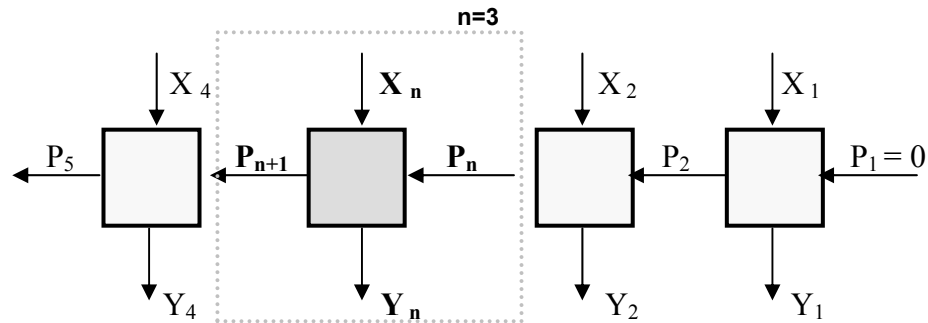
Budować układ pojedynczo dodając i uruchamiając kolejne bloki.

Wskazówka 3

Do uruchamiania układu, jako wejście licznika, zastosować generator pojedynczego impulsu z makiety uruchamiany ręcznie, następnie podłączyć w to miejsce generator zewnętrzny o częstotliwości postrzeganej optycznie na diodach świecących (z dolnego zakresu 10 Hz)

**2. Układ czterobitowego poprzednika liczby binarnej** (dający na wyjściu liczbę o 1 mniejszą) składający się z czterech bloków iteracyjnych.

Na wejście podłączyć licznik czterobitowy, wyświetlać stany wejściowe i wyjściowe na diodach.



$X_n$  - wejściowy n-ty bit liczby ;  $P_n$  - przeniesienie z poprzedniego bitu liczby  
 $Y_n$  - wyjściowy n-ty bit liczby ;  $P_{n+1}$  - przeniesienie do następnego bitu liczby

Zauważyć zasadę działania poprzednika :

```

      ↖ ↗
      X  1 0 1 0 0
      Y  1 0 0 1 1
  
```

- do pierwszej napotkanej od prawej strony  
 jedynek włącznie neguje bity na przeciwne

Uprościć pierwszy blok podstawiając warunki początkowe przeniesienia  $P_1 = 0$  (czyli tak jakby liczba miała jeszcze zera z prawej strony) do funkcji  $P_{n+1}(X_n, P_n)$  i  $Y_n(X_n, P_n)$ .  
 Najpierw, po zdefiniowaniu logiki działania bloku, ułożyć tabelkę prawdy.

$X_n$	$P_n$	$Y_n$	$P_{n+1}$

Wskazówka 1

Rozpocząć budowę od uruchomienia licznika. Przy podłączeniach wyjść licznika na wyświetlacze zwrócić uwagę na kolejność podłączanych bitów – od najmniej znaczącej pozycji *LSB* (Least Significant Bit) do najbardziej znaczącej pozycji *MSB* (Most SB).

Wskazówka 2

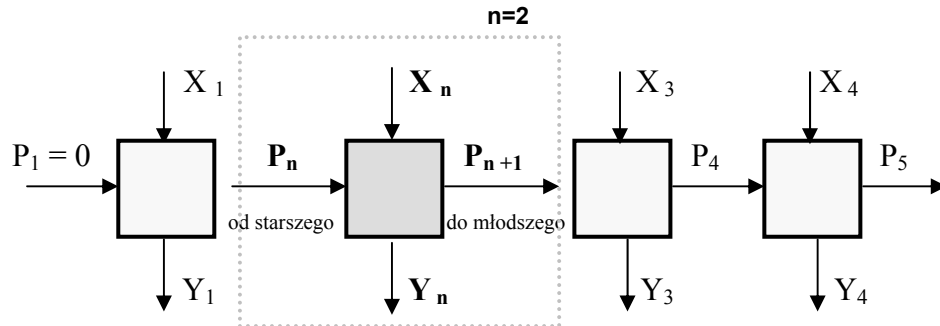
Budować układ pojedynczo dodając i uruchamiając kolejne bloki.

Wskazówka 4

Do uruchamiania układu, jako wejście licznika zastosować generator pojedynczego impulsu z makiety uruchamiany ręcznie, następnie podłączyć w to miejsce generator zewnętrzny o częstotliwości postrzeganej optycznie na diodach świecących (z dolnego zakresu 10 Hz)

### 3. Układ czterobitowy priorytetowego przekazywania informacji składający się z czterech bloków iteracyjnych.

Na wejście podłączyć licznik binarny, wyświetlać stany wejściowe i wyjściowe na diodach..



$X_n$  - wejście n-te priorytetowe ;  $P_n$  - przeniesienie od starszego  
 $Y_n$  - wyjście n-te priorytetowe ;  $P_{n+1}$  - przeniesienie do młodszego

Zauważyć zasadę działania bloku priorytetowego - gdy najstarszy w danym momencie ( $P_n = 0$ ) mówi ( $X_n = 1$ ) to go słyhać ( $Y_n = 1$ ) i blokuje ( $P_{n+1} = 1$ ) młodszego (na prawo).

Uprościć pierwszy („najstarszy”) blok podstawiając warunki początkowe przeniesienia  $P_1 = 0$  (ponieważ nikt go nie może blokować) do funkcji  $Y_n(X_n, P_n)$  oraz  $P_{n+1}(X_n, P_n)$ .  
 Najpierw, po zdefiniowaniu logiki działania bloku, ułożyć tabelkę prawdy.

$X_n$	$P_n$	$Y_n$	$P_{n+1}$

#### Wskazówka 1

Rozpocząć budowę od uruchomienia licznika. Przy podłączeniach wyjść licznika na wyświetlacze zwrócić uwagę na kolejność podłączanych bitów – od najmniej znaczącej pozycji *LSB* (Least Significant Bit) do najbardziej znaczącej pozycji *MSB* (Most SB).

#### Wskazówka 2

Budować układ pojedynczo dodając i uruchamiając kolejne bloki.

#### Wskazówka 3

Do uruchamiania układu, jako wejście licznika zastosować generator pojedynczego impulsu z makiety, następnie podłączyć w to miejsce sygnał z generatora zewnętrznego o częstotliwości nie przekraczającej 10 Hz.

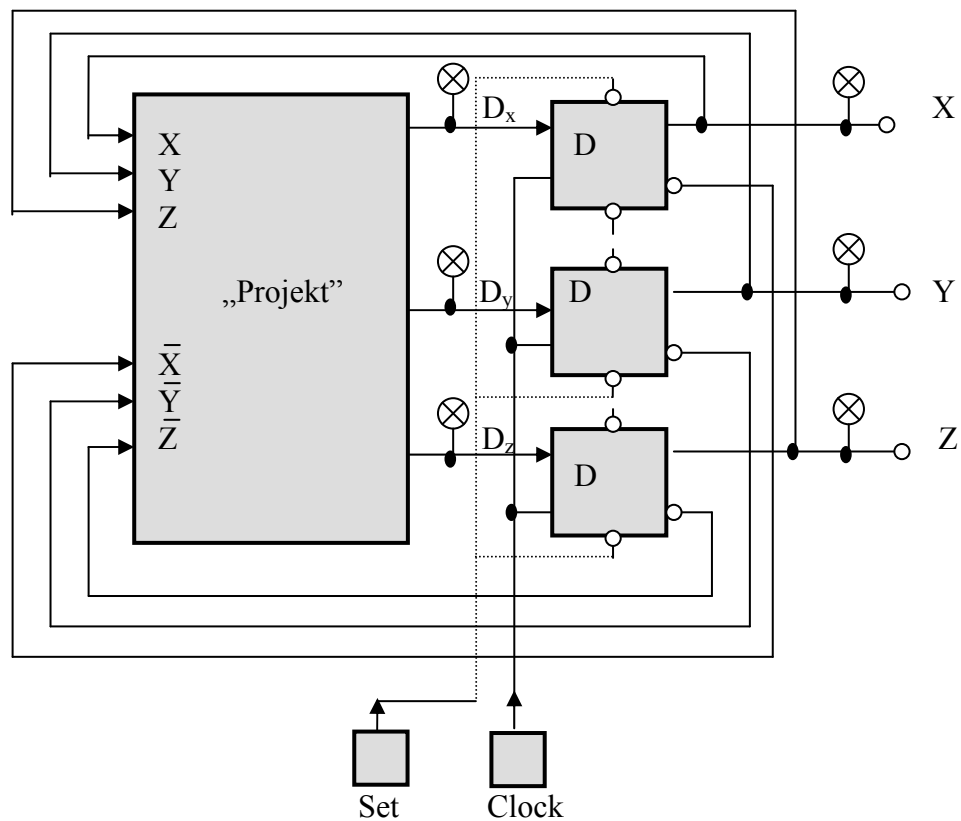
**4. Układ sekwencyjny (z elementami pamięci), symulujący działanie licznika w kodzie naturalnym dwójkowym, na trzech przerzutnikach typu „D”.**

Po naciśnięciu przycisku *Set* układ powinien rozpocząć pracę od liczby binarnej równej numerowi stanowiska np. stół "4" =  $100_{(2)}$ .

Przycisk *Clock* powinien powodować kolejne zmiany stanów na wyjściu przerzutników synchronicznych typu „D”.

X	Y	Z	$D_x$	$D_y$	$D_z$

Schemat układu powinien mieć postać zgodną z poniższym rysunkiem, projektujemy wewnątrz bloku „Projekt”



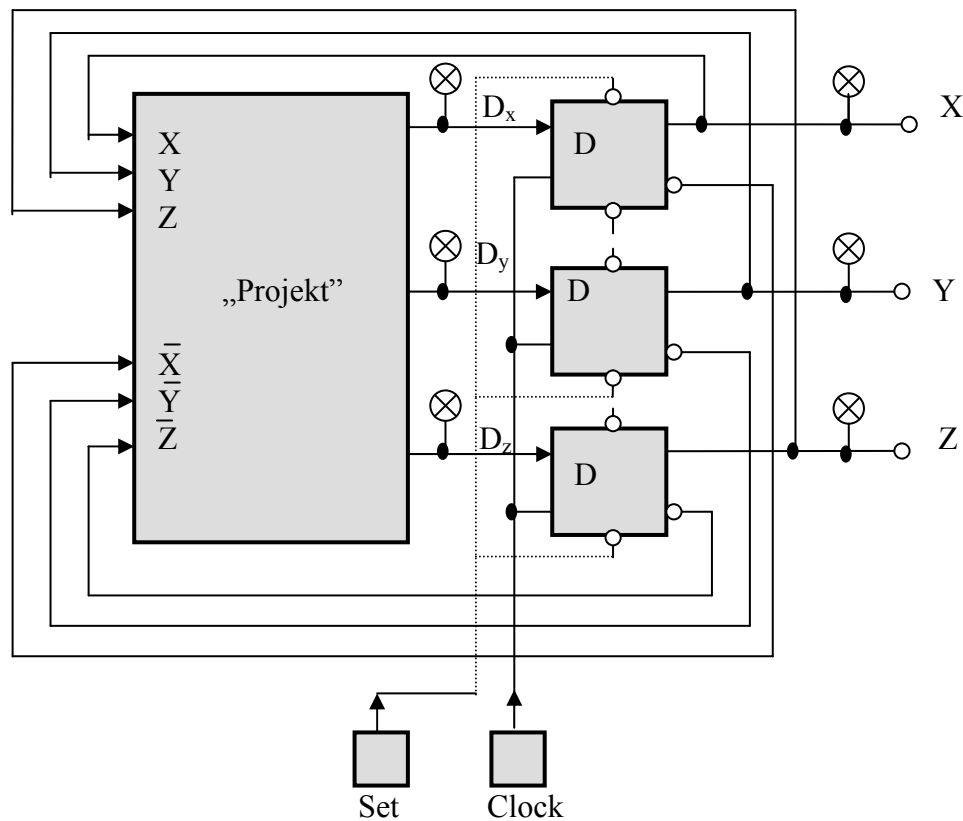
**5. Układ sekwencyjny (z elementami pamięci), generujący na wyjściu kod Grey'a i zbudowany na trzech przerzutnikach typu „D”.**

Po naciśnięciu przycisku *Set* układ powinien rozpoczynać pracę od liczby binarnej równej numerowi stanowiska np. stół "4" =  $100_{(2)}$ .

Przycisk *Clock* powinien powodować kolejne zmiany stanów na wyjściu przerzutników synchronicznych typu „D”.

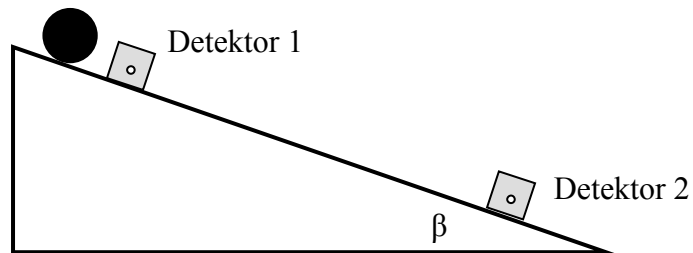
X	Y	Z	D <sub>x</sub>	D <sub>y</sub>	D <sub>z</sub>

Schemat układu powinien mieć postać zgodną z poniższym rysunkiem, projektujemy wewnątrz bloku „Projekt”



### 6. Pomiar czasu przebiegu kulki po równi pochyłej – stoper (UCY 7486).

Układ stopera sprzęgnąć z dostępnym w pracowni modelem równi pochyłej, która wyposażona jest w dwa detektory przebiegu kulki.

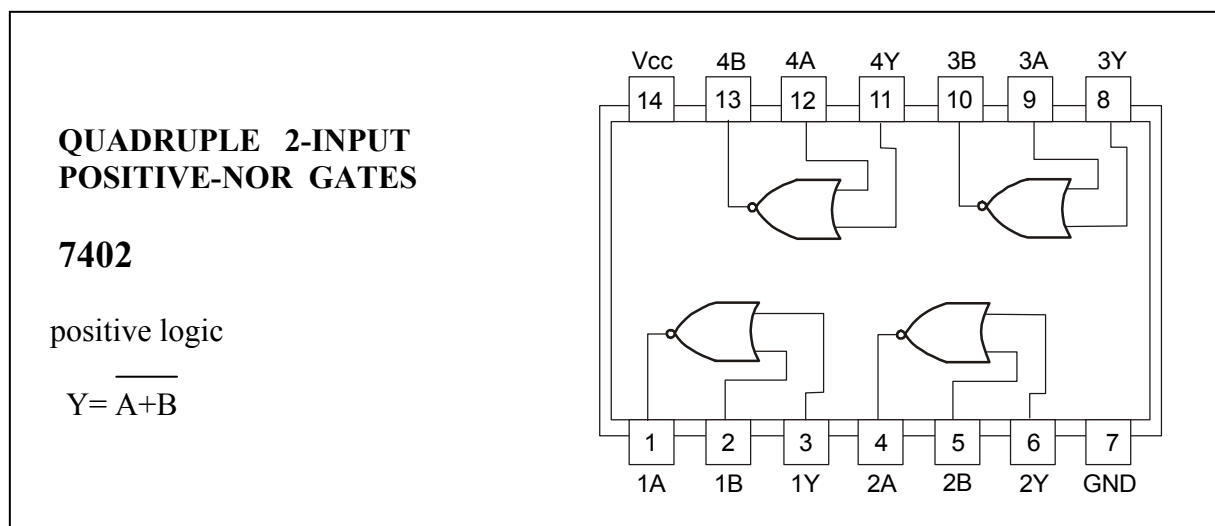
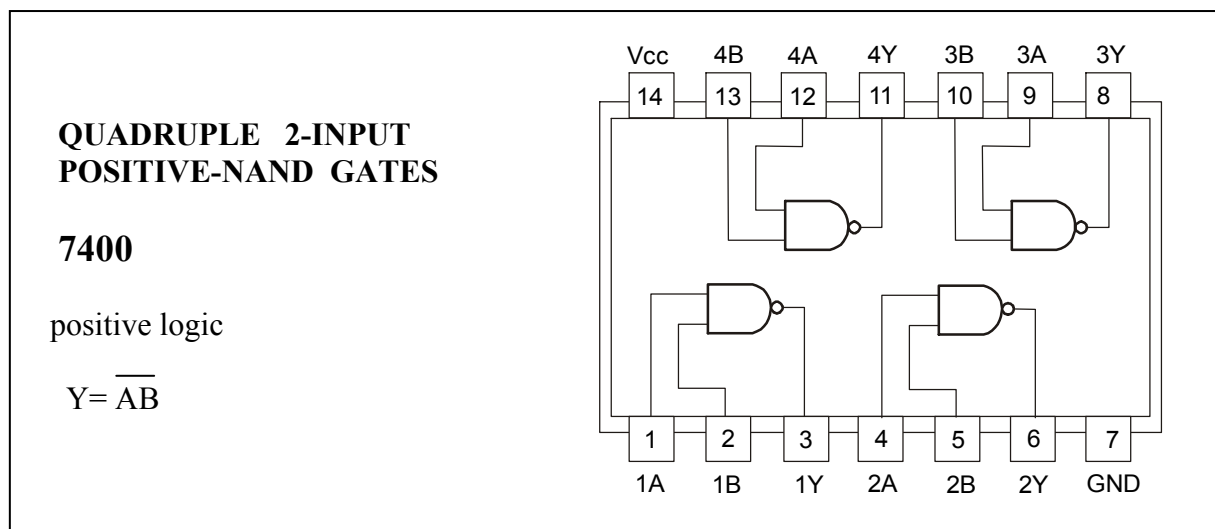


Rys. Równia pochyła

Detektory potwierdzają przelot kulki impulsem TTL w postaci " \_|\_ ".





### SUPLEMENT

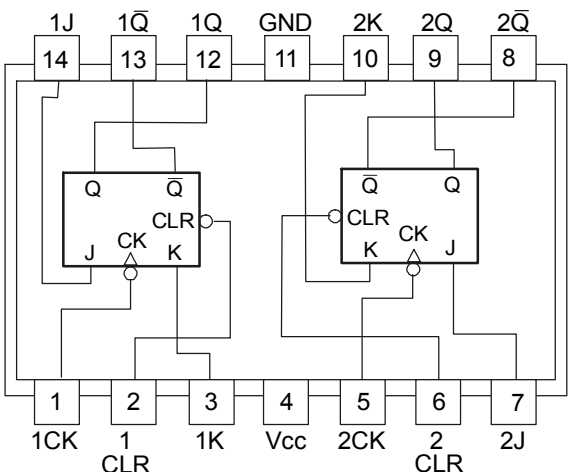
Uproszczone karty katalogowe układów scalonych serii 74xx badanych w ćwiczeniach :



**DUAL J-K FLIP-FLOPS WITH CLEAR**

**7473** FUNCTION TABLE

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	$\sim Q$
L	X	X	X	L	H
H		L	L	$Q_0$	$\sim Q_0$
H		H	L	H	L
H		L	H	L	H
H		H	H	TOGGLE	

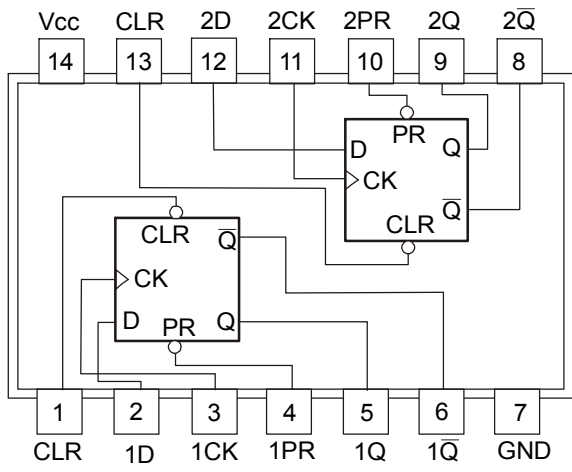


**DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR**

**7474** FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	$\sim Q$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	$\uparrow$	H	H	L
H	H	$\uparrow$	L	L	H
H	H	L	X	$Q_0$	$\sim Q_0$

\* This configuration is nonstable

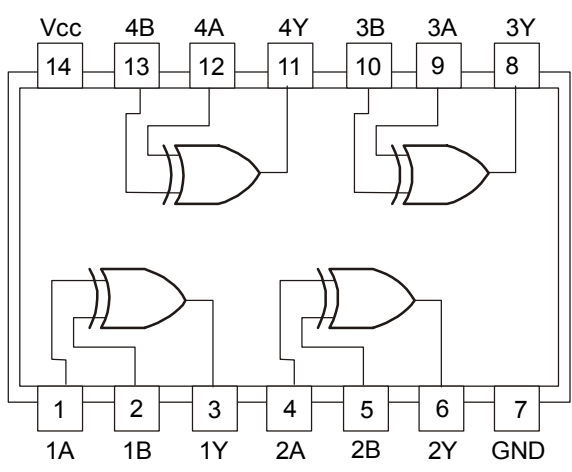


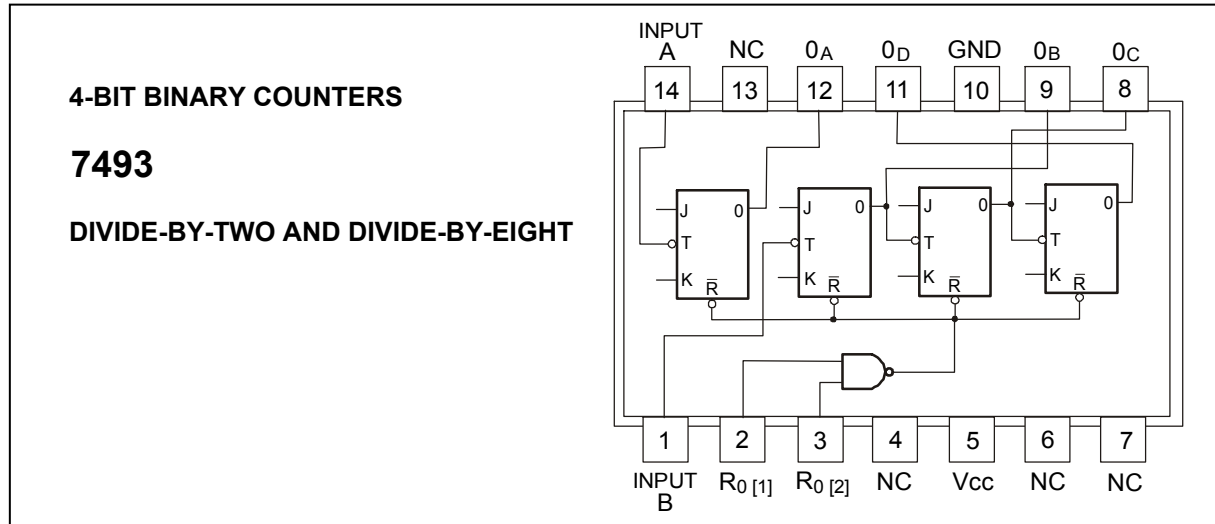
**QUADRUPLE 2-INPUT EXCLUSIVE-OR GATES**

**7486**  $Y = A \oplus B = \bar{A}B + A\bar{B}$

FUNCTION TABLE

INPUTS		OUTPUT
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L





**SPIS UKŁADÓW SCALONYCH SERII 74XX DOSTĘPNYCH W PRACOWNI:**

00, 02, 03, 04, 06, 08, 10, 20, 30, 37, 40, 42, 47, 50, 53, 54, 60, 72, 74, 75, 83, 85, 86,  
90, 92, 93, 107, 121, 123, 132, 145, 150, 151, 153, 155, 157, 174, 175, 180, 181, 192, 193

**LITERATURA**

1. Skrypt „Cyfrowe układy scalone” (Pracownia Elektroniczna)
2. J. Piętkos, J. Turczyński „Układy scalone TTL w systemach cyfrowych”.  
(WKŁ, Warszawa 1980)
3. T. Stacewicz, A. Kotlicki „Elektronika w laboratorium naukowym”. (PWN 1994)
4. W. Sasal „Układy scalone serii UCY 74 parametry i zastosowanie”.  
(WKŁ, Warszawa 1985)
5. J. Kalisz „Cyfrowe układy scalone TTL i ich zastosowanie”
6. W. Anderson „Projektowanie układów z obwodami scalonymi TTL ”
7. Katalog układów “The TTL Data Book for Design Engineers”, Texas Instruments, 1981

\* \* \* \*

*Uprzejmie proszę uwagi i komentarze do materiału ćwiczeń kierować do  
Krzysztofa Sulowskiego*

e-mail: [Krzysztof.Sulowski@fuw.edu.pl](mailto:Krzysztof.Sulowski@fuw.edu.pl)

\* \* \* \*