

Pracownia Konstrukcji i Eksploatacji Urządzeń Cyfrowych

Badanie układów cyfrowych

Badanie układów kombinacyjnych

1. Badanie bramek logicznych NAND, NOR, OR, AND, EXOR, EXNOR, NOT
2. Testowanie i diagnostyka układów cyfrowych
3. Synteza układów kombinacyjnych (minimalizacja funkcji logicznych metodą tablic Karnaugh)
4. Badanie układów komutacyjnych:
 - a. Badanie multiplekserów i demultiplekserów
 - b. Badanie koderów, dekodekserów, transkoderów i wyświetlaczy siedmiosegmentowych typu LED

Sprawozdania wykonał:.....

Klasa:.....

opracowała:
mgr inż. Irena Hoja

rok szk. 2017 /2018

1. Wprowadzenie

W technice cyfrowej ważną grupę układów stanowią układy kombinacyjne.

Układ kombinacyjny jest to układ, w którym aktualny stan wyjść zależy od aktualnego stanu wejść. Najprostszymi układami kombinacyjnymi są bramki realizujące podstawowe funkcje logiczne.





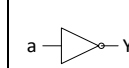
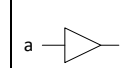
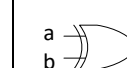
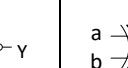
2. Cel ćwiczeń

Celem ćwiczeń jest zapoznanie uczniów z:

- właściwościami i parametrami podstawowych bramek logicznych,
- metodami testowania i diagnostyką układów cyfrowych,
- obsługą testera układów cyfrowych,
- zastosowaniem bramek do projektowania prostych funkcji logicznych,
- metodami optymalnego projektowania układów kombinacyjnych.

3. Wstęp teoretyczny

Bramki logiczne

Symbol bramki								
Nazwa bramki	AND	NAND	OR	NOR	NOT	bufor	EXNOR	EXOR
funkcja	$Y = a \cdot b$	$Y = \overline{a \cdot b}$	$Y = a + b$	$Y = \overline{a + b}$	$Y = \overline{a}$	$Y = a$	$Y = ab + \overline{a}\overline{b}$	$Y = a\overline{b} + \overline{a}b$
a	b	Y	Y	Y	Y	Y	Y	Y
0	0	0	1	0	1	1	0	1
0	1	0	1	1	0	1	0	0
1	0	0	1	1	0	0	1	0
1	1	1	0	1	0	0	1	1

Podstawowe prawa algebry Boole'a

$A + A = A$ $A \cdot A = A$	$A + 0 = A$ $A + 1 = 1$	$A \cdot 0 = 0$ $A \cdot 1 = A$	$A + \overline{A} = 1$ $A \cdot \overline{A} = 0$
$AB + \overline{A}\overline{B} = A$ $(A + B)(\overline{A} + \overline{B}) = A$	<i>Prawa de Morgana</i> $\overline{A \cdot B \cdot C} = \overline{A} + \overline{B} + \overline{C}$ $\overline{A + B + C} = \overline{A} \cdot \overline{B} \cdot \overline{C}$		$\overline{\overline{A}} = A$

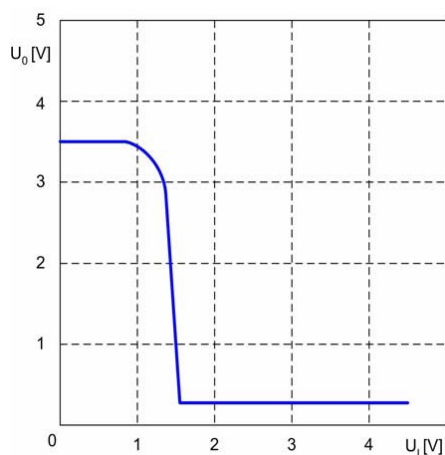
Podstawowe parametry układów cyfrowych to:

- napięcie zasilania
- zakresy napięć stanów logicznych 0 i 1 na wejściu i wyjściu układu
 - ✓ U_{iLmax} - najwyższe dopuszczalne napięcie wejściowe w stanie niskim,
 - ✓ U_{iHmax} - najniższe dopuszczalne napięcie wejściowe w stanie wysokim,
 - ✓ U_{oLmax} - najwyższe dopuszczalne napięcie wyjściowe w stanie niskim,
 - ✓ U_{oHmax} - najniższe dopuszczalne napięcie wyjściowe w stanie wysokim
- moc strat
- marginesy zakłóceń
- obciążalność
 - ✓ I_{oLmax} - największy dopuszczalny prąd wyjściowy w stanie niskim,
 - ✓ I_{oHmax} - największy dopuszczalny prąd wyjściowy w stanie wysokim
- czas propagacji

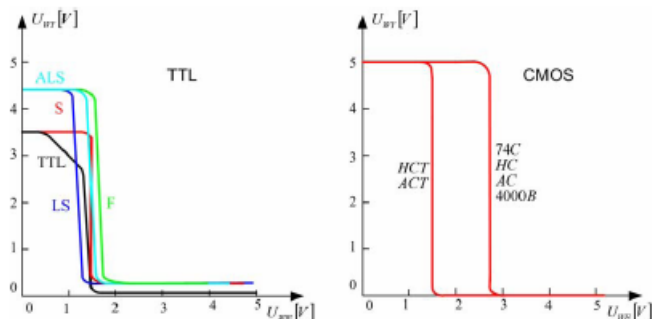
Tabela poniższa przedstawia parametry cyfrowych układów scalonych TTL oraz dla porównania układów wykonanych w technologii CMOS. Wartości tych parametrów są z reguły różne dla różnych rodzin układów cyfrowych i dlatego łączenie odmiennych grup wymaga stosowania układów pośrednich zapewniających „zgodność łączeniową”.

Parametr	Rodzina TTL			Rodzina CMOS			
	LS	ALS	F	4000B 74C	HC HCT	AHC AHCT	AC ACT
Napięcie zasilające U_{CC} [V]	$5 \pm 5\%$	$5 \pm 10\%$	$5 \pm 5\%$	3 - 18	2 - 6	2 - 5,5	2 - 6
Moc strat na bramkę w stanie statycznym P_{typ} [mW]	2	1	5,5	0,001	0,0025	0,0025	0,0025
Czas propagacji t_{prop} [ns] przy $C_L = 50$ pF	9	5	3,5	125	8	5,2	3
Maksymalna częstotliwość pracy f_{max} [MHz]	33	50	150	4	50	115	160
Prąd wyjściowy $I_{OH max}$ [mA] przy $U_{OH min}$	0,4	0,4	1	2,1 przy 2,5V	6 przy 4,5V	8 przy 4,5V	24 przy 3,8V
Prąd wyjściowy $I_{OL max}$ [mA] przy $U_{OL max}$	-8	-8	-20	-0,44 przy 0,4V	-6 przy 0,4V	-8 przy 0,4V	-24 przy 0,4V
Prąd wejściowy $I_{IH max}$ [μ A]	20	20	20	0,1	1	1	1
Prąd wejściowy $I_{IL max}$ [μ A]	400	200	600	0,1	1	1	1
Margines zakłóceń M_{min} [V]	0,3	0,4	0,3	$0,3 U_{CC}$	0,28 U_{CC} : 1,25V przy $U_{CC}=4,5V$ oraz 1,4V przy $U_{CC}=5,0V$		

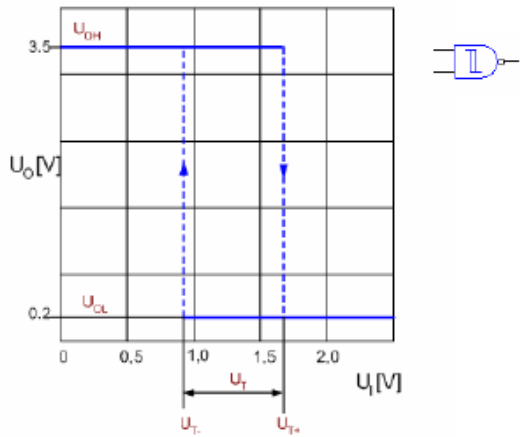
1. Charakterystyki przejściowe bramek



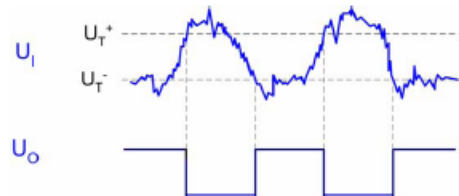
Rys.1 Charakterystyka przejściowa bramki NAND (TTL 74S00)



Rys.2 Charakterystyki przejściowe układów TTL i CMOS

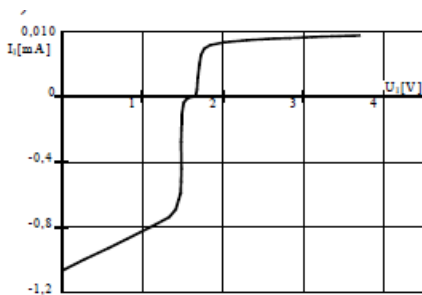


Rys.3 Charakterystyka przejściowa bramki NAND Schmitta

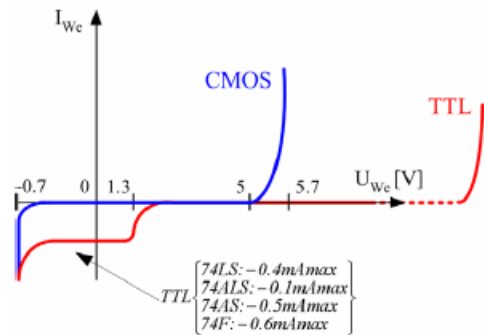


Rys.4 Zakłócenia na wejściu bramki z układem Schmitta oraz sygnał na jej wyjściu

2. Charakterystyki wejściowe bramek



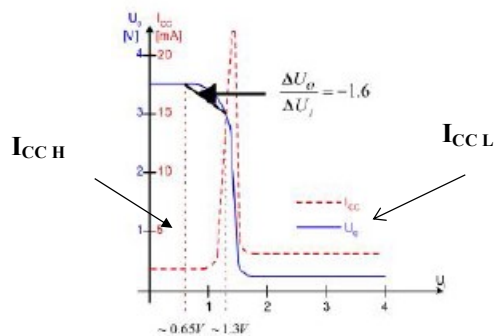
Rys.5 Typowa charakterystyka wejściowa bramki TTL



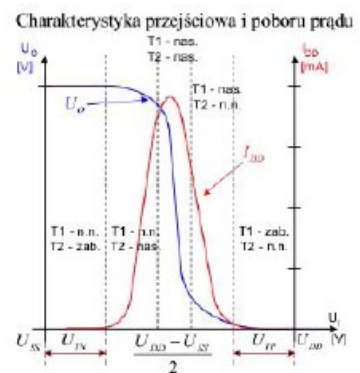
Rys.6 Porównanie charakterystyk wejściowych układów CMOS oraz TTL

3. Charakterystyki poboru prądu przez bramkę

a) Charakterystyka przejściowa i poboru prądu

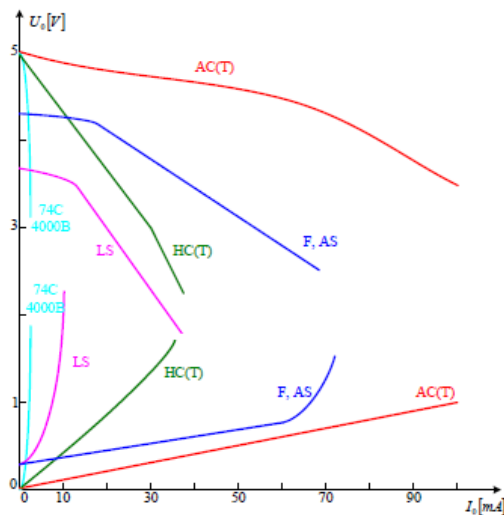


b)



Rys.7 Charakterystyki poboru prądu przez bramkę: a) TTL – NAND; b) CMOS

4. Charakterystyki wyjściowe



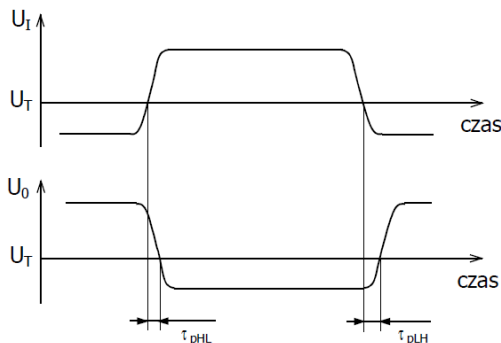
Na podstawie charakterystyk wyjściowych określamy maksymalne prądy obciążenia: $I_{OH\ max}$ i $I_{OL\ max}$

Rys.8 Charakterystyki wyjściowe rodziny TTL i CMOS w stanie L oraz H na wyjściu bramki

5. Czas propagacji

Czas propagacji jest to czas upływający od chwili zmiany stanu wejścia układu logicznego lub elementu logicznego do chwili ustalenia stanu wyjść, będącej reakcją na tę zmianę wejścia. Czas propagacji jest podstawowym parametrem charakteryzującym szybkość działania elementów i układów logicznych.

Czas propagacji układu określa się zazwyczaj w warunkach obciążenia jego wyjścia rzeczywistym lub symulowanym wejściem układu tego samego typu. Np. dla techniki CMOS przy pomiarach wykorzystuje się obciążenie pojemnością kilkudziesięciu pF.



Rys.9 Definicja czasów propagacji

W zależności od charakteru zmiany stanu wyjścia wyróżnia się dwa czasy propagacji:

- dla przejścia ze stanu niskiego do wysokiego (t_{pLH})
- dla przejścia ze stanu wysokiego do niskiego (t_{pHL})

Odstęp czasowy na zboczach sygnałów mierzy się dla określonej wartości amplitudy napięcia wejściowego (tzw. napięcia przełączania, U_T) i wyznacza średnią arytmetyczną

$$t_p = \frac{t_{pLH} + t_{pHL}}{2}$$

Dla układów TTL wejściowe napięcie przełączania to 1,4 V, zaś dla CMOS jest to $U_{CC}/2$. W praktyce, pomiaru dokonujemy nakładając przebiegi „na siebie” i mierząc czas opóźnienia w połowie amplitudy sygnałów wejściowego i wyjściowego

6. Margines zakłóceń (odporność na zakłócenia)

Zakłócenia mogą spowodować krótkotrwałe zmiany stanu logicznego na wyjściu układu. Miarą odporności układu na zakłócenia są marginesy zakłóceń.

Margines zakłóceń – określa dopuszczalną wartość amplitudy sygnału zakłócającego, która nie powoduje jeszcze nieprawidłowej pracy układu. Definiuje się dwa marginesy zakłóceń:

- margines zakłóceń stanu niskiego $M_L = U_{IL\ max} - U_{OL\ max}$
- margines zakłóceń stanu wysokiego $M_H = U_{OH\ min} - U_{IH\ min}$

Minimalne gwarantowane wartości dla układów TTL to $M_{H\ min} = M_{L\ min} = 0,4\ V$. W praktyce są większe: $M_{H\ min} = 2,1\ V$, zaś $M_{L\ min} = 1,2\ V$.

Testowanie i diagnostyka cyfrowych układów scalonych

Testowanie dużych układów cyfrowych jest poważnym i złożonym problemem technicznym. Celem testowania jest stwierdzenie, czy układ działa poprawnie, czy jest uszkodzony. Nie jest to tożsame z **diagnostyką** układu, polegającą na ustaleniu przyczyny uszkodzenia i ewentualnej lokalizacji defektu. Zbadanie, czy układ zawierający dziesiątki milionów tranzystorów wykonuje poprawnie swe zwykle bardzo złożone funkcje dla wszystkich możliwych stanów i sekwencji sygnałów na wejściach jest nie do wykonania w rozsądnie krótkim czasie. Badanie sprawności układu wymaga więc użycia specjalnych metod. Już na etapie projektu architektury i projektu logicznego układu projektant powinien przewidzieć, w jaki sposób układ będzie testowany. W systemie elektronicznym defekty mogą powstawać zarówno na etapie produkcyjnym jak i podczas użytkowania sprzętu. Defekty typowe dla układów scalonych o wysokiej skali integracji VLSI to:

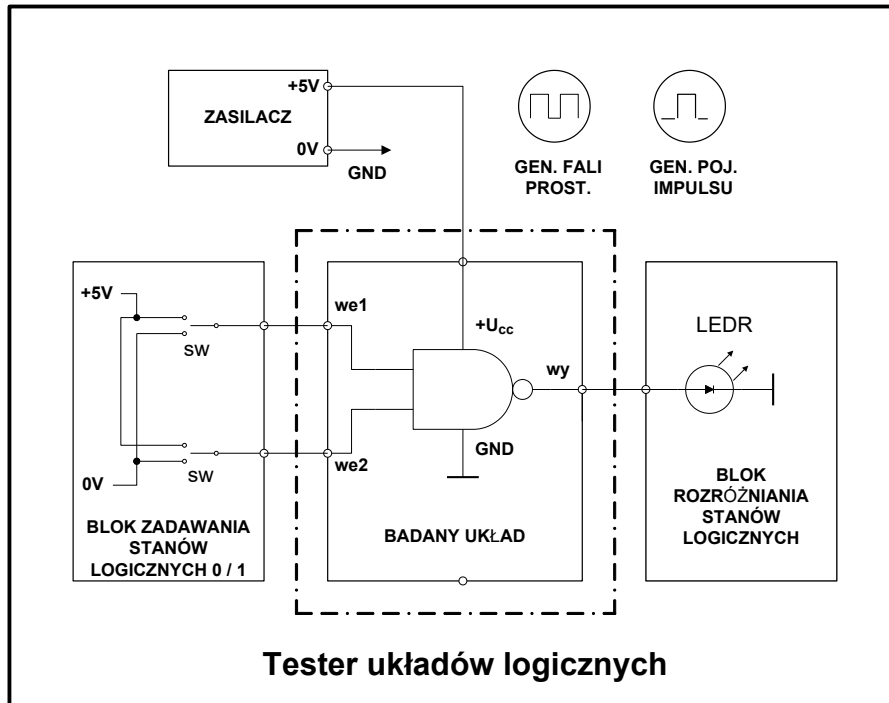
- defekty powstałe podczas procesu produkcyjnego, np. brak kontaktu, zwarcia w tlenku brankowym, zwarcia i rozwarca między ścieżkami;
- defekty materiałowe oraz związane z procesem starzenia;
- defekty powstałe na skutek oddziaływania środowiska np. defekty związane z temperaturą, wysoką wilgotnością, wibracjami, promieniowaniem

Testowanie układu będzie polegać na:

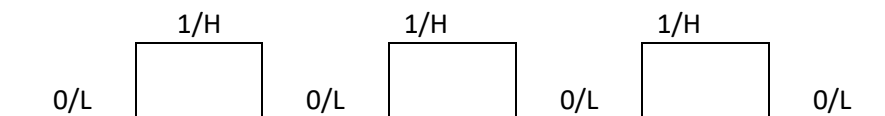
- obserwowaniu poziomów logicznych w danych węzłach układu,
- pomiarze statycznego poboru prądu
- badaniu zmian opóźnień w układzie
- badaniu zmian parametrów specyficznych dla danego typu układów.

TESTER SCALONYCH UKŁADÓW LOGICZNYCH (CYFROWYCH) wyposażony jest w:

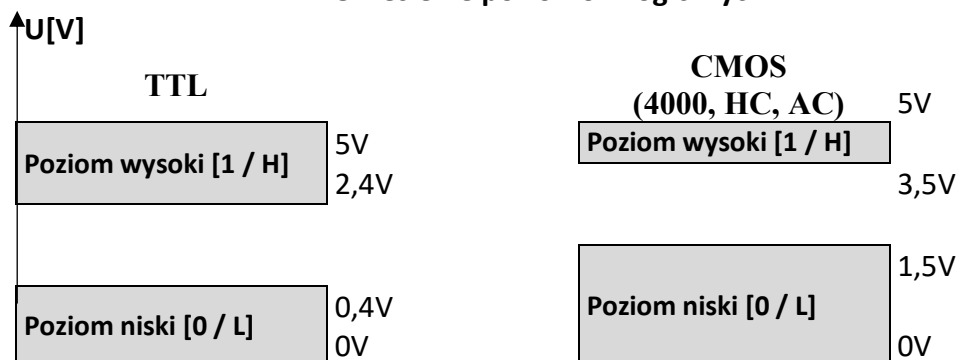
- zasilacz stabilizowany – +5V,
- blok zadawania stanów logicznych 0 [L] / 1 [H],
- blok rozróżniania stanów logicznych (wskaźniki stanów logicznych – diody LEDR),
- generator przebiegu prostokątnego,
- generator pojedynczego impulsu,



Sygnał cyfrowy



Określenie poziomów logicznych



4. Literatura

- Głocki W.: Układy cyfrowe, WSiP – 1998,
- Głocki W., Grabowski L.: Pracownia podstaw techniki cyfrowej, WSiP – 1998.

Imię i nazwisko:	Klasa:	Stanowisko:	Nr w dzienniku:
Skład grupy:.....			
Temat ćwiczenia: Badanie bramki logicznej NAND, NOR oraz bramek specjalnych Testowanie i diagnostyka cyfrowych układów scalonych		Data:	
Przygotowanie do ćwiczenia	Wykonanie ćwiczenia	Sprawozdanie z ćwiczenia	
Pkt.:	Pkt.:	Pkt.:	
Suma punktów:			
Ocena z przeprowadzonego ćwiczenia:		Podpis nauczyciela:	

Przypomniano o stosowaniu zasad bezpiecznej pracy i przestrzeganiu instrukcji BHiP.

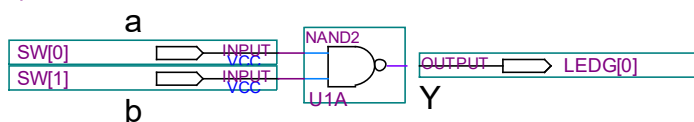
Zestaw przyrządów:.....

ZADANIA POMIAROWE

1. Mając do dyspozycji tylko dwuwejściowe bramki **NAND** zaprojektować układy realizujące następujące funkcje logiczne: **NOT, AND, OR, NOR.**

◆ **Typ badanego układu: bramka NAND**

Narysuj symbol graficzny badanej bramki oraz schemat układu pomiarowego.



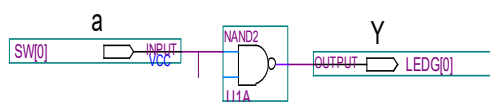
$$Y = \overline{a \cdot b}$$

Tabela prawdy

Wejścia		Wyjście
b	a	Y
0	0	
0	1	
1	0	
1	1	

◆ **Realizacja funkcji NOT**

Schemat funkcjonalny

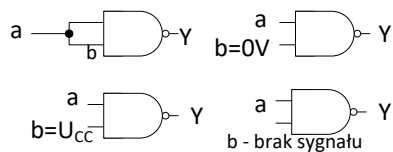


$$Y = \overline{a \cdot a} = \overline{a}$$

Tabela prawdy

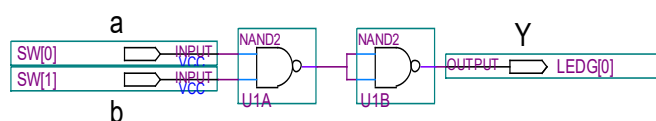
i	a	b=a	b=0V	b=U _{CC}	b – brak sygnału
		Y	Y	Y	Y
0	0				
1	1				

Sprawdź jak zachowuje się bramka NAND w następujących połączeniach:



◆ **Realizacja funkcji AND**

Schemat funkcjonalny



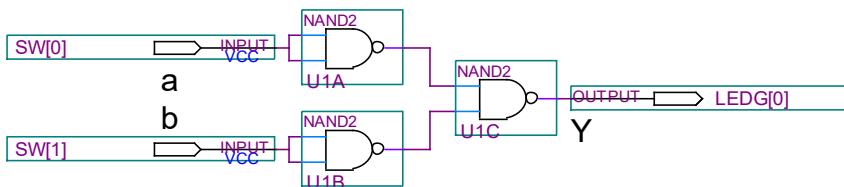
$$Y = \overline{\overline{a \cdot b}} = a \cdot b$$

Tabela prawdy

i	b	a	Y
0			
1			
2			
3			

◆ Realizacja funkcji OR

Schemat funkcjonalny



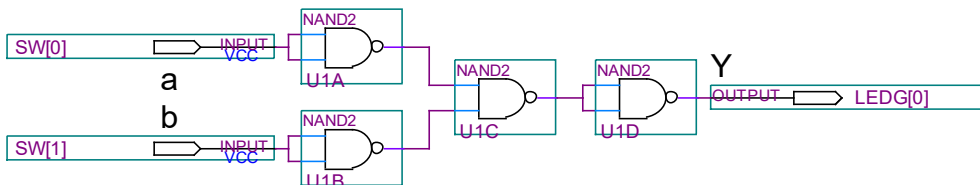
$$Y = \overline{\overline{a \cdot b}} = \overline{\overline{a + b}} = a + b$$

Tabela prawdy

<i>i</i>	b	a	Y
0			
1			
2			
3			

◆ Realizacja funkcji NOR

Schemat funkcjonalny



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

Tabela prawdy

<i>i</i>	b	a	Y
0			
1			
2			
3			

◆ Realizacja funkcji $Y = a \cdot b \cdot c$

Schemat funkcjonalny

Tabela prawdy

<i>i</i>	c	b	a	Y
0				
1				
2				
3				
4				
5				
6				
7				

2. Badanie bramek z otwartym kolektorem

- ◆ Sprawdzić działanie układu przy niewłaściwie dobranych rezystorach.
- ◆ Sprawdzić działanie układu oraz zrealizować zadane przez prowadzącego funkcje logiczne.

Schemat funkcjonalny

Tabela prawdy

ZADANIA POMIAROWE

1. Mając do dyspozycji tylko dwuwejściowe bramki **NOR** zaprojektować układy realizujące następujące funkcje logiczne: **NOT, OR, AND, NAND**.

◆ **Typ badanego układu: bramka NOR**

Narysuj symbol graficzny badanej bramki oraz schemat układu pomiarowego.

Tabela prawdy

Wejścia		Wyjście
b	a	Y
0	0	
0	1	
1	0	
1	1	

Y =

◆ **Realizacja funkcji NOT**

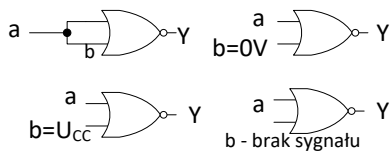
Schemat funkcjonalny

Tabela prawdy

i	a	b=a	b=0V	b=U _{CC}	b – brak sygnału
		Y	Y	Y	Y
0	0				
1	1				

Y =

Sprawdź jak zachowuje się bramka NOR w następujących połączeniach:



◆ **Realizacja funkcji OR**

Schemat funkcjonalny

Tabela prawdy

i	b	a	Y
0			
1			
2			
3			

Y =

◆ **Realizacja funkcji AND**

Schemat funkcjonalny

Tabela prawdy

i	b	a	Y
0			
1			
2			
3			

Y =

◆ **Realizacja funkcji NAND**

Schemat funkcjonalny

Tabela prawdy

i	b	a	Y
0			
1			
2			
3			

Y =

◆ Realizacja funkcji $Y = a+b+c$

Schemat funkcjonalny

Tabela prawdy

i	c	b	a	Y
0				
1				
2				
3				
4				
5				
6				
7				

2. Zaprojektuj układy realizujące funkcję EXOR i EXNOR tylko na bramkach NAND (NOR)

NAND

EXOR Y=
EXNOR Y=

NOR

3. Wnioski

1. Właściwości bramki NAND (NOR)

2. Właściwości bramek z otwartym kolektorem

3. Jakie znasz jeszcze bramki, których wyjścia można łączyć ze sobą. Narysuj symbol takiej bramki oraz napisz jej tabelę stanów?

4. Co należy zrobić z niewykorzystanymi wejściami bramek?

5. Co oznaczają symbole: H, L, S, LS, F, AS, ALS występujące w oznaczeniu bramki TTL, np.: UCY74S00N i czym tak oznaczone bramki różnią się od bramki standardowej?

6. Co oznaczają symbole AC, ACT, HC, HCT, C – występujące w oznaczeniach układów scalonych CMOS np.: 74ACT00?

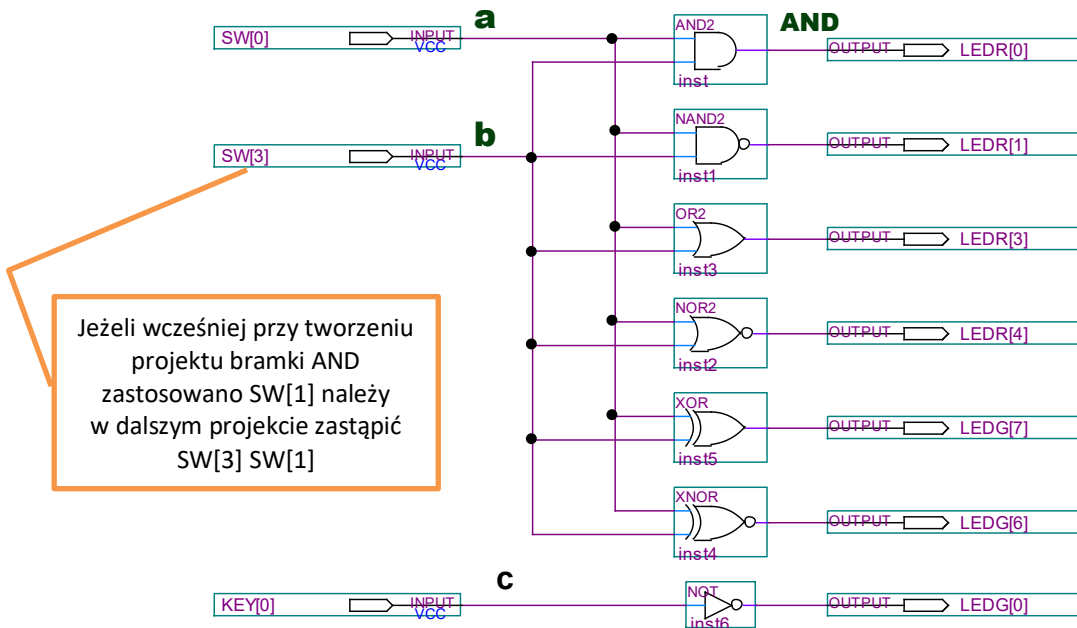
7. Czym będą różniły się bramki oznaczone: 7400, 74C00, 74AC00, 74ACT00?

Wprowadzenie do programowania układów FPGA (ALTERA) z wykorzystaniem programu QuartusII

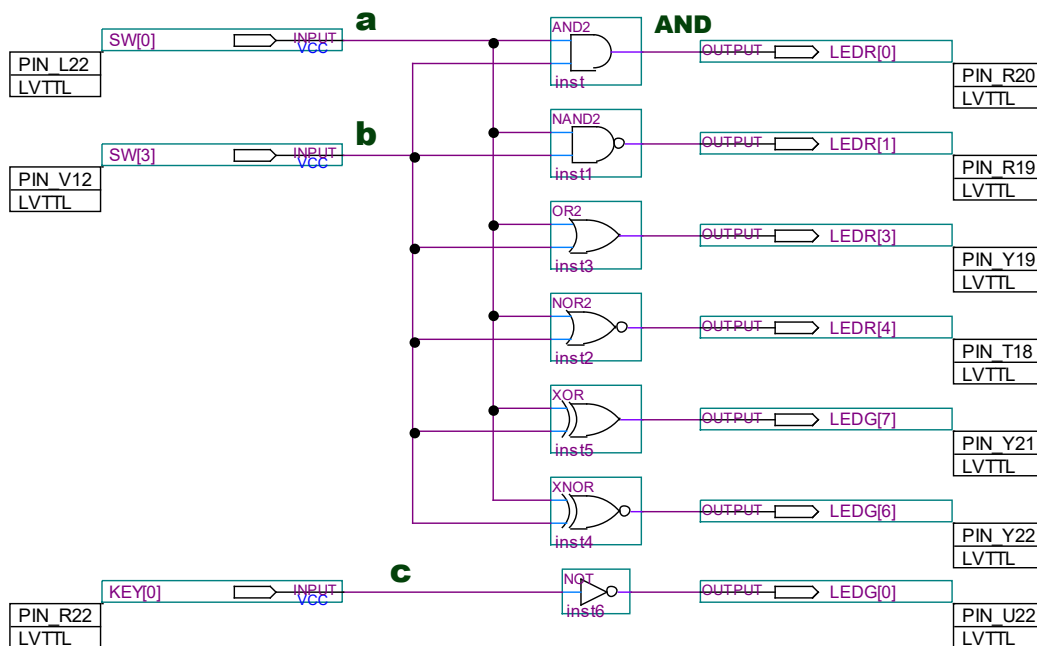
Zaimplementuj w układzie FPGA wszystkie bramki logiczne zgodnie z przedstawionym schematem. Sprawdź działanie zaimplementowanych bramek zadając stany logiczne na ich wejścia za pomocą przełączników SW[0] i SW[3], obserwując stan na ich wyjściach na odpowiednich diodach. Wyniki zapisz w tabeli.

BADANIE BRAMEK

1. Projekt układu



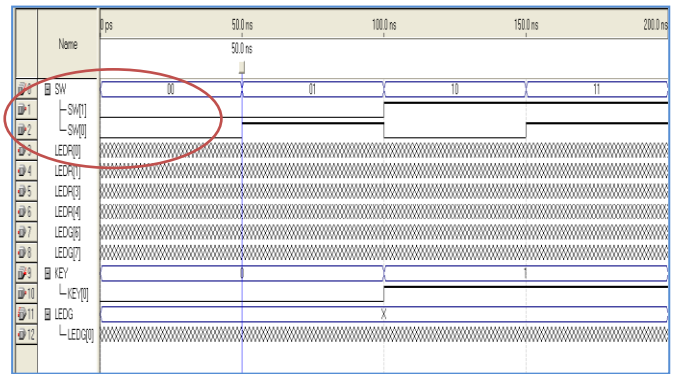
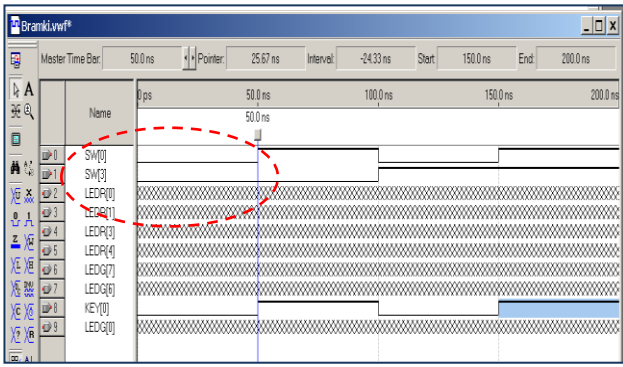
2. Po przyporządkowaniu pinów ALTERY



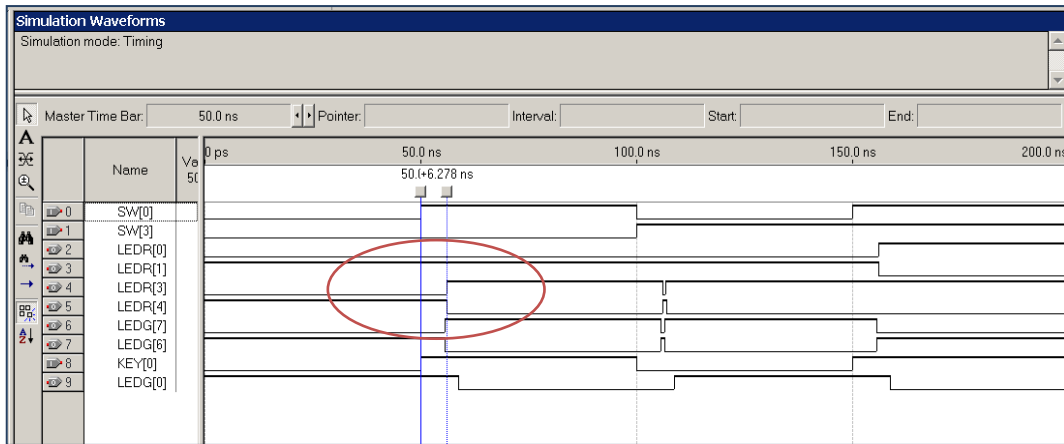
3. Symulacja pracy układu:

End Time = 200ns,

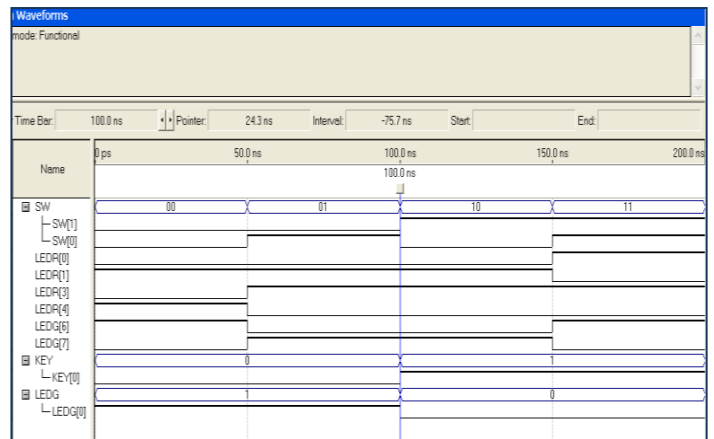
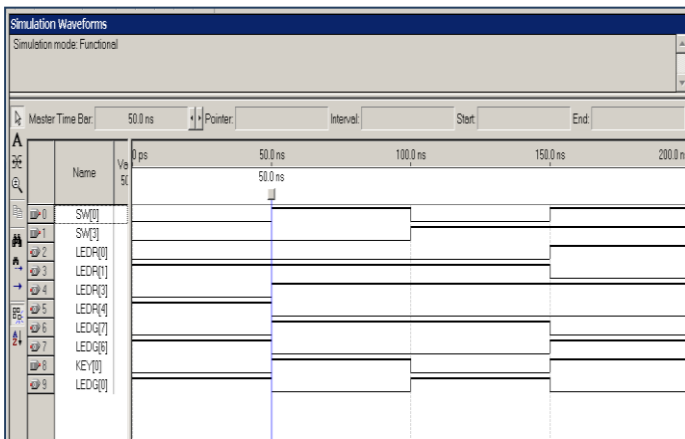
Grid size = 50ns



- Odpowiedź układu – symulacja czasowa (uwzględnia czasy propagacji układu)
 - Odczytaj czas propagacji bramki: $t_p =$



- Odpowiedź układu – symulacja funkcjonalna

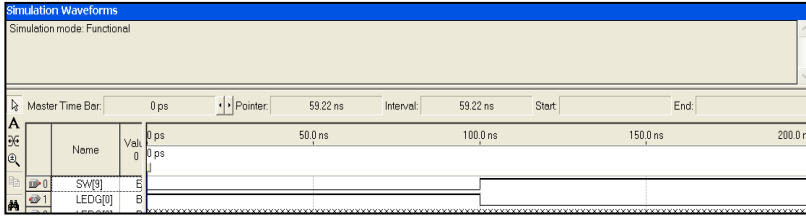
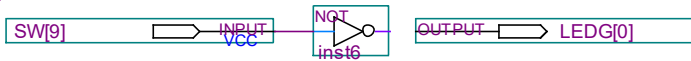


Wyniki badania bramek umieść w tabeli:

i	b SW[3]	a SW[0]	AND LEDR[0]	NAND LEDR[1]	OR LEDR[3]	NOR LEDR[4]	EXOR LEDG[7]	EXNOR LEDG[6]	c KEY[0]	NOT LEDG[0]
0	0	0							0	
1	0	1							1	
2	1	0							0	
3	1	1							1	

BADANIE BRAMEK TRÓJSTANOWYCH

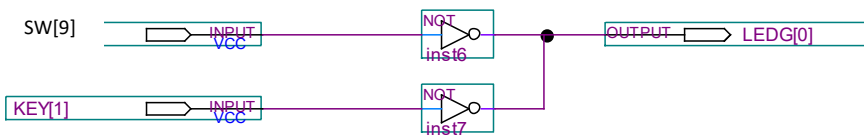
1. W programie Quartus II utwórz nowy projekt o nazwie *nr1_nr_2_kl_bramki_spec*
2. Badanie bramki **NOT standardowej**



Bramka działa –

SW[9]	LEDG[0]
0	
1	

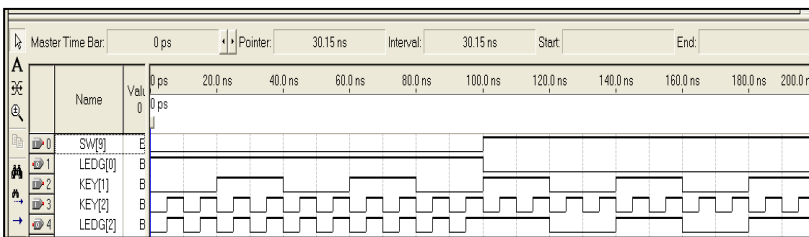
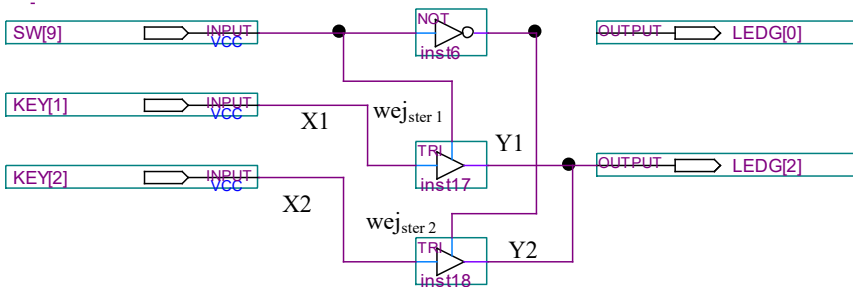
3. Dołącz drugą bramkę jak na rysunku:



- Skompiluj projekt
- Jaki jest wynik kompilacji?
- Czy wyjścia bramek standardowych można łączyć ze sobą?

4. Badanie **bramek trójstanowych**

- Popraw schemat, uzupełniając go o bramki trójstanowe:



- Uzupełnij tabelę:

wej _{ster1} SW[9]	wej _{ster2} SW[9]	X1	X2	Y1	Y2	LEDG[2]
0	1	KEY[1]	KEY[2]			KEY[2]
1	0	KEY[1]	KEY[2]			KEY[1]

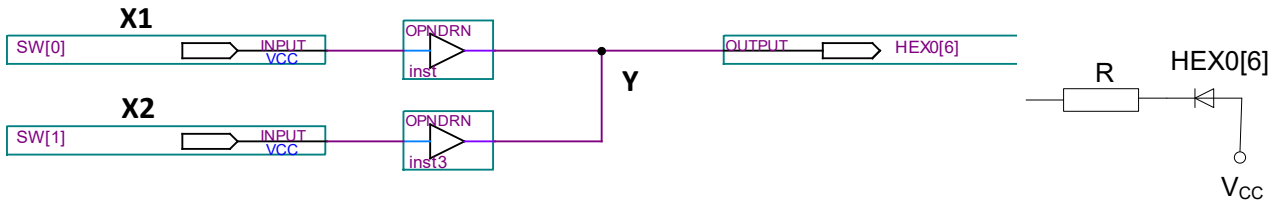
Tabela prawdy bramki trójstanowej

wej _{ster}	wej X	wyj Y
0	0/1	
1	0/1	

Wnioski:

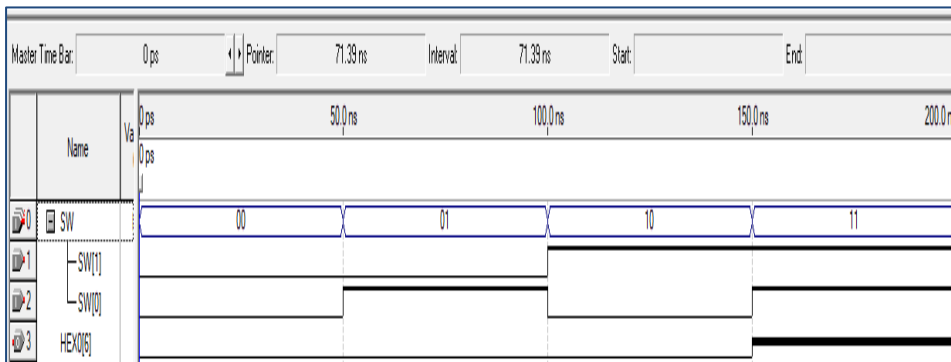
- Wyjścia bramek trójstanowych
- Jeżeli stan wejścia sterującego bramki trójstanowej jest w stanie **0** to wyjście bramki jest w, jeżeli na wejściu sterującym jest stan **1** to na wyjściu

5. Badanie bramek z **otwartym drenem (open dren – OPNDRN)**



Uwaga! Wyświetlacz w zestawie DE1 jest wyświetlaczem ze wspólną anodą. Segment wyświetlacza świeci się jeżeli na wyjściu układu Y pojawi się stan niski [0]. Jeżeli segment jest wygaszony wyjście Y jest w stanie wysokim [1].

- Wyjścia bramek z otwartym drenem (kolektorem) można
- Realizowana jest wówczas tak zwana "suma montażowa"
- Wyniki badań UKŁADU umieść w tabeli



SW[1] X1	SW[2] X2	HEX0[6] Y
0	0	
0	1	
1	0	
1	1	
Y =		

Imię i nazwisko:	Klasa:	Stanowisko:	Nr w dzienniku:
Skład grupy:			
Temat ćwiczenia: Synteza układów kombinacyjnych (minimalizacja funkcji logicznych metodą tablic Karnaugh)			Data:
Przygotowanie do ćwiczenia	Wykonanie ćwiczenia	Sprawozdanie z ćwiczenia	
Pkt.:	Pkt.:	Pkt.:	
Suma punktów:			
Ocena z przeprowadzonego ćwiczenia:			Podpis nauczyciela:

Przypomniano o stosowaniu zasad bezpiecznej pracy i przestrzeganiu instrukcji BHiP.

1. Cel i zakres ćwiczenia

Celem ćwiczenia jest praktyczna weryfikacja graficznej metody minimalizacji funkcji logicznych, połączenie i uruchomienie prostego układu kombinacyjnego.

2. Wstęp teoretyczny

- ◆ Wymień sposoby opisu układów kombinacyjnych:

.....

.....

- ◆ Czy charakteryzuje się postać kanoniczna sumacyjna (iloczynowa) funkcji logicznej

.....

.....

- ◆ Na czym polega minimalizacja funkcji logicznych metodą tablic Karnaugh,

.....

.....

Materiały: [A_Boolea_Min_funkcji_log_Prezentacja](http://zsl.gda.pl/~ihoja/Pracownia_Eksploatacji_Urzadzen_Elektronicznych/uklady_kombinacyjne) ([http://zsl.gda.pl/~ihoja/Pracownia Eksploatacji Urządzeń Elektronicznych/układy kombinacyjne](http://zsl.gda.pl/~ihoja/Pracownia_Eksploatacji_Urzadzen_Elektronicznych/uklady_kombinacyjne))

- ◆ Narysować schemat logiczny układu opisanego powyższą funkcją korzystając z bramek **NOT, AND i OR**.

- ◆ Wpisać funkcję opisującą projektowany układ do tablicy **Karnaugh**.

$X_1 X_0$	00	01	11	10
$X_3 X_2$				
00				
01				
11				
10				

$X_2 X_1 X_0$	000	001	011	010	110	111	101	100
$X_4 X_3$								
00								
01								
11								
10								

- ◆ Przeprowadzić minimalizację funkcji metodą grupowania **1**. Przedstawić zminimalizowaną postać sumacyjną funkcji.

.....

.....

- ◆ Narysować schemat logiczny układu opisanego zminimalizowaną funkcją sumacyjną korzystając z bramek **NOT, AND i OR**.

- ◆ Narysować i zrealizować projektowany układ stosując tylko bramki **NAND**.

- ◆ Działanie układu zaprezentować nauczycielowi

2. Wnioski

Imię i nazwisko:	Klasa:	Stanowisko:	Nr w dzienniku:
Skład grupy:			
Temat ćwiczenia: Badanie multiplekserów i demultiplekserów			Data:
Przygotowanie do ćwiczenia	Wykonanie ćwiczenia	Sprawozdanie z ćwiczenia	
Pkt.:	Pkt.:	Pkt.:	
Suma punktów:			
Ocena z przeprowadzonego ćwiczenia:			Podpis nauczyciela:

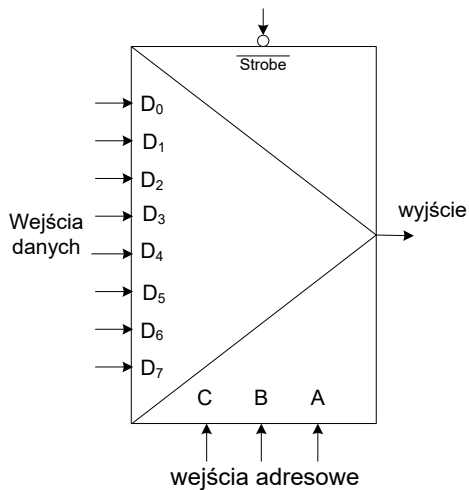
Przypomniano o stosowaniu zasad bezpiecznej pracy i przestrzeganiu instrukcji BHiP.

Aparatura:.....

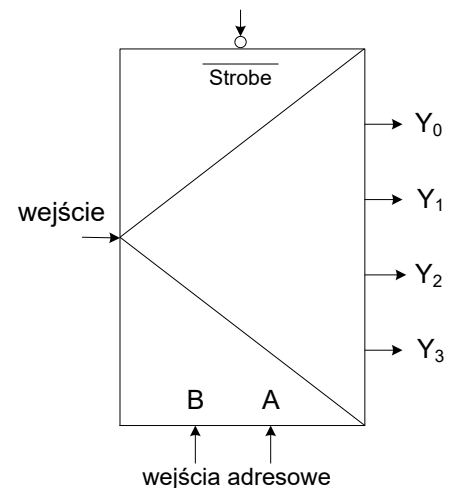
1. Cel i zakres ćwiczenia

Celem ćwiczenia jest poznanie działania multipleksera i demultipleksera oraz nabycie umiejętności analizowania ich pracy i wykorzystania w systemach cyfrowych.

2. Wstęp teoretyczny



Multiplekserem nazywamy układ służący do przesłania na wyjście jednego z wielu sygnałów doprowadzonych do jego wejść danych. Wybór wejścia odbywa się za pośrednictwem wejść adresowych.



DEMUPLEKSER jest układem służącym do przesłania na jedno z wielu wyjść układu sygnału doprowadzonego do jego wejścia. Wybór wyjścia odbywa się za pośrednictwem wejść adresowych.

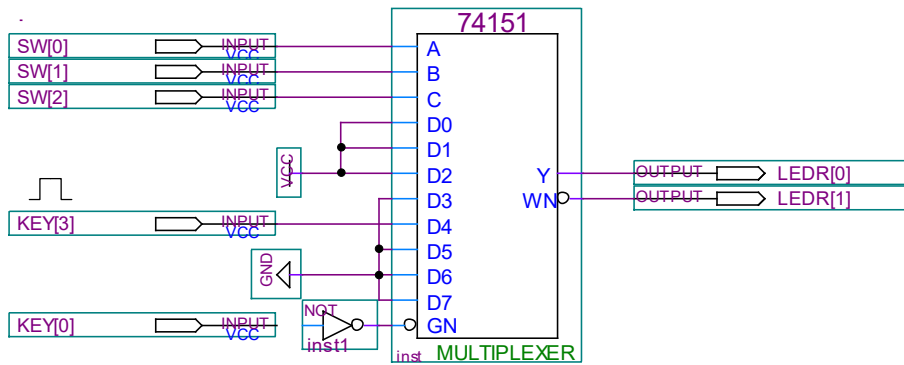
MULTIPLEKSERY I DEMULTIPLEKSERY stosujemy:

- w multipleksowym systemie przesyłania danych,
- jako układy przetwarzające postać informacji z równoległej na szeregową (MUX) oraz z szeregową na równoległą (DMUX),
- do realizacji układów kombinacyjnych. **Zadania pomiarowe**

3.1. Badanie multipleksera

3.1.1. Określenie tablicy prawdy multipleksera

- Zaimplementować w układzie FPGA zestawu DE_1 multiplekser o 3 - wejściach adresowych np. '151 zgodnie z przedstawionym schematem.
- Doprowadzając do wejść adresowych, wejść danych i wejścia strobojujące różne kombinacje stanów "0" i "1" określić działanie układu, zapisując wyniki obserwacji w tablicy.
- Przedstawić również działanie multipleksera na diagramie czasowym.

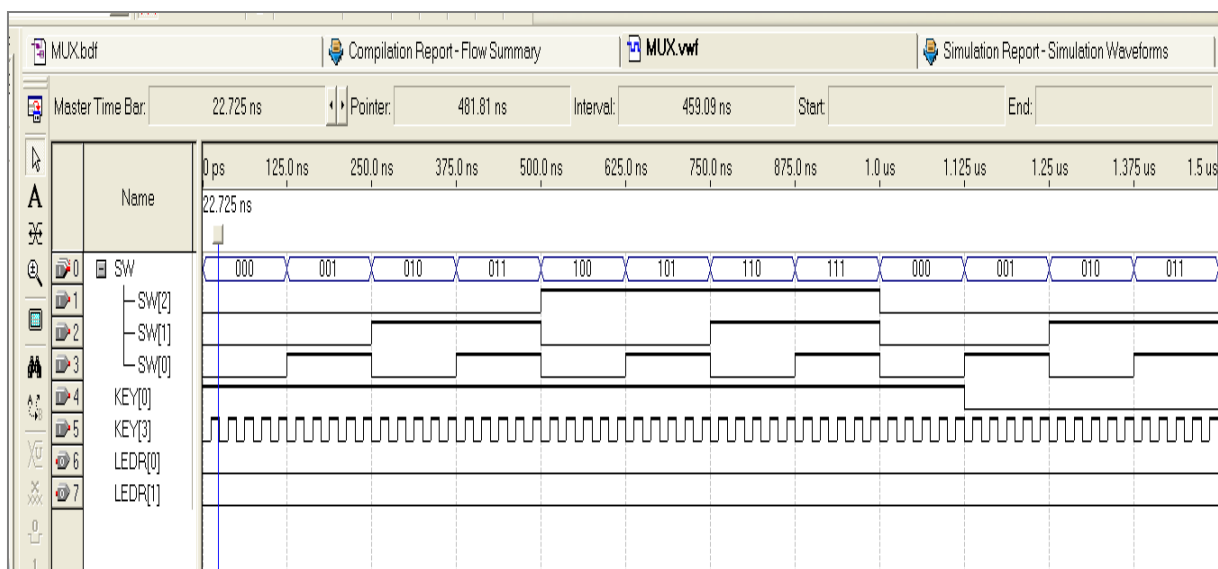
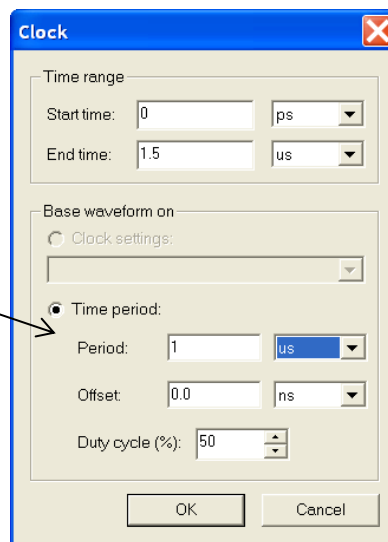


Tablica prawdy multiplexera 151'

adres DEC	wejścia												wyjścia		
	adresowe			strob.											
	C	B	A	GN (\bar{s})	D0	D1	D2	D3	D4	D5	D6	D7	Y	WN	
X	X	X	X	1	X	X	X	X	X	X	X	X	0	1	
0	0	0	0	0											
1	0	0	1	0											
2	0	1	0	0											
3	0	1	1	0											
4	1	0	0	0											
5	1	0	1	0											
6	1	1	0	0											
7	1	1	1	0											

Ustawienia parametrów symulacji

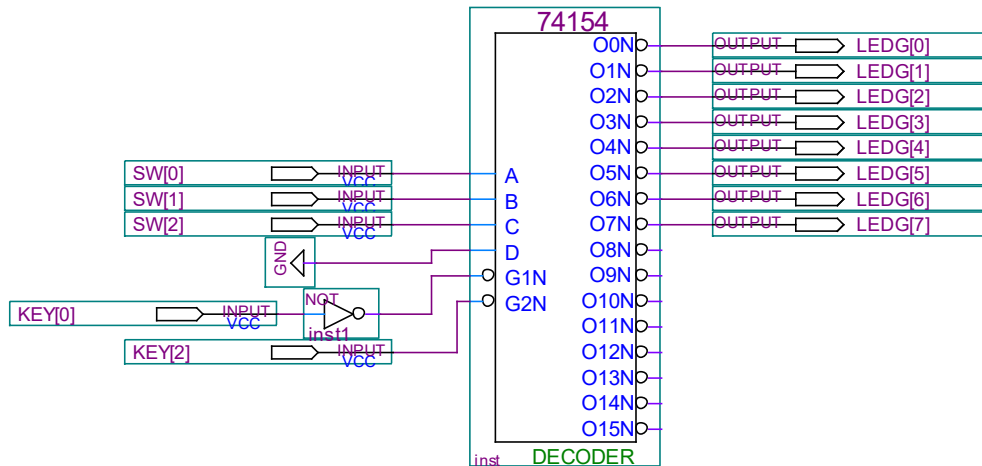
- End time = 1,5 μ s
- Grid size = 125ns
- Wejścia adresowe:
 - ✓ C = SW[2] = 1 μ s
 - ✓ B = SW[1] = 500 ns
 - ✓ A = SW[0] = 250 ns
- Wejście danych:
 - ✓ KEY[3] = 25 ns



3.2. Badanie demultipleksera

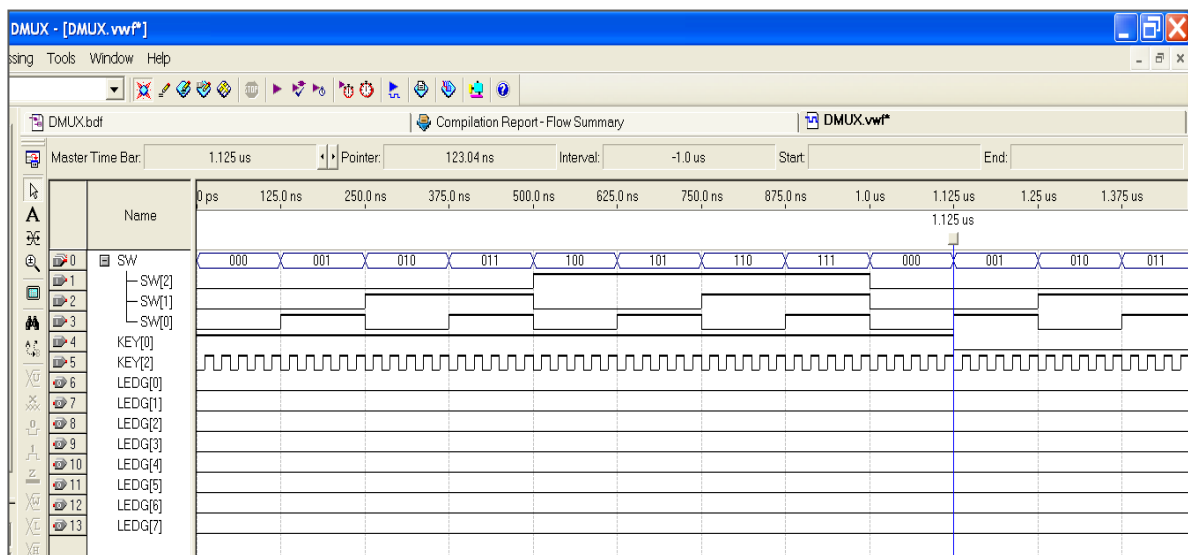
3.2.1. Określenie tablicy prawdy demultipleksera

- Zaimplementuj w układzie FPGA zestawu DE_1 demultipleksler o 4 - wejściach adresowych np. '154 zgodnie z przedstawionym schematem.
- **W czasie pomiarów ogranicz się do 3 – wejść adresowych.**
- Doprowadzając do wejść adresowych, wejścia danych i wejścia strobingowego różne kombinacje stanów "0" i "1" określić działanie układu, zapisując wyniki obserwacji w tablicy.
- Przedstawić również działanie demultipleksera na diagramie czasowym.



Tablica prawdy multipleksera '154

adres DEC	wejścia					wyjścia								
	adresowe			strob.		O0	O1	O2	O3	O4	O5	O6	O7
	C	B	A	G1N	G2N									
X	X	X	X	1	X									1
X	X	X	X	X	1									
0	0	0	0	0	⌋									
1	0	0	1	0										
2	0	1	0	0										
3	0	1	1	0										
4	1	0	0	0										
5	1	0	1	0										
6	1	1	0	0										
7	1	1	1	0										
.....	0										



3.3. Multipleksowanie danych

- * Mając do dyspozycji multiplekser i demultiplekser zaprojektować układ multipleksowania danych..*

4. Wnioski

5. LITERATURA

1. W. Głocki .:Układy cyfrowe. Warszawa, WSiP 1998
2. W. Głocki, L. Grabowski .:Pracownia podstaw techniki cyfrowej, WSiP 1998