

# Elektronika Cyfrowa

## Sprawozdanie z Laboratorium 4

Tomasz Dziób

Grupa 15

24.04.2024

## Spis treści

<b>1 Wstęp teoretyczny</b>	<b>2</b>
1.1 Algebra Boole'a . . . . .	2
1.2 Układy cyfrowe . . . . .	2
1.2.1 Układy kombinacyjne . . . . .	2
1.2.2 Układy sekwencyjne . . . . .	2
1.2.2.1 Asynchroniczne . . . . .	2
1.2.2.2 Synchroniczne . . . . .	2
1.3 Mapy Karnaugh'a . . . . .	3
1.4 Bramki logiczne . . . . .	3
1.4.1 Podstawowe operacje zbudowane z bramek . . . . .	4
1.4.2 Układy TTL . . . . .	4
1.4.2.1 Charakterystyka układów TTL . . . . .	4
1.5 Asynchroniczny przerzutnik RS . . . . .	4
<b>2 Ćwiczenia</b>	<b>5</b>
2.1 Ćwiczenie 4.1 . . . . .	5
2.1.1 Sprawdzenie poprawności działania impulsatorów . . . . .	6
2.1.2 Sprawdzenie poprawności działania diod próbnika . . . . .	7
2.2 Ćwiczenie 4.2 . . . . .	8
2.2.1 NAND(7400) . . . . .	8
2.2.2 NOR(7402) . . . . .	9
2.2.3 EXOR(7486) . . . . .	9
2.3 Ćwiczenie 4.3 . . . . .	10
2.3.1 Realizacja przy użyciu bramek NAND . . . . .	10
2.3.2 Realizacja przy użyciu bramek NOR . . . . .	12
2.4 Ćwiczenie 4.4 . . . . .	12
2.4.1 7400 . . . . .	13
2.4.2 74S00 . . . . .	13
2.5 Ćwiczenie 4.5 . . . . .	13
2.6 Ćwiczenie 4.6 . . . . .	15
<b>3 Omówienie wyników</b>	<b>16</b>
3.1 Ćwiczenie 4.1 . . . . .	16
3.2 Ćwiczenie 4.2 . . . . .	16
3.3 Ćwiczenie 4.3 . . . . .	16
3.4 Ćwiczenie 4.4 . . . . .	16
3.5 Ćwiczenie 4.5 . . . . .	16
3.6 Ćwiczenie 4.6 . . . . .	16
<b>4 Podsumowanie</b>	<b>16</b>
<b>5 Notatki z zajęć</b>	<b>16</b>

# 1 Wstęp teoretyczny

Poniższe sprawozdanie dotyczy dotyczy czwartych zajęć których głównym celem było zapoznanie się z podstawami elektroniki cyfrowej czyli bramkami logicznymi.

## 1.1 Algebra Boole'a

Algebra Boole'a jest strukturą matematyczną gdzie dany jest:

- zbiór dwuelementowy
- działania alternatywy (sumy)
- koniunkcji (iloczynu)
- negacji (dopełnienia)

Dla uproszczenia zapisu koniunkcję  $p \wedge q$  zapisujemy jako  $p \cdot q$  lub  $pq$ , a negację jako  $\neg p$  lub  $\bar{p}$ . Poniżej prezentuje jedno z ważniejszych praw algebry Boole'a:

Prawo	Iloczyn	Alternatywa
<b>Prawo rozdzielnosci</b>	$A(B + C) = AB + AC$	$A + (BC) = (A + B)(A + C)$
<b>Prawo tożsamości</b>	$1 \cdot A = A$	$0 + A = A$
<b>Prawo odwrotności</b>	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$
—	$0 \cdot A = 0$	$1 + A = 1$
—	$A \cdot A = A$	$A + A = A$
<b>Prawa De Morgana</b>	$\overline{(A \cdot B)} = \bar{A} + \bar{B}$	$\overline{(A + B)} = \bar{A} \cdot \bar{B}$

Tablica 1: Podstawowe prawa działania algebry Boole'a

## 1.2 Układy cyfrowe

Układ cyfrowy, znany również jako układ logiczny, to układ przeprowadzający działania na danych wejściowych i wypuszczający je na kanały wyjściowe. Układy cyfrowe są budowane w oparciu o bramki logiczne, które realizują elementarne operacje logiczne. Układy cyfrowe są zaprojektowane do realizacji takich zadań jak przetwarzanie informacji (w tym obliczenia) oraz sterowanie urządzeniami i innymi systemami lub obiektami.

Układy cyfrowe możemy podzielić na układy kombinacyjne oraz sekwencyjne.

### 1.2.1 Układy kombinacyjne

W takich układach stan wyjść jest jednoznacznie określony przez stan wejść układu. W prostszych słowach, są to układy "bez pamięci", w których sygnały wyjściowe są zawsze takie same dla określonych sygnałów wejściowych.

### 1.2.2 Układy sekwencyjne

W układach sekwencyjnych stan wyjść zależy od stanu wejść oraz od poprzednich stanów układu. Czyli można powiedzieć że "pamiętają" one, stany wcześniejsze i wpływa to na następne wyniki.

W tym przypadku można wyróżnić jeszcze jeden podział: na układy sekwencyjne asynchroniczne i synchroniczne.

#### 1.2.2.1 Asynchroniczne

W układach asynchronicznych zmiana sygnałów wejściowych natychmiast powoduje zmianę wyjść. W związku z tym układy te są szybkie, ale jednocześnie podatne na zjawisko hazardu i wyścigu, czyli gdy co najmniej dwa sygnały wejściowe zmieniają swój stan w jednej chwili.

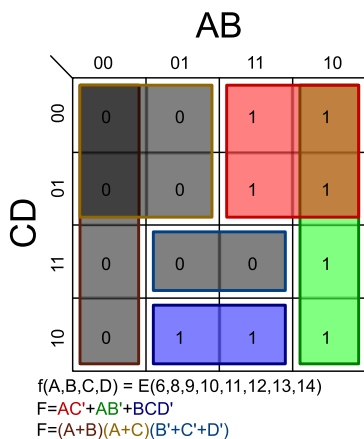
#### 1.2.2.2 Synchroniczne

W układach synchronicznych zmiana stanu wewnętrznego następuje wyłącznie w określonych chwilach, które wyznacza sygnał zegarowy. Każdy układ synchroniczny ma wejście zegarowe oznaczane zwyczajowo symbolami C, CLK lub CLOCK.

### 1.3 Mapy Karnaugh

Mapy Karnaugh to metoda upraszczania wyrażeń logicznych algebry Boole'a. Odzworowuje ona sygnały wejściowe układu w sygnały wyjściowe. Oznacza to, że część zmiennych binarnych przypisana jest wierszom, a część kolumnom. Indeksy i kolumny numerowane są przy pomocy kodu Graya. Grupowanie osobno jedynek lub zer pozwala uzyskać różne schematy budowy układu. Po zapisaniu równania, można je uprościć zasadami algebry Boole'a aby uzyskać najoptymalniejszą wersję układu.

Mapy Karnaugh są szeroko stosowane w kombinacyjnych układach logicznych w celu uproszczenia i zapobiegania potencjalnym warunkom wyścigu. Dzięki zastosowaniu kodu Graya możliwe jest znalezienie w wizualny sposób pól sąsiednich logicznie, czyli różniących się wartością dokładnie jednej zmiennej.



Rysunek 1: Przykładowa mapa Karnaugh,  
Źródło: [Wikipedia](#)

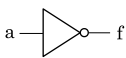
### 1.4 Bramki logiczne

Są to element konstrukcyjne mechanizmów realizujących funkcję logiczną, której argumenty (zmiennne logiczne) oraz sama funkcja mogą przybierać jedną z dwóch wartości, np. 0 lub 1 (jak w algebrze Boole'a).

Bramki te realizują głównie operacje logiczne takie jak negacja iloczyn oraz suma, jednak istnieją też ich zanegowane wersje jak i operacja wyłącznej sumy logicznej.


#### Bramki logiczne (rodzaj, funkcja logiczna, symbol, tablica prawdy)

NOT (negacja):
$$f = \bar{a}$$



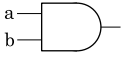
a	f
0	1
1	0

EXOR (Exclusive OR, wyłączna suma logiczna):
$$f = a \oplus b$$



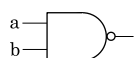
a	b	f
0	0	0
0	1	1
1	0	1
1	1	0

AND (iloczyn):
$$f = a \cdot b$$




a	b	f
0	0	0
0	1	0
1	0	0
1	1	1

NAND (NOT-AND, negacja iloczynu):
$$f = \overline{a \cdot b}$$



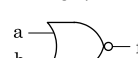
a	b	f
0	0	1
0	1	1
1	0	1
1	1	0

OR (suma):
$$f = a + b$$



a	b	f
0	0	0
0	1	1
1	0	1
1	1	1

NOR (NOT-OR, negacja sumy):
$$f = \overline{a + b}$$



a	b	f
0	0	1
0	1	0
1	0	0
1	1	0

Rysunek 2: Rodzaje bramek logicznych, ich symbole oraz tablice prawdy,  
Źródło: [Strona wykładów](#)

Tablica prawdy — to lista wszystkich możliwych stanów osiąganych przez bramkę dla wszystkich istniejących kombinacji danych wejściowych. Pokazuje ona w prosty i szybki sposób działanie bramki.

#### 1.4.1 Podstawowe operacje zbudowane z bramek

Bramki **NAND** (negacja koniunkcji) oraz **NOR** (negacja sumy logicznej) nazywa się funkcjonalnie pełnymi lub zupełnymi ponieważ z każdej z nich można zbudować układ realizujący dowolną funkcję logiczną.

Oto przykłady operacji logicznych zrealizowanych przy pomocy bramek NAND oraz NOR:

	Bramki NAND	Bramki NOR
Negacja		
Iloczyn		
Suma		

Rysunek 3: Schematy operacji logicznych zrealizowanych przy pomocy bramek NAND oraz NOR,  
Źródło: Opracowanie własne

#### 1.4.2 Układy TTL

Ze względu na stosowane technologie, bramki logiczne tworzą tzw. rodziny (np. TTL ECL, CMOS). Jedną z najbardziej rozpowszechnionych jest rodzina bramek TTL (Transistor - Transistor Logic). Ten typ bramek był również wykorzystywany na podczas zajęć.

##### 1.4.2.1 Charakterystyka układów TTL

- Układy pracują w logice dodatniej
- Logicznemu zeru (L — stan niski) odpowiada napięcie z przedziału:  
0 — 0.8 V (sygnały wejściowe)  
0 — 0.4 V (sygnały wyjściowe)
- Logicznej jedynce (H — stan wysoki) odpowiada napięcie z zakresu:  
2.0 – 5 V (sygnały wejściowe)  
2.7 — 5 V (sygnały wyjściowe)
- Wejście bramki niepodłączone do niczego znajduje się w stanie logicznym 1
- Układy zasilają się napięciem +5 V

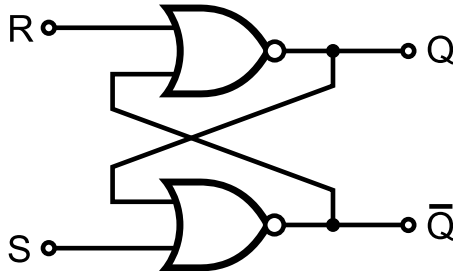
### 1.5 Asynchroniczny przerzutnik RS

Przerzutnik typu RS to najprostszy rodzaj przerzutnika asynchronicznego. Można go zbudować zarówno z dwóch bramek logicznych NOR lub dwóch bramek logicznych NAND. Przerzutnik RS ma dwa wejścia i dwa wyjścia:  
Wejścia:

**S** (Set) — wejście ustawiające  
**R** (Reset) — wejście zerujące

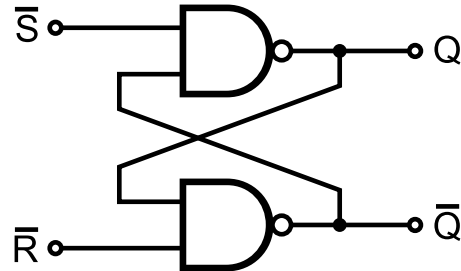
Wyjścia:

**Q** — wyjście zwykłe (główne)  
 **$\bar{Q}$**  — wyjście zanegowane (komplementarne)



Rysunek 4: Schemat realizacji przerzutnika RS z bramek NOR,

Źródło: [Wikipedia](#)



Rysunek 5: Schemat realizacji przerzutnika SR z bramek NAND,

Źródło: [Wikipedia](#)

Przerzutnik RS działa na zasadzie sprzężenia zwrotnego, gdzie wyjścia są podłączone do wejść. Dzięki temu przerzutnik jest w stanie utrzymać ostatni stan wyjść, nawet po przejściu stanów logicznych na wejściach w stan neutralny (nic nie jest podpięte na wejścia). Stan, w którym oba wejścia są w stanie wysokim, jest stanem zabronionym. Jest on niezgodny z definicją stanów wyjściowych przerzutnika.

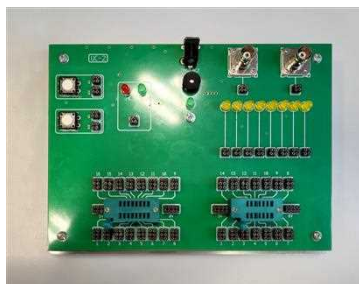
S	R	$Q_n$	$\bar{Q}_n$
0	0	stan pamiętania	stan pamiętania
0	1	0	1
1	0	1	0
1	1	stan zabroniony	stan zabroniony

Tablica 2: Tablica prawdy dla przerzutnika RS zbudowanego na bramkach NOR

## 2 Ćwiczenia

### 2.1 Ćwiczenie 4.1

Pierwsze zadanie polegało na zapoznaniu się z płytką UC-2 do badania układów scalonych TTL. Zostały przeprowadzone wnikliwe badania wszystkich oferowanych funkcji przez płytkę, w celu upewnienia się, że funkcjonuje ona poprawnie.



Rysunek 6: Płytką UC-2 do badania układów scalonych TTL używana podczas zajęć,

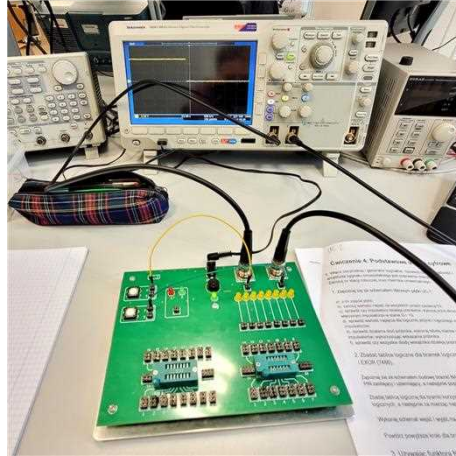
Źródło: Opracowanie własne

W pierwszej kolejności zostały sprawdzone piny zasilające 5V, znajdujące się przy mocowaniach do układów TTL. Oba wskazały równo odczyt 5V podczas pomiarów miernikiem.

Następnie dokonany został pomiar dotyczący działania diod wskaźnika. Żadna z nich nie wskazała nieprawidłowego działania.

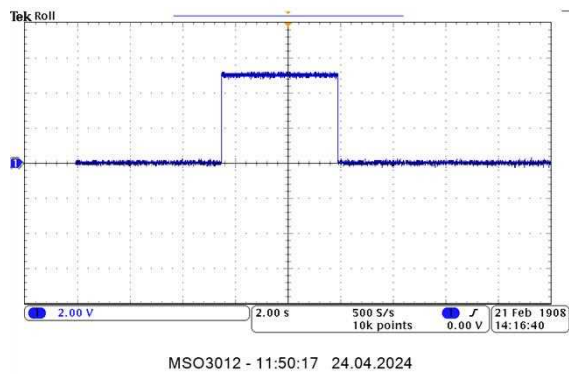
### 2.1.1 Sprawdzenie poprawności działania impulsatorów

Do wykonania powyższych pomiarów został wykorzystany oscyloskop. Oba impulsatory zostały wpięte do oscyloskopu w celu wizualizacji zmiany napięcia w czasie.

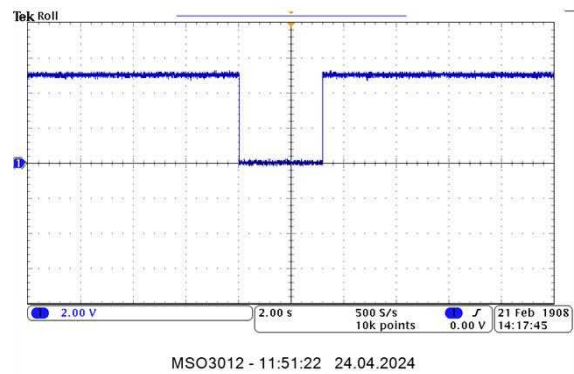


Rysunek 7: Realizacja podpięcia impulsatorów do oscyloskopu,  
Źródło: Opracowanie własne

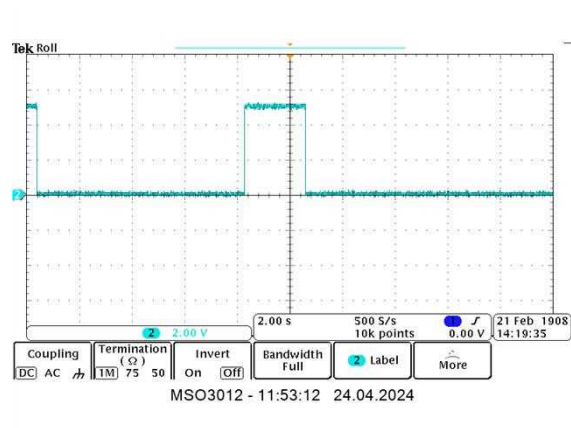
Górny impulsator został wpięty na kanał pierwszy a dolny na kanał drugi.



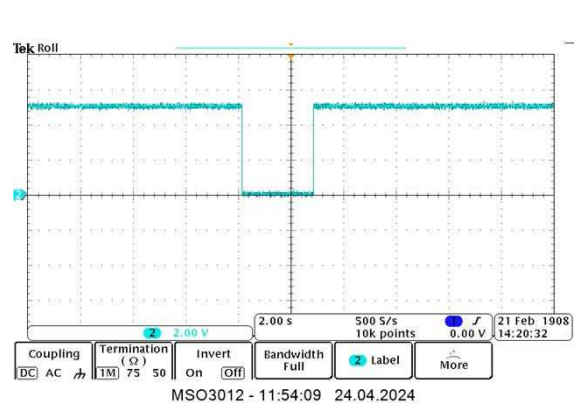
Rysunek 8: Zmiana w czasie wyjścia  $Q$  górnego impulsatora z widocznym momentem wciśnięcia przycisku,  
Źródło: Opracowanie własne



Rysunek 9: Zmiana w czasie wyjścia  $\overline{Q}$  górnego impulsatora z widocznym momentem wciśnięcia przycisku,  
Źródło: Opracowanie własne



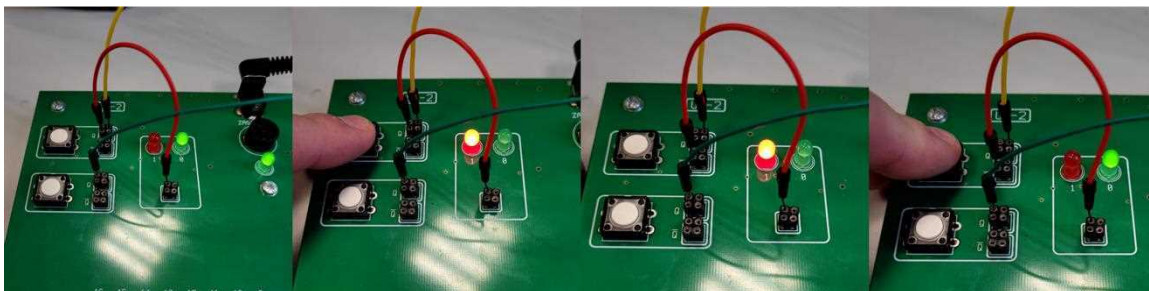
Rysunek 10: Zmiana w czasie wyjścia  $Q$  dolnego impulsatora z widocznym momentem wciśnięcia przycisku,  
Źródło: Opracowanie własne



Rysunek 11: Zmiana w czasie wyjścia  $\bar{Q}$  dolnego impulsatora z widocznym momentem wciśnięcia przycisku,  
Źródło: Opracowanie własne

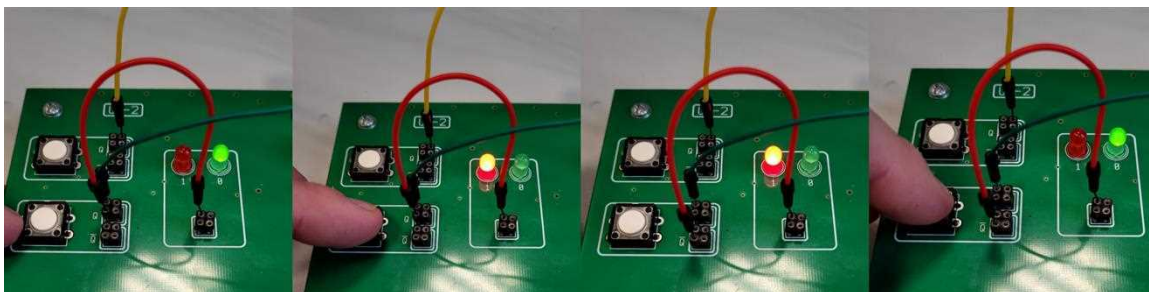
### 2.1.2 Sprawdzenie poprawności działania diod próbnika

Sprawdzenie to miało na celu uniknięcie błędów na późniejszym etapie ćwiczeń, gdzie próbnik diod led był często używany. Test polegał na sprawdzeniu wszystkich możliwych kombinacji które dało się uzyskać z dwoma impulsatorami.



Rysunek 12: Seria zdjęć ukazująca działanie próbnika diod dla podłączonego górnego impulsatora,  
Źródło: Opracowanie własne

Na obu seriach zdjęć widzimy sprawdzenia dla sytuacji gdy przycisk nie był wciśnięty oraz gdy był. Sprawdzenie zostało przeprowadzone dla obu wejść  $Q$  oraz  $\bar{Q}$ .



Rysunek 13: Seria zdjęć ukazująca działanie próbnika diod dla podłączonego dolnego impulsatora,  
Źródło: Opracowanie własne

Przedstawiając to w formie tabelki:



Impulsator	Q	$\overline{Q}$	Q oraz wciśnięty przycisk	$\overline{Q}$ oraz wciśnięty przycisk
Górny	0	1	1	0
Dolny	0	1	1	0

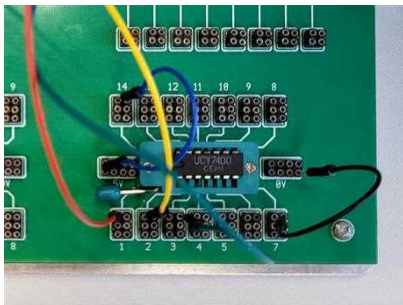
Tablica 3: Prezentacja działania diod próbnika dla dwóch różnych impulsatorów

## 2.2 Ćwiczenie 4.2

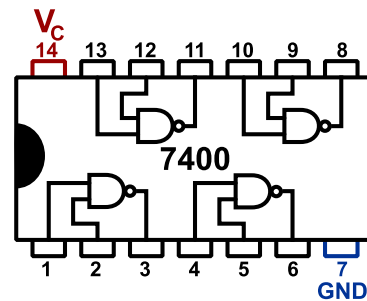
Zadanie drugie dotyczyło badania tablic logicznych bramek NAND, NOR oraz EXOR. Do tego zadania zostały użyte układy scalone odpowiednio o numerach 7400, 7402 oraz 7486.

### 2.2.1 NAND(7400)

W celu zbadania wejść oraz wyjść wpiętych został sporządzony schemat działania układu. Wykorzystana została do tego dokumentacja układu dostępna w Internecie. Przedstawiony poniżej rysunek został zmodyfikowany tak, jak opisane są wejścia na płytce UC-2 w celu uproszczenia opisów zadań.

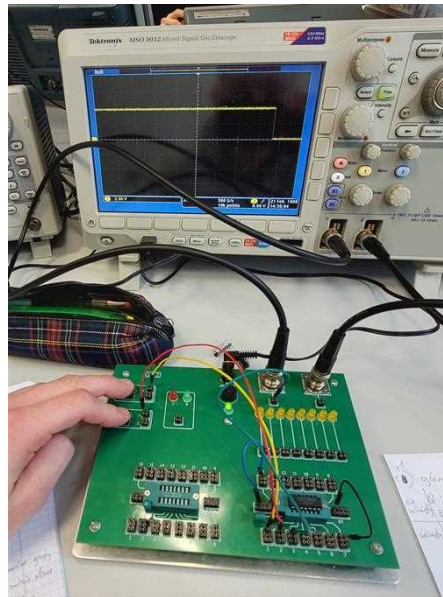


Rysunek 14: Układ 7400 umieszczony poprawnie w gnieździe na płytce UC-2,  
Źródło: Opracowanie własne



Rysunek 15: Schemat układu 7400 z bramkami NAND – opis wejść/wyjść,  
Źródło: [Wikipedia](#)

Z pomocą impulsatorów zostały dokonane testy każdej z bramek (1,2,3), (4,5,6), (8,9,10) oraz (11,12,13).



Rysunek 16: Poprawna reakcja Bramki (1,2,3) układu 7400 na dwie bitowe jedynki (spadek napięcia),  
Źródło: Opracowanie własne

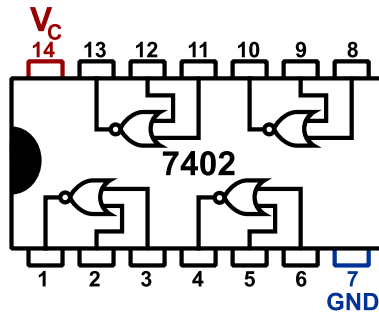
Żadna z bramek nie wykazała błędnego działania. Na powyższym rysunku jesteśmy w stanie zauważyć poprawną reakcję jednej z nich. Wyjścia  $Q$  impulsatorów zostały wpięte na wejścia 1 oraz 2 układu a wyjście 3 jest obrazowane na ekranie



oscylloskopu. Naciśnięcie przycisków impulsatorów wywołuje zanegowane wyjście  $Q$ , co dla bramki NAND oznacza spadek napięcia.

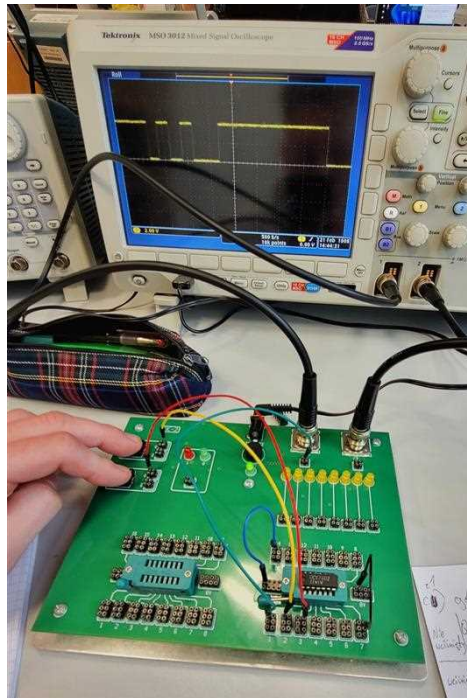
### 2.2.2 NOR(7402)

Aby zbadać wejścia oraz wyjścia układu wpierw został sporządzony schemat działania układu. Użyta została dokumentacja układu dostępna w Internecie. Przedstawiony poniżej rysunek został zmodyfikowany tak, jak wejścia na płytce UC-2 aby uprościć opis zadań.



Rysunek 17: Schemat układu 7402 z bramkami NOR – opis wejść/wyjść,  
Źródło: Opracowanie własne

Z pomocą impulsatorów zostały dokonane testy każdej z bramek (1,2,3), (4,5,6), (8,9,10) oraz (11,12,13).



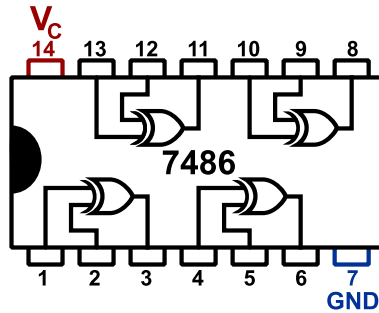
Rysunek 18: Poprawna reakcja Bramki (1,2,3) układu 7402 na dwie bitowe jedynki (spadek napięcia),  
Źródło: Opracowanie własne

Żadna z bramek nie wykazała błędnego działania. Na powyższym rysunku jesteśmy w stanie zauważyć poprawną reakcję jednej z nich. Wyjścia  $Q$  impulsatorów zostały wpięte na wejścia 2 oraz 3 układu a wyjście 1 jest obrazowane na ekranie oscylloskopu. Jak jesteśmy w stanie dostrzec na ekranie oscylloskopu, gdy oba przyciski nie były wciśnięte, sygnał pozywał ok. 4V co odbierane jest jako stan wysoki.

### 2.2.3 EXOR(7486)

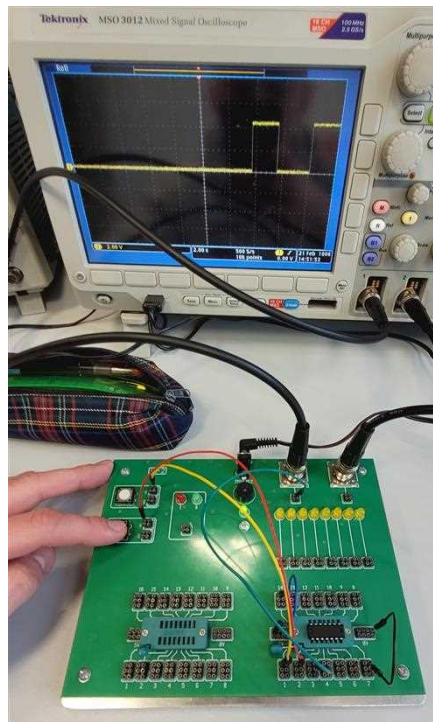
Żeby zbadać wejścia oraz wyjścia układu na początku został sporządzony schemat działania układu. Użyta została dokumentacja układu dostępna w Internecie. Przedstawiony poniżej schemat został zmodyfikowany tak, jak wejścia na

płytkę UC-2 aby uprościć opis zadań.



Rysunek 19: Schemat układu 7486 z bramkami EXOR – opis wejść/wyjść,  
Źródło: Opracowanie własne

Z pomocą impulsatorów zostały dokonane testy każdej z bramek (1,2,3), (4,5,6), (8,9,10) oraz (11,12,13).



Rysunek 20: Poprawna reakcja Bramki (1,2,3) układu 7486 na jedną bitową jedynekę i jedno zero (wzrost napięcia),  
Źródło: Opracowanie własne

Żadna z bramek nie wykazała błędnego działania. Na powyższym rysunku jesteśmy w stanie zauważyć poprawną reakcję jednej z nich. Wyjścia  $Q$  impulsatorów zostały wpięte na wejścia 1 oraz 2 układu a wyjście 3 jest obrazowane na ekranie oscyloskopu. Jak jesteśmy w stanie dostrzec na ekranie oscyloskopu, wciśnięcie jednego przycisku, wywołuje pożądany wzrost napięcia.

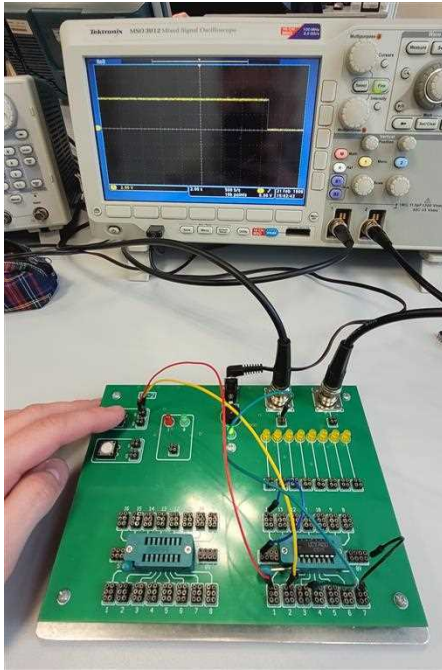
## 2.3 Ćwiczenie 4.3

Zadanie numer 3 polegało na zrealizowaniu operacji negacji, iloczynu oraz sumy przy użyciu tylko jednego typu bramek. Użyte do tego zostały schematy na rysunku(3).

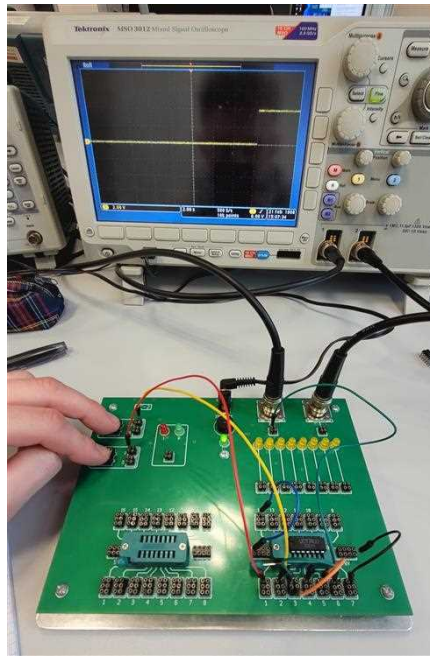
### 2.3.1 Realizacja przy użyciu bramek NAND

#### Negacja

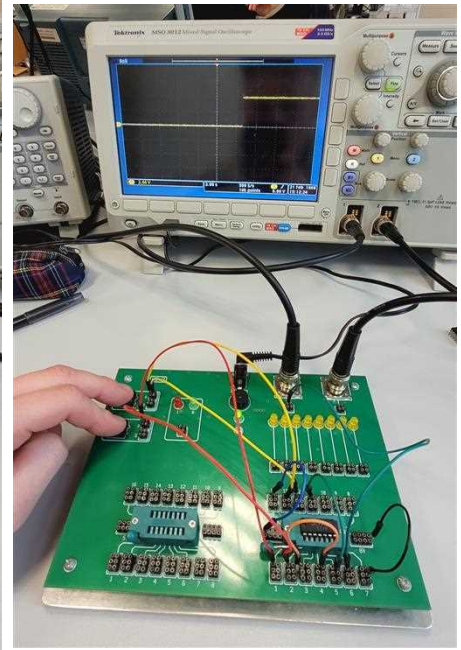
W tej implementacji została użyta jedna bramka z wejściami 1 oraz 2 na które podawany jest ten sam sygnał. Wyjście posiada numer 3.



Rysunek 21: Operacji negacji zrealizowana przy użyciu bramki NAND,  
Źródło: Opracowanie własne



Rysunek 22: Operacji iloczynu zrealizowana przy użyciu bramki NAND,  
Źródło: Opracowanie własne



Rysunek 23: Operacji sumy zrealizowana przy użyciu bramki NAND,  
Źródło: Opracowanie własne

### Iloczyn

Do stworzenia układu wykonującego iloczyn potrzebne były dwie bramki NAND. Skorzystałem z bramek (1,2,3) oraz (4,5,6). Układ przyjmował sygnał na wejścia 1 oraz 2. Wynik operacji natomiast był dostępny na wyjściu numer 6.

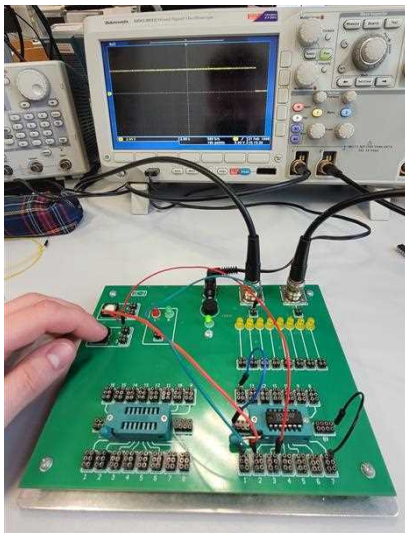
Obie bramki zostały połączone poprzez wpięcie wyjścia 3 na wejścia 4 oraz 5.

### Suma

Funkcja logiczna sumy potrzebowała już 3 wykorzystanych bramek (1,2,3), (4,5,6) oraz (13,12,11). Pierwsza i ostatnia bramka przyjmowały ten sam sygnał na oba wejścia a wyjścia 3 oraz 11 zostały podłączone do drugiej bramki. Wynik uzyskujemy na wyjściu numer 6.

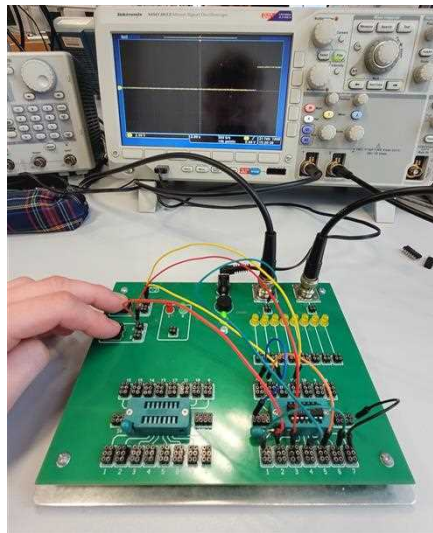


### 2.3.2 Realizacja przy użyciu bramek NOR



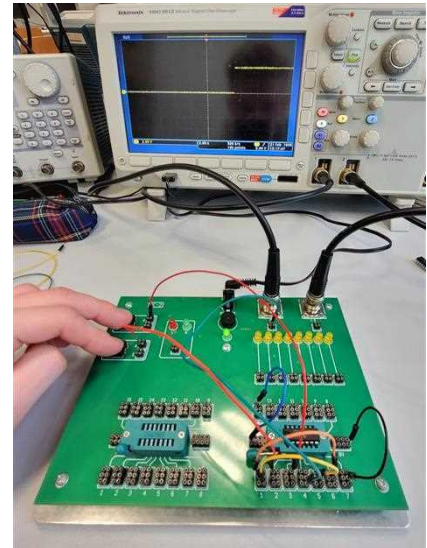
Rysunek 24: Operacji negacji zrealizowana przy użyciu bramki NOR,

Źródło: Opracowanie własne



Rysunek 25: Operacji iloczynu zrealizowana przy użyciu bramki NOR,

Źródło: Opracowanie własne



Rysunek 26: Operacji sumy zrealizowana przy użyciu bramki NOR,

Źródło: Opracowanie własne

#### Negacja

W tej implementacji została użyta jedna bramka z wejściami 2 oraz 3 na które podawany jest ten sam sygnał. Wyjście posiada numer 1.

#### Iloczyn

Funkcja logiczna iloczynu potrzebowała już 3 wykorzystanych bramek (1,2,3), (4,5,6) oraz (13,12,11). Pierwsza i ostatnia bramka przyjmowały ten sam sygnał na oba wejścia a wyjścia 1 oraz 13 zostały połączone do drugiej bramki. Wynik uzyskujemy na wyjściu numer 4.

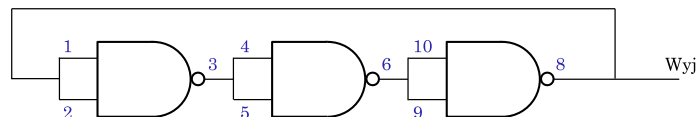
#### Suma

Do stworzenia układu wykonującego sumę potrzebne były dwie bramki NOR. Skorzystałem z bramek (1,2,3) oraz (4,5,6). Układ przyjmował sygnał na wejścia 2 oraz 3. Wynik operacji natomiast był dostępny na wyjściu numer 4.

Obie bramki zostały połączone poprzez wpięcie wyjścia 1 na wejścia 5 oraz 6.

## 2.4 Ćwiczenie 4.4

Zadanie czwarte dotyczyło wyznaczenia średniego czasu propagacji impulsu przez bramkę mierząc okres drgań generatora zbudowanego z trzech bramek. Do budowy generatora miały być użyte bramki 7400 oraz 74S00.



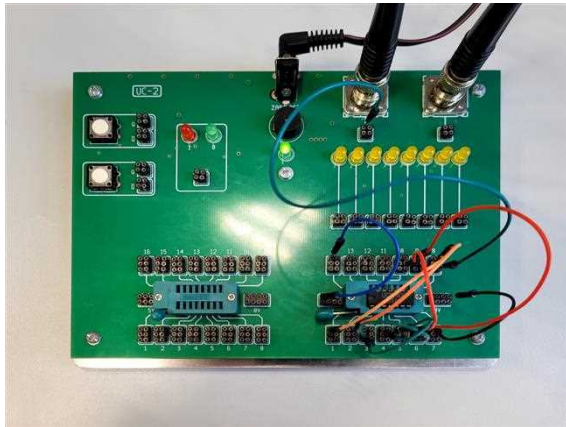
Rysunek 27: Schemat generatora który miał zostać zbudowany w tym zadaniu (przy użyciu bramek NAND),

Źródło: Opracowanie własne

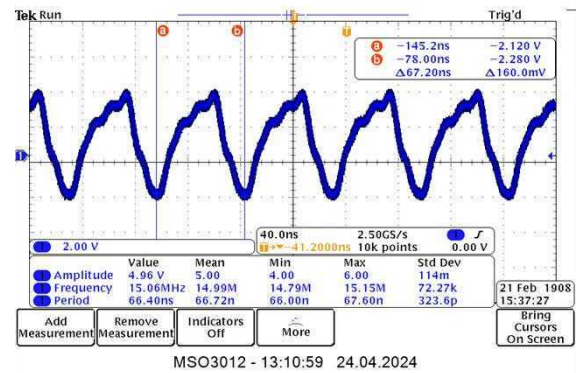
Powyższy schemat został dodatkowo wzbogacony o numery wejść/wyjść jakie zostały użyte przy budowie układu na płytce UC-2. Sygnał uzyskany z bramek został wpuszczony na oscylator przez kanał 1 który posiadał oporność  $1M\Omega$ .

### 2.4.1 7400

Jak jesteśmy w stanie zaobserwować, okres drgań dla układu 4700 wynosi ok.  $66,4ns$ .



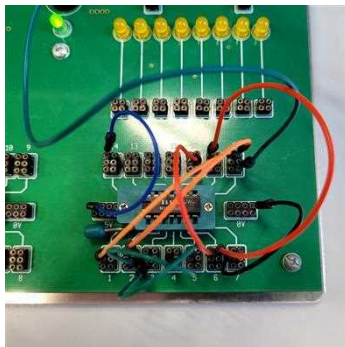
Rysunek 28: implementacja generatora zbudowana przy użyciu bramek NAND 4700,  
Źródło: Opracowanie własne



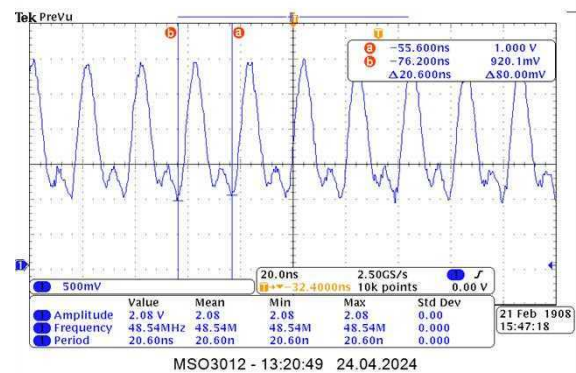
Rysunek 29: Pomiar okresu drgań generatora zbudowanego z bramek NAND 4700,  
Źródło: Opracowanie własne

### 2.4.2 74S00

W tym przypadku natomiast okres drgań jest mniejszy, wynoszący ok. jedynie  $20,6ns$ . Różnicę zauważyć idzie również w amplitudzie która zamiast 5V wskazuje tylko okolice 2V.



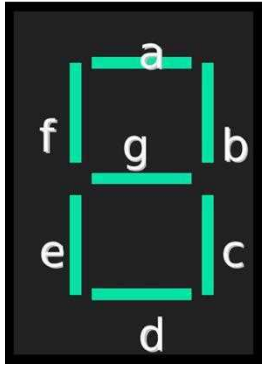
Rysunek 30: implementacja generatora zbudowana przy użyciu bramek NAND 47S00,  
Źródło: Opracowanie własne



Rysunek 31: Pomiar okresu drgań generatora zbudowanego z bramek NAND 47S00,  
Źródło: Opracowanie własne

## 2.5 Ćwiczenie 4.5

Zadanie to było robione na końcu zajęć oraz polegało na zbudowaniu funkcji logicznej dla jednego wybranego segmentu (a, b, c, d, e, f, g) wskaźnika 7-segmentowego, którego zadaniem będzie wyświetlanie liczb w systemie ósemkowym.



Rysunek 32: Zdjęcie poglądowe wyświetlacza z podpisanymi segmentami,  
Źródło: [Strona wykładów](#)

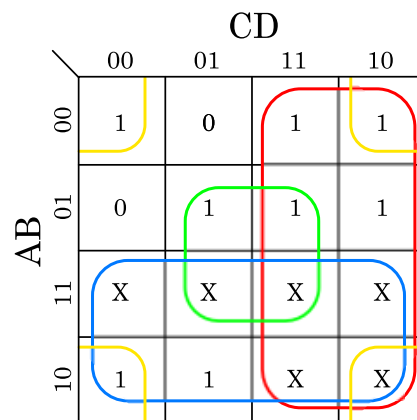
CYFRA	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Rysunek 33: Tabela przedstawiająca stany dla każdego możliwego segmentu wyświetlacza,  
Źródło: [Strona wykładów](#)

Do zbudowania wybrany został segment *a*. Każda z liczb została rozpisana binarnie na bity w celu stworzenia mapy Karnaugh.

Binarnie (ABCD)	Cyfra	Segment A
0000	0	1
0001	1	0
0010	2	1
0011	3	1
0100	4	0
0101	5	1
0110	6	1
0111	7	1
1000	8	1
1001	9	1

Rysunek 34: Reakcja segmentu *A* dla poszczególnych liczb wraz z ich binarnym zapisem,  
Źródło: Opracowanie własne

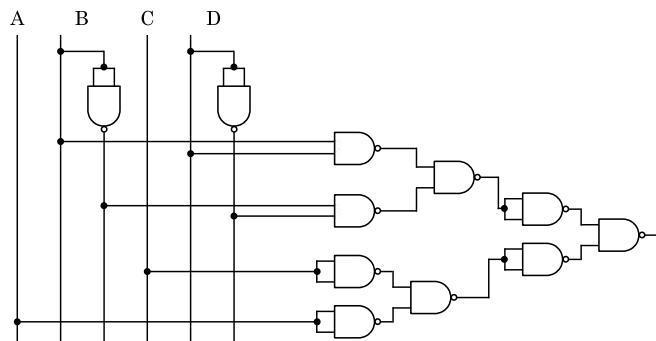


Rysunek 35: Mapa Karnaugh stworzona w celu minimalizacji funkcji logicznej,  
Źródło: Opracowanie własne

Funkcja po zminimalizowaniu mapą Karnaugh prezentuje się następująco:

$$C + A + BD + \overline{B} \overline{D} \quad (1)$$

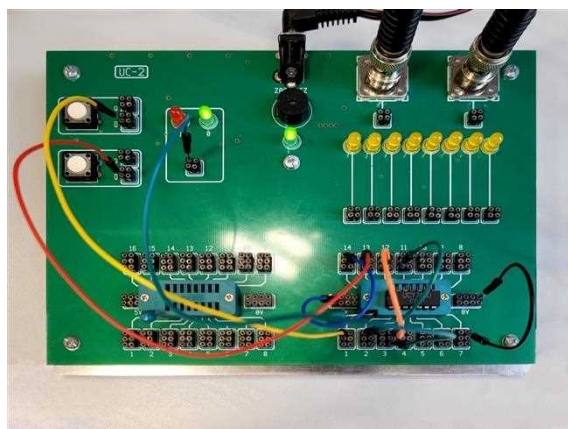
Znaki *X* oznaczają dowolne pola, czyli takie których stan nas nie interesuje. Pozwala to na większą minimalizację w sytuacji gdy ustawimy je wszystkie na wartość 1.



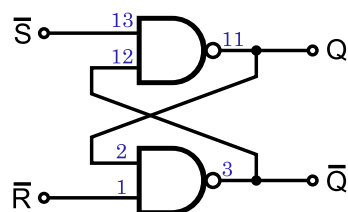
Rysunek 36: Schemat segmentu *a* wykonanu przy użyciu tylko bramek NAND,  
Źródło: Opracowanie własne

## 2.6 Ćwiczenie 4.6

Ostatnim zadaniem było zaprojektowanie układu działającego jak przerzutnik asynchroniczny RS. Na widocznym poniżej schemacie zostały dopisane numery wejść/wyjść z których skorzystano przy jego budowie.



Rysunek 37: Poprawnie zmontowany układ RS z wykorzystaniem bramek NAND,  
Źródło: Opracowanie własne



Rysunek 38: Schemat przerzutnika RS zbudowanego z bramek NAND z podpisanymi numerami wejść/ wyjść,  
Źródło: Opracowanie własne

Sposób działania układu został przeze mnie zarejestrowany cyfrowo w formie filmu.[\[Link do nagrania\]](#). Na jego podstawie została stworzona tablica stanów układu.

Górny impulsator na płytce służył jako wejście *RESET* a dolny jako *SET*. Warto zauważyć, że przerzutnik, nawet gdy wejścia pozostają odpięte od źródła zasilania, “pamięta” swój stan.

$\bar{S}$	$\bar{R}$	$Q_n$	$\bar{Q}_n$
0	0	stan zabroniony	stan zabroniony
0	1	1	0
1	0	0	1
1	1	stan pamiętania	stan pamiętania

Tablica 4: Tablica prawdy dla przerzutnika RS zbudowanego nan NAND’ach



## 3 Omówienie wyników

### 3.1 Ćwiczenie 4.1

Każda z funkcji płytki działa poprawnie. Nie zostały napotkane żadne anomalie podczas przeprowadzania pomiarów kontrolnych.

### 3.2 Ćwiczenie 4.2

We wszystkich 3 układach sprawdzone zostały każde z wejść oraz wyjść. Każde z nich dawało oczekiwany wynik względem odpowiadających im bramek. Odczyt z oscyloskopu we każdym z przypadków utrzymywał się w granicach przewidywanego napięcia wyjściowego — przyjmował wartości ok.  $4V$  w stanie wysokim.

### 3.3 Ćwiczenie 4.3

Dla każdego ze zbudowanych układów logicznych proces przebiegł pomyślnie. Układy odpowiadały według tablic prawdy im odpowiadających (2). Potwierdza to fakt, że przy pomocy tylko bramek NAND bądź NOR można zbudować każdą funkcję logiczną.

### 3.4 Ćwiczenie 4.4

Budowę zdanego układu udało wykonać zgodnie z planem. Udało się uzyskać pożądany efekt drgań. Niestety odczyt wykonany z użyciem oscyloskopu nie został przeprowadzony równie uważnie. Samo wyświetlenie napięcia wyjściowego zostało jak najbardziej wykonane, łącznie z prostym pomiarem okresu drgań. Jednak z niewiadomych przyczyn zapisany został tylko screenshot z napięciem wyjściowym, bez  $U_{WE}$ . Uniemożliwiło to w późniejszym procesie obliczenie wymaganego czasu propagacji impulsu.

To co możemy wywnioskować z uzyskanych okresów to to, że wersja 47S00 jest szybsza od zwykłego układu 4700. Pokrywa się to z informacjami dostępnymi w sieci które wprost mówią, że jest to seria pochodnych, wykonywana w technologiach S, czyli o podwyższonej szybkości.

### 3.5 Ćwiczenie 4.5

Z racji wykonywania tego zadania na samym końcu zajęć, zostało ono wpierw wykonane z błędem. Wykrycie złej funkcjonalności odkryto dopiero na etapie testowania zbudowanego układu. Błąd polegał na zinterpretowaniu pól  $X$  jako zer zamiast jedynek. Wybranie jedynek pozwoliło na stworzenie układu działającego według zadanej tablicy prawdy, jednak zostało dopiero dokonane na etapie pracy nad tym sprawozdaniem. Z tego też powodu zadanie nie posiada zdjęcia zbudowanego układu — źle zbudowany został umieszczony w sekcji *Notatki z zajęć*.

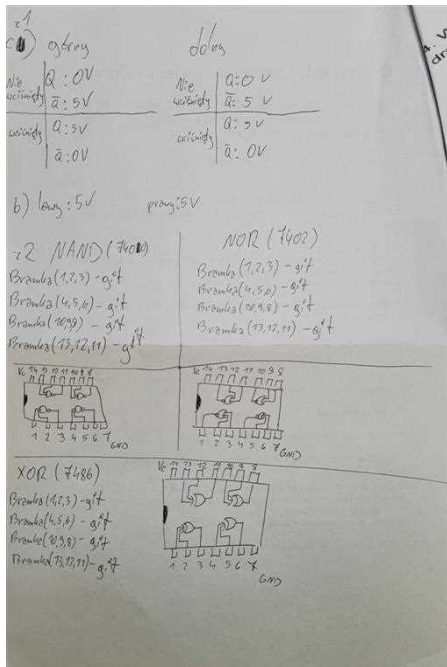
### 3.6 Ćwiczenie 4.6

Ostatnie zadanie przebiegło bez większych problemów, układ został zmontowany pomyślnie. Ważnym aspektem tego układu jest faktyczne “pamiętanie” stanu, nawet podczas odpięcia wejść. Najlepiej obrazuje to nagranie wideo stworzone podczas zajęć.

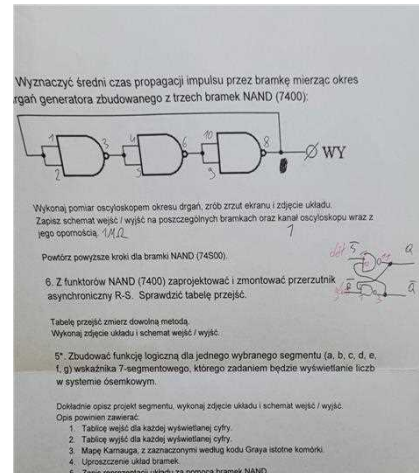
## 4 Podsumowanie

Podsumowując, serie eksperymentów przyniosły różnorodne wnioski i doświadczenia. Zadania coraz bardziej wywierają nacisk na myślenie zamiast na wykonywaniu zadań bez zastanowienia. Wypracowuje to w studentach umiejętności potrzebne do samodzielnej pracy. Większość z ćwiczeń przebiegła pomyślnie, kilka zostały dotknięte czynnikiem ludzkim który nie zawsze jest perfekcyjny. Ważne jest, żeby brać pod uwagę zarówno sukcesy, jak i trudności, ponieważ oba mogą prowadzić do cennych wniosków i lepszych umiejętności w przyszłych projektach.

## 5 Notatki z zajęć



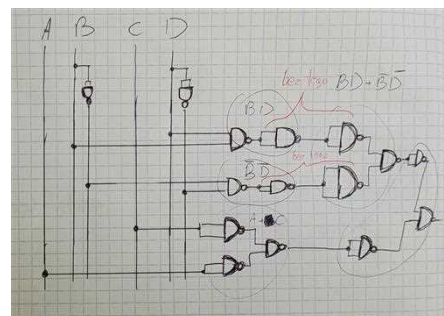
Rysunek 39: Notatki sporządzone podczas sprawdzania  
bramek,  
Źródło: Opracowanie własne



Rysunek 40: Notatki wykonane podczas robienia zadań 4  
oraz 6,  
Źródło: Opracowanie własne

binarna cyfra	ośmiocyfrowa
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Rysunek 41: tablica prawdy do segmentu a z zadania 5,  
Źródło: Opracowanie własne



Rysunek 42: Poprawny schemat z bramek NAND do  
zadania 5, wykonany w czasie późniejszym,  
Źródło: Opracowanie własne

