

Politechnika Wrocławskiego
Wydział Informatyki i Telekomunikacji

Układy Cyfrowe i Systemy Wbudowane

Sprawozdanie z pierwszego ćwiczenia

15.10.2025

Autorzy:
Daniel Gościński 280878
Łukasz Duda 280916

Prowadzący:
dr inż. Sławomir Sambor

Spis treści

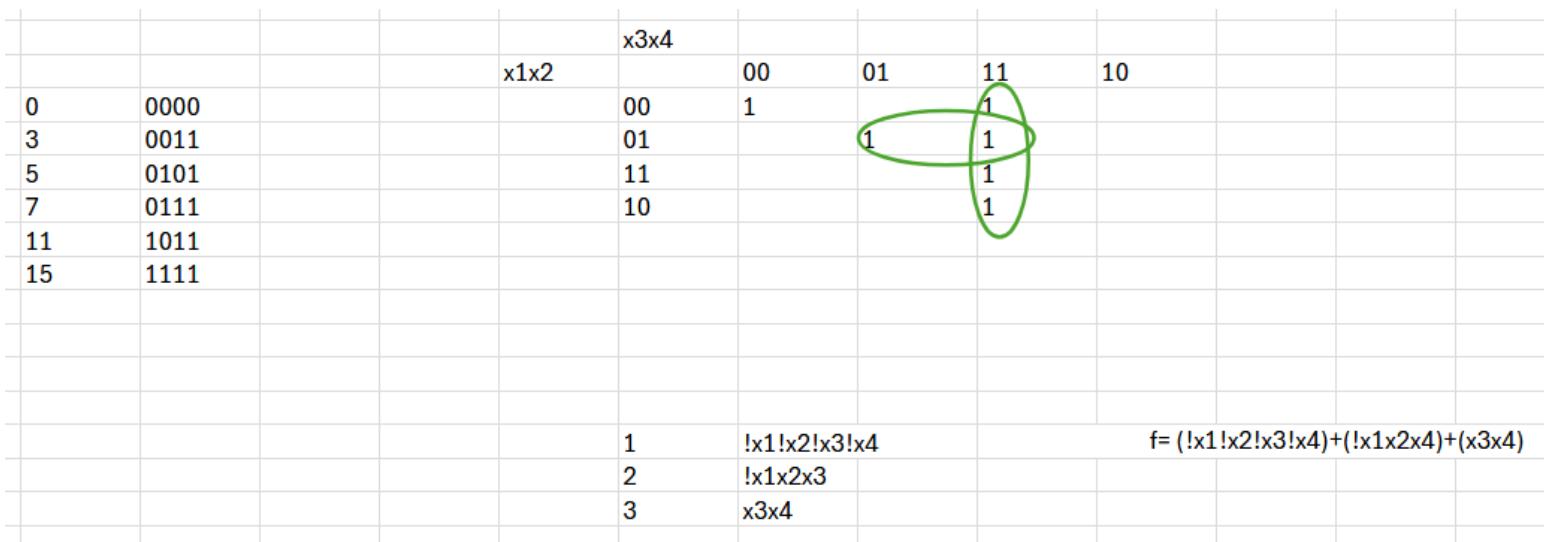
1. Treść zadania	3
2. Schemat VHDL	3
3. Opis schematu VHDL	4
4. Wyniki Symulacji	4
5. Wnioski	6

1. Treść zadania

Zadanie polegało na utworzeniu schematu pozwalającego na zapalanie diod na płytce ZL-9572 zgodnie z realizacją funkcji $f(x_1, x_2, x_3, x_4) = \Sigma(0, 3, 5, 7, 11, 15)$. Cztery pierwsze diody mają zawsze wyświetlać aktualnie wartość naciśniętą przyciskami, a piąta dioda ma zapalać się tylko we wskazanych wartościach, w innych przypadkach dioda jest zgaszona. Dodatkowo podłączony jest wyświetlacz 7-segmentowy, który zawsze wyświetla aktualną wartość.

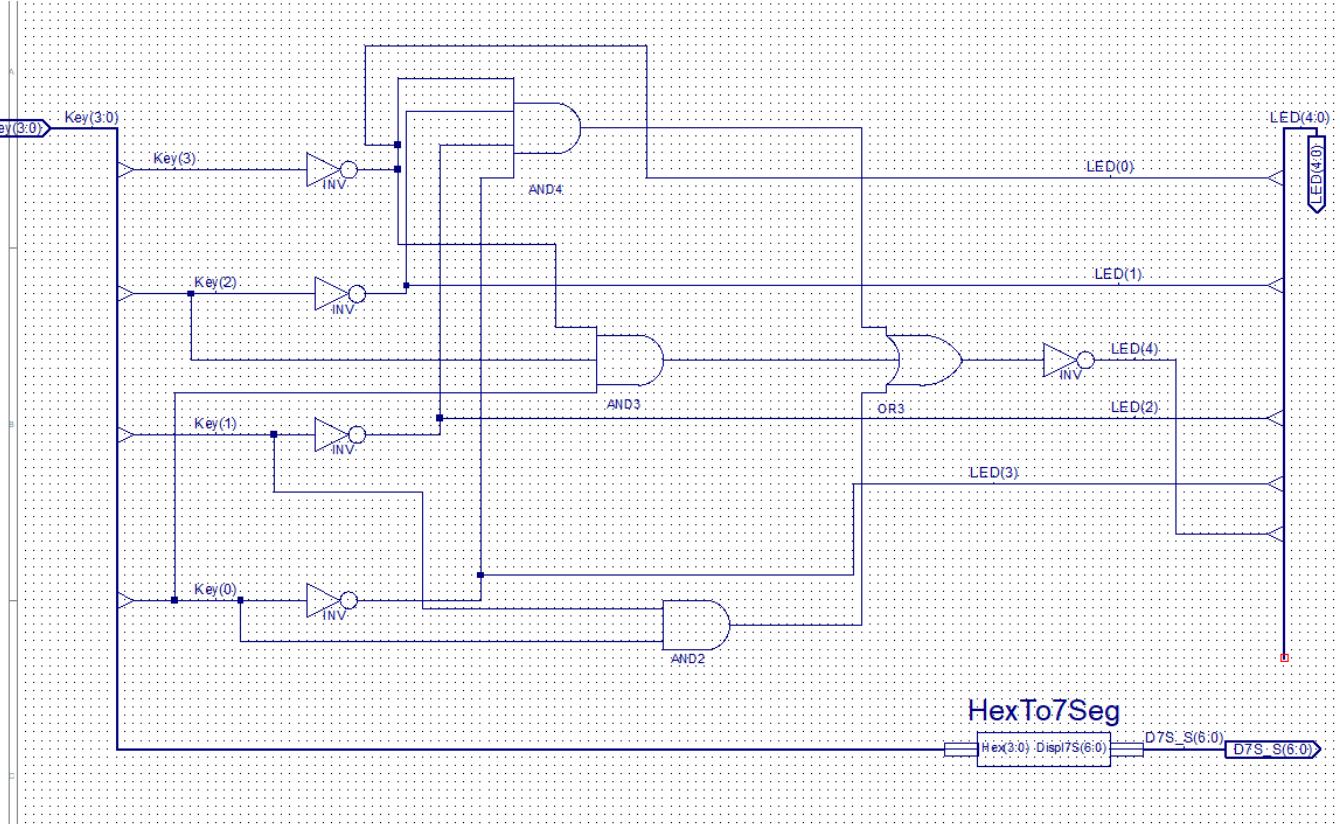
2. Schemat VHDL

W celu utworzenia schematu wykonano minimalizację funkcji uzyskanej z treści zadania przy pomocy siatki Karnaugha.



Zdjęcie 1: Minimalizacja funkcji metodą siatki Karnaугha

Uzyskaną funkcję $f = (!x1!x2!x3!x4) + (!x1x2x4) + (x3x4)$ zamieniono następnie na schemat.



Zdjęcie 2: Schemat utworzonego układu

3. Opis schematu VHDL

Utworzony schemat posiada 4 sygnały wejściowe **Key**, który wprowadzany jest bezpośrednio do bloku **HexTo7Seg**, który zajmuje się operacją wyświetlacza 7-segmentowego. Sygnały po odpowiedniej transformacji zgodnej z zadaną funkcją wychodzą sygnałem **LED**, które dalej idą na fizyczne diody LED. Należy zaznaczyć, że wynik operacji logicznych musi być zanegowany, ponieważ diody LED działają w logice ujemnej.

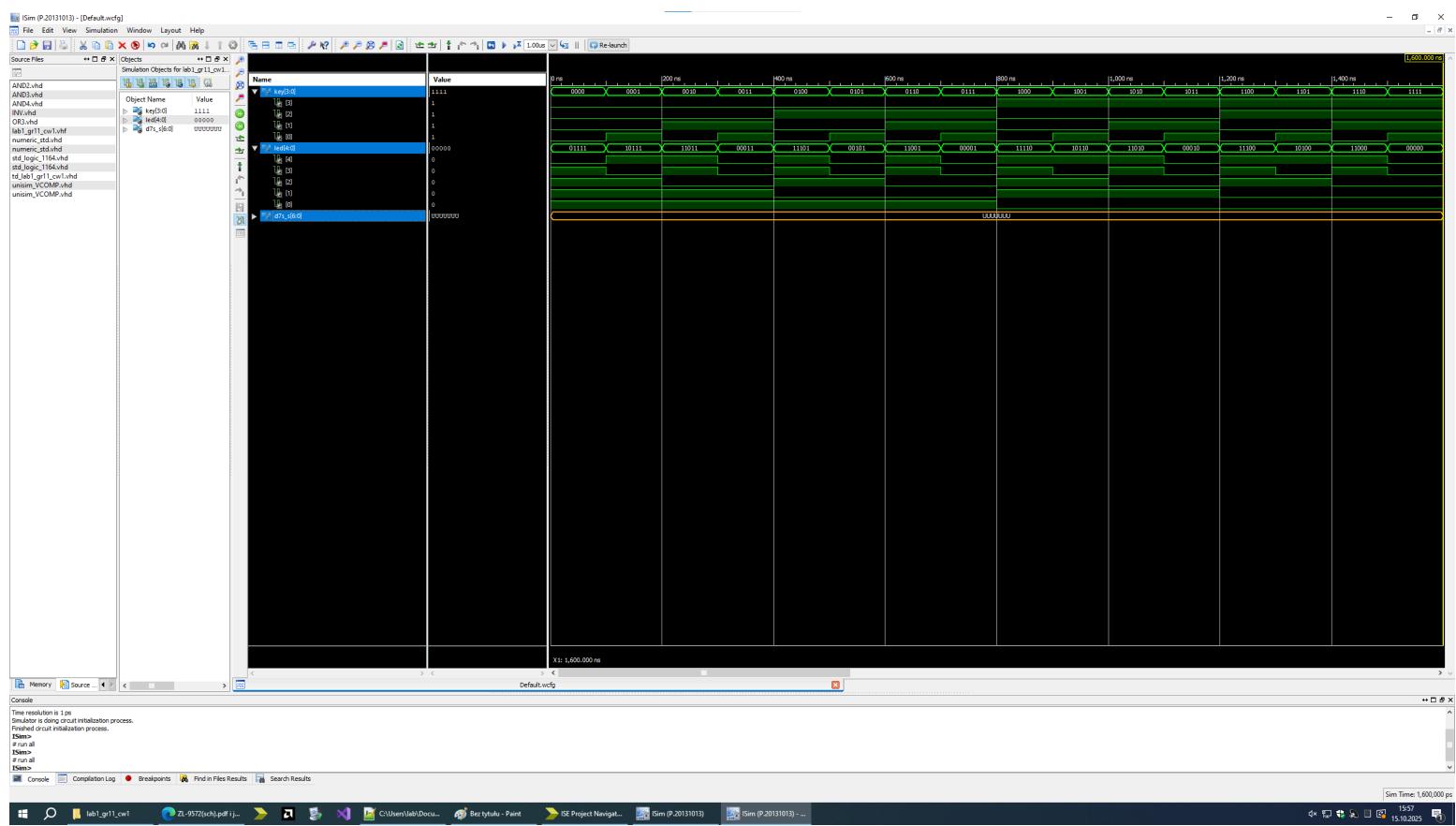
4. Wyniki Symulacji

Symulację wykonano poprzez ustawienie bitów wejścia **Key** na wszystkie wartości z zakreju 0-15. Na tej podstawie utworzono testbench, którym wykonywano symulację.

```
12 -- *** Test Bench - User Defined Section ***
42 tb : PROCESS
43 BEGIN
44   Key<="0000";
45   wait for 100 ns;
46   |   Key<="0001";
47   |   wait for 100 ns;
48   |   Key<="0010";
49   |   wait for 100 ns;
50   |   Key<="0011";
51   |   wait for 100 ns;
52   |   Key<="0100";
53   |   wait for 100 ns;
54   |   Key<="0101";
55   |   wait for 100 ns;
56   |   Key<="0110";
57   |   wait for 100 ns;
58   |   Key<="0111";
59   |   wait for 100 ns;
60   |   Key<="1000";
61   |   wait for 100 ns;
62   |   Key<="1001";
63   |   wait for 100 ns;
64   |   Key<="1010";
65   |   wait for 100 ns;
66   |   Key<="1011";
67   |   wait for 100 ns;
68   |   Key<="1100";
69   |   wait for 100 ns;
70   |   Key<="1101";
71   |   wait for 100 ns;
72   |   Key<="1110";
73   |   wait for 100 ns;
74   |   Key<="1111";
75   |   wait for 100 ns;
76   WAIT; -- will wait forever
77 END PROCESS;
78 -- *** End Test Bench - User Defined Section ***
80
81 END;
82
```

Zdjęcie 3: Kod testbencha zajmujący się ustawianiem sygnałów wejścia Key

Wykonana symulacja testbencha wskazuje na poprawne działanie układu, gdzie sygnał LED(4) poprawnie reaguje na wybrane w zadaniu sygnały

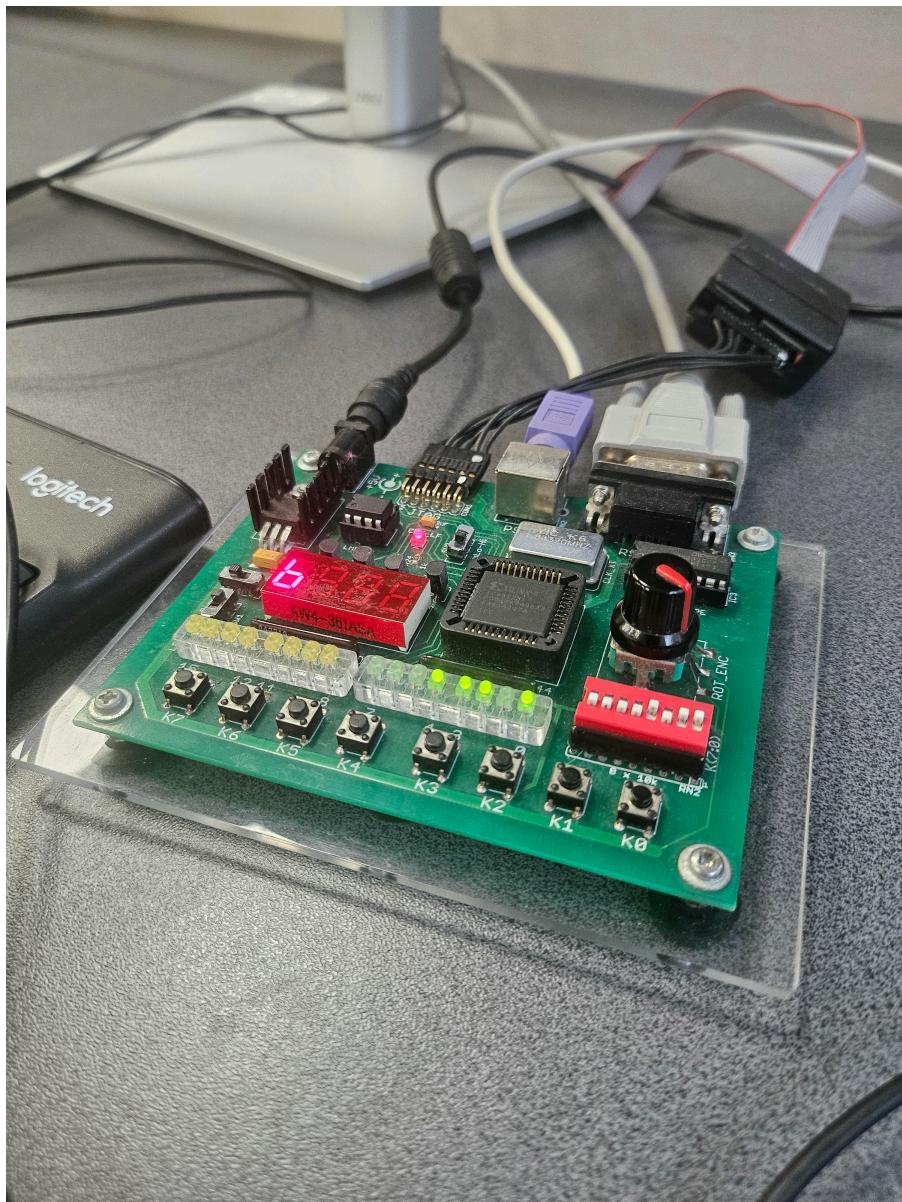


Zdjęcie 4: Wyniki symulacji

Należy zaznaczyć, że ponieważ diody LED płytka działają w logice ujemnej ('0' zapala diodę), sygnały LED są też w logice ujemnej. Sygnał wchodzący do układu `d7s` nie są możliwe do symulacji, ponieważ układ ten traktowany jest jako "czarna skrzynka".

5. Wnioski

Utworzony schemat poprawnie przeszedł symulację. Po załadowaniu programu do pamięci układu płytka, naciskanie diod LED poprawnie zapalało 4 ledy wskazujące na binarny zapis wartości, a piąta dioda poprawnie zapalała się w wartościach 0,3,5,7,11,15. Uzyskane rezultaty wskazują na poprawnie wykonane zadanie.



Zdjęcie 5: Efekt działania układu dla wawrztości 11

Wykonane zadanie pozwoliło na zapoznanie się z podstawami obsługi programu pozwalającego na pisanie w języku VHDL. Zapoznano się z podstawami utworzenia projektu, dodwania plików źródłowych oraz podstawami tworzenia układów. Uzyskana wiedza pozwoli na dalsze wykonywanie zadań na zajęciach.