

# PRURE

## Sprawozdanie

### Laboratorium nr 1

Data: 18.11.2024

Wykonująca: Anna Dzieżyk

#### Zadanie nr 1:

##### Opis wykonanych czynności:

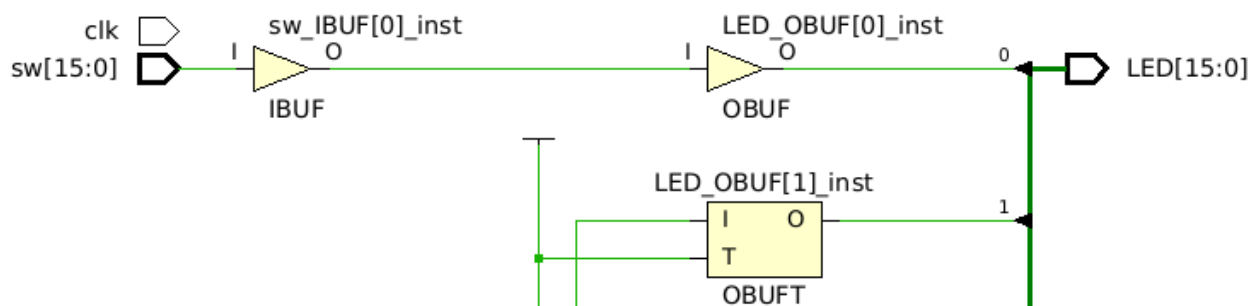
Zapoznanie ze środowiskiem IDE/CAD FPGA, płytą deweloperską Zybo Z7-10 oraz uruchomienie pierwszego programu na FPGA.

a) odpowiednia edycja pliku Basys3\_Master\_2.xdc – odkomentowanie linii odnoszących się tylko do obsługi LEDów, switchów oraz zegara;

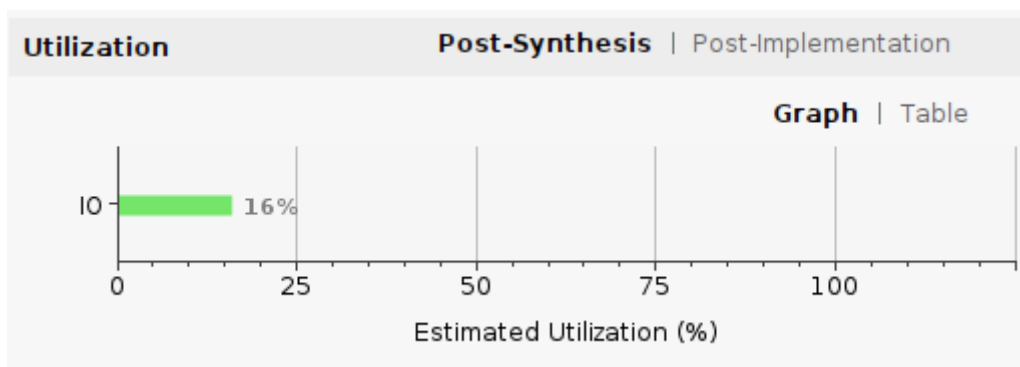
b) Inicjalizacja peryferiów w pliku first.vhd – przypisanie portów oraz nakazanie wykonania akcji – po ustawieniu switcha o indeksie zero w stan wysoki, ma się zapalić dioda o indeksie 0.

##### Wynik syntezy:

(np. schemat, zajętość zasobów)



Zrzut 1.: Najważniejsze elementy schematu po sytezie – switch, który ustawiamy (IBUF) orz led, który mu przypisaliśmy (OBUF)



Zrzut 2.: Zajętość zasobów po syntezie i po inplementacji – w tym przypadku to były te same wartości.

**Opracowany kod:**

(krótki opis głównych modułów)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_misc.all;
use ieee.numeric_std.all;

entity TOP is
  generic(
    constant N : integer := 10
  );
  port(
    clk: in std_logic;
    sw: in std_logic_vector (15 downto 0);
    LED: out std_logic_vector(15 downto 0)
  );
end TOP;

architecture behavioural of TOP is

begin
  -- zadanie 1
  LED(0) <= sw(0)
```

**Podsumowanie oraz wyniki:**

(krótki opis działania układu, wynik testów, screenshot np. z programowania itp.)

Tak, jak się spodziewano – po przełączeniu switcha 0 w górę – led przypisany do niego się zapalił.

**Zadanie nr 2:****Opis wykonanych czynności:**

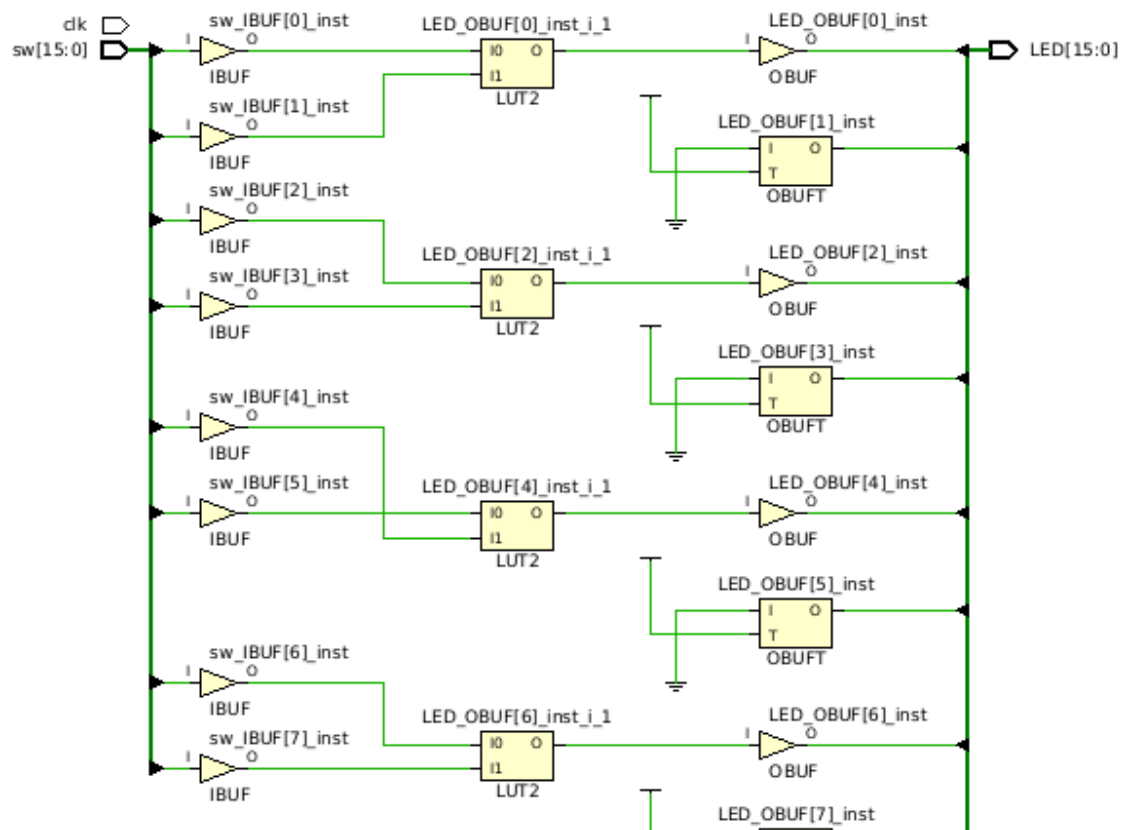
Na bazie przykładowego programu należało opracować układ, który:

- Realizuje operacje logiczne OR, AND, XOR, NAND, gdzie:
  - Dane wejściowe: przyciski switchy – 4 pary
  - Dane wyjściowe: diody LED o indeksach parzystych

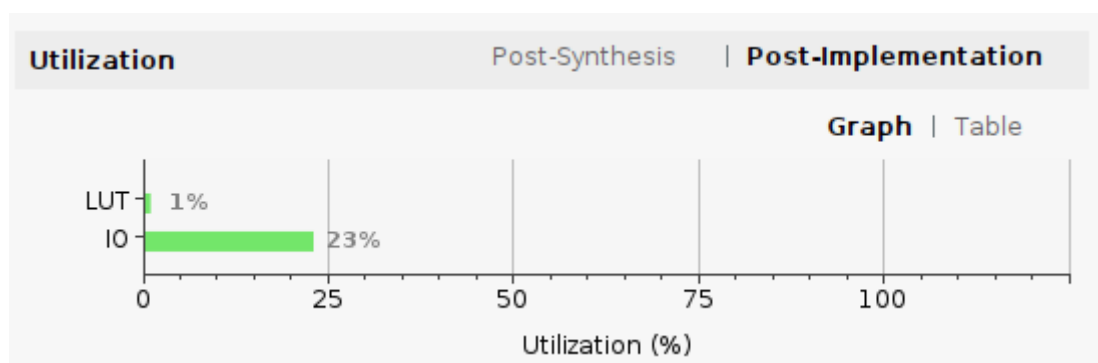
Każda operacja logiczna miała 2 dane wejściowe – z switchy i jako wyjście przypiano im diodę led o indeksie takim, jak parzysty indeks jednego ze switchów wejściowych.

**Wynik syntezy:**

(np. schemat, zajętość zasobów)



Zrzut 3.: Schemat układu – użyte zostały 4 pary switchy jako wejścia i 4 diody led jako wyjścia



Zrzut 4.: Zajętość zasobów po syntezie i po inplementacji – w tym przypadku to były te same wartości.

### Opracowany kod:

(krótki opis głównych modułów)

```

1  | library ieee;
2  | use      ieee.std_logic_1164.all;
3  | use      ieee.std_logic_unsigned.all;
4  | use      ieee.std_logic_misc.all;
5  | use      ieee.numeric_std.all;
6  |
7  | entity TOP is
8  |     generic(
9  |         constant N          : integer := 10
10 |     );
11 |     port(
12 |         clk: in std_logic;
13 |         sw: in std_logic_vector (15 downto 0);
14 |         LED: out std_logic_vector(15 downto 0)
15 |     );
16 | end TOP;
17 |
18 | architecture behavioural of TOP is
19 |
20 | begin
21 |     --
22 |     --      zadanie 2
23 |     LED(0) <= sw(0) AND sw(1);
24 |     LED(2) <= sw(2) OR sw(3);
25 |     LED(4) <= sw(4) XOR sw(5);
26 |     LED(6) <= sw(6) NAND sw(7);
27 | end behavioural;

```

### Podsumowanie oraz wyniki:

(krótki opis działania układu, wynik testów, screenshot np. z programowania itp.)

Tak, jak się spodziewano – układ działa poprawnie, zgodnie z tabelami prawdy bramek.