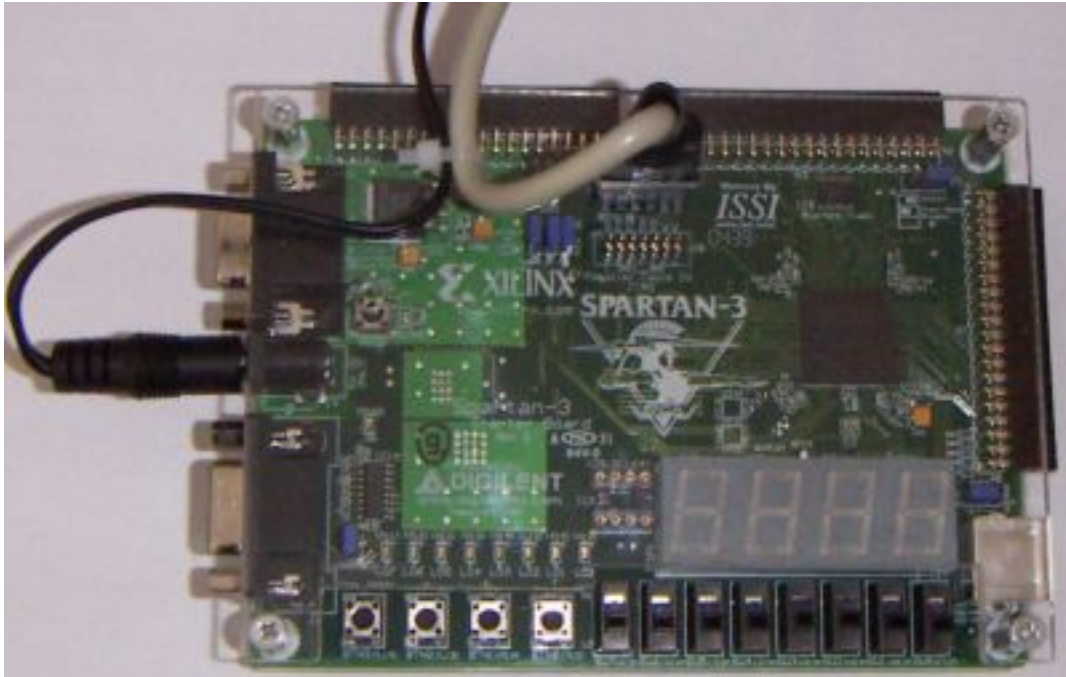


# Xilinx Spartan3



## Przykład 1

Bramka AND, wykonywana razem z prowadzącym

### Wejście:

przełącznik1: sw\_0

przełącznik2: sw\_1

### Wyjście:

dioda: led\_0

## Przykład 2

Zaprojektuj układ realizujący funkcję:  
wykorzystując jedynie bramki NAND 2 i 3 wejściowe.

A B		C D			
		00	01	11	10
00	00	1	1	0	1
01	00	1	1	0	1
11	00	0	0	0	1
10	00	1	1	0	1

1: Zaczynij od połączenia jedynek w bloki i wyznacz funkcję zminimalizowaną (od których wejść blok jest zależny?).

2: Pamiętaj, by '0' oznaczać negacją (kreska nad literką).

3: Gotową funkcję przedstaw do sprawdzenia prowadzącemu.

4: stwórz nowy projekt zgodnie z instrukcją, nazwij go "przykład2"

5: dodaj do projektu nowy schemat, nazwij go "przyklad2"

6: wklej plik przyklad2.ucf do odpowiedniego folderu (windows: C:/designs/przyklad2; Linux: folder "home" na pulpicie)

7: dodaj .ucf do projektu (kliknij na plik schematu prawym -> add files)

8: dodaj tyle bramek AND ile jest składowych wyrażenia, przykładowo wyrażenie  $xy+yz$  ma dwie składowe, jedna dwu i jedna trójwejściowa. Umieść bramki w linii jedna nad drugą, zachowaj między nimi odstęp około dwóch wysokości bramki. Aby dodać bramkę w wyszukiwarce elementów wpisujemy AND2 i AND3

9: dodaj bufony wejściowe (ibuf4) po lewej stronie i bufor wyjściowy (obuf) po prawej

**10:** dodaj do buforów etykiety i nazwij je zgodnie z tabelką

**11:** wyjścia bramek dołącz do bramki zbiorczej, oczywiście powinna mieć tyle wejść ile jest składowych wyrażenia

**12:** wyjście bramki zbiorczej dołącz do bufora wyjściowego

**13:** podłącz odpowiednie sygnały wejściowe do bramek, pamiętaj o negacjach.

**14:** dodaj drugie wyjście, podłącz tak, by było zanegowane i podłączona do niego dioda świeciła gdy wyjście układu będzie równe 0.

**15:** sprawdź, czy funkcja jest realizowana prawidłowo, dane do testu w kodzie Graya:

**podpowiedź:** bramkę NOT można stworzyć z bramki NAND-2 z połączonymi ze sobą wejściami

ABCD	out
0000	1
0001	1
0011	0
0010	1
0110	1
0111	0
0101	1
0100	1
1100	0
1101	0
1111	0
1110	1
1010	1
1011	0
1001	1
1000	1

**Wejście:**

przełącznik 1: sw\_A  
przełącznik 2: sw\_B  
przełącznik 3: sw\_C  
przełącznik 4: sw\_D

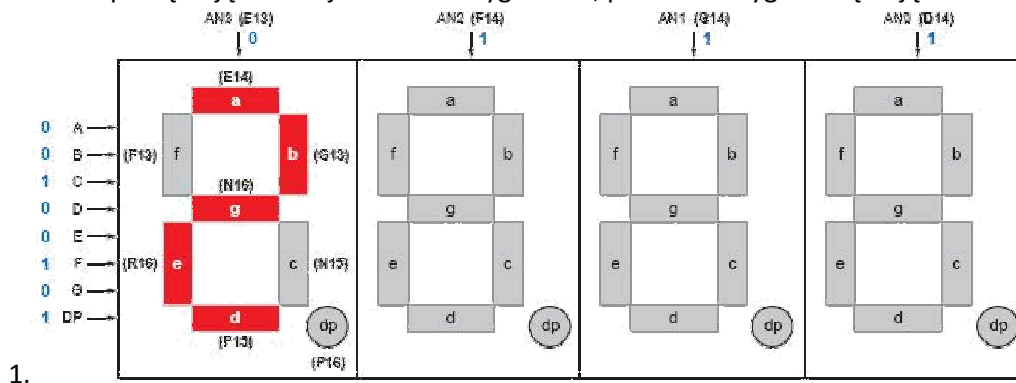
**Wyjście:**

dioda 1: led\_0  
dioda 2: led\_1

**Przykład 3**

**A)** Stwórz układ wyświetlający dziesiętnie na wyświetlaczu 7-segmentowym liczbę podaną dwójkowo przez przełączniki sw3...0

- 1: Stwórz nowy projekt, nazwij go "przyklad3"
- 2: dodaj nowy plik schematu i nazwij go "przyklad3"
- 3: wklej do folderu projektu plik *przyklad3.ucf* i dołącz go do projektu
- 4: wklej do folderu pliki *dekoder\_bcd.vhd* i *dekoder\_bcd.sym*
- 5: dodaj *dekoder\_bcd.vhd* do projektu tak samo jak plik .ucf
- 6: w wyszukiwarce elementów wpisz "dekoder" i dodaj ten element do schematu. Element ten
- 7: dodaj bufor wejściowy (ibuf4) oraz 3 zestawy buforów wyjściowych (obuf4), nadaj im odpowiednie nazwy
- 8: podłącz wyjścia dekodera do buforów z etykietami A...G i dot (kropka)
- 9: do pozostałych 4 buforów wyjściowych dodaj etykiety AN, wybierz jeden z wyświetlaczy i włącz go podłączając do wejścia bufora sygnał "0", pozostałe wygasz dołączając



#### Wejście:

przełącznik 1: sw\_0  
 przełącznik 2: sw\_1  
 przełącznik 3: sw\_2  
 przełącznik 4: sw\_3

#### Wyjście:

katody LCD:  
 lcd\_A  
 lcd\_B  
 lcd\_C  
 lcd\_D  
 lcd\_E  
 lcd\_F  
 lcd\_G  
 lcd\_dot

#### anody LCD:

lcd\_3  
 lcd\_2  
 lcd\_1  
 lcd\_0

**B)** Rozwiń poprzedni układ poprzez dodanie drugiego zestawu czterech przycisków i obie liczby podaj na wejścia sumatora (add4). Wynik sumowania prześlij do dekodera

Zauważ, że blok sumatora ma odwróconą kolejność wejść i wyjść (patrząc od góry do dołu są numerowane 0->3), upewnij się, że połączyłeś ze sobą właściwe końcówki

**Wejście:**

Dodać:

przełącznik 5: sw\_4

przełącznik 6: sw\_5

przełącznik 7: sw\_6

przełącznik 8: sw\_7

---

Dodatek: pliki do niektórych ćwiczeń znajdują się w katalogu:

[www.ue.eti.pg.gda.pl/~bpa/aeti/pliki](http://www.ue.eti.pg.gda.pl/~bpa/aeti/pliki)