


[Strona główna](#)

Katedra Systemów Mikroelektronicznych, Wydział Elektroniki,  
Telekomunikacji i Informatyki, Politechnika Gdańska


[English version](#)

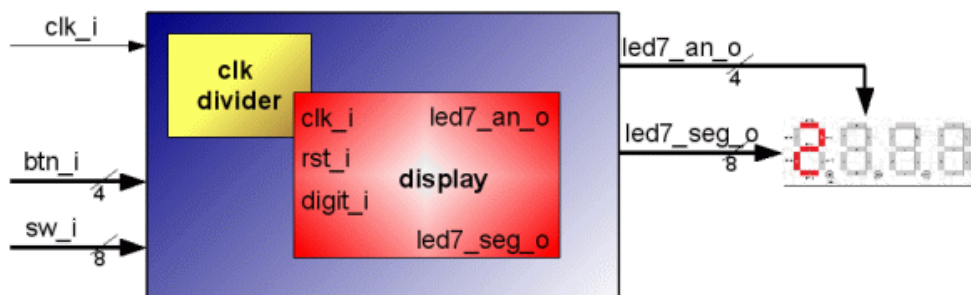
## Obsługa wyświetlacza LED

**Zadanie:** Głównym celem ćwiczenia jest opracowanie podbloku sterującego czterema cyframi wyświetlacza LED do wykorzystania w następnych ćwiczeniach.

Do sprawdzenia działania tego bloku należy go osadzić w układzie umożliwiającym ustawienie (za pomocą przycisków i przełączników) oraz wyświetlenie na wyświetlaczu LED dowolnej 4-znakowej liczby szesnastkowej wraz z przecinkami.



Rys. 1 Zdjęcie płytki Digilent Spartan 3

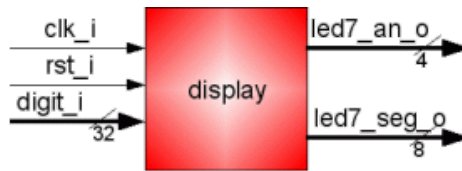


Rys. 2 Schemat blokowy całego układu

Dla przetestowania bloku należy dodać układy zapewniające następujące działanie:

- 1) Przełącznikami **SW3, SW2, SW1, SW0** ustawiamy 4-bitową wartość.
- 2) Naciśnięcie przycisku **BTN3** spowoduje zapalenie na cyfrze **AN3** wyświetlacza aktualnie ustawionej wartości na przełącznikach **SW3-SW0**. Wartość należy wyświetlić w postaci szesnastkowej. Takie samo działanie należy przypisać do pozostałych przycisków, tj. odpowiednio: naciśnięcie **BTN2** powoduje zapalenie cyfry **AN2**, **BTN1** -> **AN1** oraz **BTN0** -> **AN0**. Po naciśnięciu przycisku, znak na wyświetlaczu powinien się palić trwale, aż do wpisania nowej wartości.
- 3) Załączenie przełącznika **SW7** powinno spowodować zapalenie kropki dziesiętnej **DP** wyświetlacza **AN3**. Wyłączenie przełącznika powinno spowodować zgaszenie kropki **DP** tego wyświetlacza. Identyczne działanie dotyczy także **SW6** -> **DP AN2**, **SW5** -> **DP AN1**, **SW4** -> **DP AN0**.

Podstawowy blok sterowania wyświetlaczem powinien mieć następujący interfejs (blok ten przyda się w następnych ćwiczeniach):

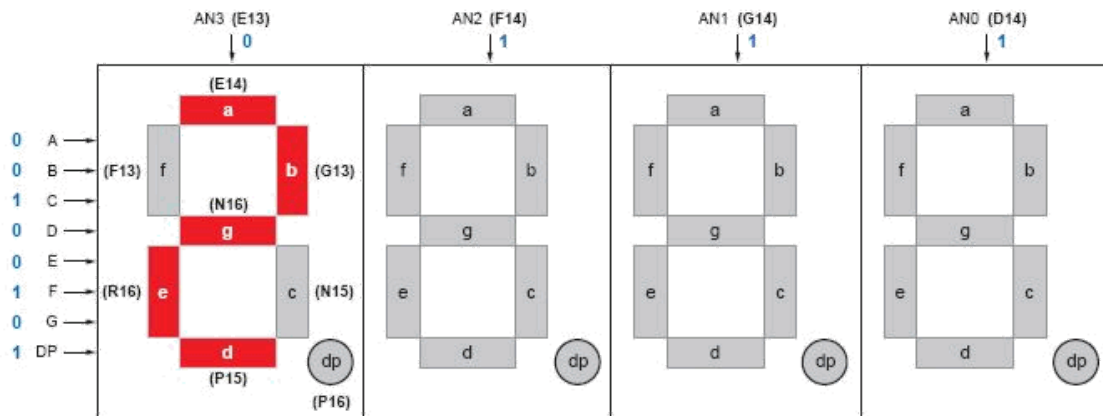


Rys. 3 Interfejs bloku sterowania wyświetlaczem

Znaczenie portów:

- **clk\_i** - na to wejście będzie podawany zegar o częstotliwości 1kHz.
- **rst\_i** - reset asynchroniczny aktywny stanem wysokim '1'. Po wystąpieniu sygnału reset, powinny się zapalić wszystkie segmenty wyświetlacza (nie wykorzystane w tym projekcie).
- **digit\_i** - 32-bitowe wejście, każdy bit odpowiada poszczególnemu segmentowi (8 najstarszych bitów dotyczy cyfry AN3, itd).

Wyświetlacz składa się z czterech 7-segmentowych wyświetlaczy LED. Odpowiednie segmenty poszczególnych wyświetlaczy są ze sobą połączone, więc trzeba zastosować multipleksowanie. Aby zapalić odpowiedni segment należy podać '0' zarówno na cyfrę (końcówka ANx) oraz na segment (A-G lub DP). Zalecana częstotliwość multipleksowania 1kHz. Do podziału częstotliwości wykorzystać dzielnik z poprzedniego zadania.



Rys. 4 Opis podłączeń poszczególnych segmentów wyświetlacza. Przykład wyświetlenia cyfry '2'

W układzie należy osadzić, jako *component*, odpowiednio zmodyfikowany dzielnik częstotliwości z poprzedniego ćwiczenia.

Napisać testbench. Jeśli do celów symulacji okaże się to niezbędne, można dodać do bloku głównego port wejściowy reset.

Przed implementacją wykonać symulację układu z wykorzystaniem pliku testbench.

#### Plik ucf do zadania, płytki Digilent Spartan-3, układ Spartan-3 3S200 FT256-4:

```
#####
# Clock:
NET "clk_i" LOC = "T9" ; # 50 MHz clock
#####

#####
# Seven-segment LED display:
NET "led7_an_o<3>" LOC = "E13" ; # leftmost digit, active low
NET "led7_an_o<2>" LOC = "F14" ; # active low
NET "led7_an_o<1>" LOC = "G14" ; # active low
NET "led7_an_o<0>" LOC = "d14" ; # rightmost digit, active low
#
NET "led7_seg_o<7>" LOC = "E14" ; # segment 'a', active low
NET "led7_seg_o<6>" LOC = "G13" ; # segment 'b', active low
NET "led7_seg_o<5>" LOC = "N15" ; # segment 'c', active low
NET "led7_seg_o<4>" LOC = "P15" ; # segment 'd', active low
NET "led7_seg_o<3>" LOC = "R16" ; # segment 'e', active low
NET "led7_seg_o<2>" LOC = "F13" ; # segment 'f', active low
NET "led7_seg_o<1>" LOC = "N16" ; # segment 'g', active low
NET "led7_seg_o<0>" LOC = "P16" ; # segment 'dp', active low
#####
#####
# Slide switches:
# sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0
```

```
NET "sw_i<0>" LOC = "F12" ; # active high when in UP position
NET "sw_i<1>" LOC = "G12" ; # active high when in UP position
NET "sw_i<2>" LOC = "H14" ; # active high when in UP position
NET "sw_i<3>" LOC = "H13" ; # active high when in UP position
NET "sw_i<4>" LOC = "J14" ; # active high when in UP position
NET "sw_i<5>" LOC = "J13" ; # active high when in UP position
NET "sw_i<6>" LOC = "K14" ; # active high when in UP position
NET "sw_i<7>" LOC = "K13" ; # active high when in UP position
#####

#####
# Push-buttons:
# @ @ @ @
# btn3 btn2 btn1 btn0
NET "btn_i<0>" LOC = "M13" ; # active high
NET "btn_i<1>" LOC = "M14" ; # active high
NET "btn_i<2>" LOC = "L13" ; # active high
NET "btn_i<3>" LOC = "L14" ; # active high
#####
```