

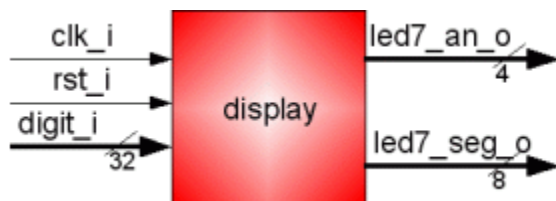
Sterowanie wyświetlaczem LED

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 kropkami. Następnie należy wykonać symulację funkcjonalną oraz zweryfikować układ praktycznie poprzez zaprogramowanie płytki testowej.

W użyciu jest prawa połowa (4 cyfry) wyświetlacza 7-segmentowego (lewa połowa może także coś wyświetlać – ignorujemy to).

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



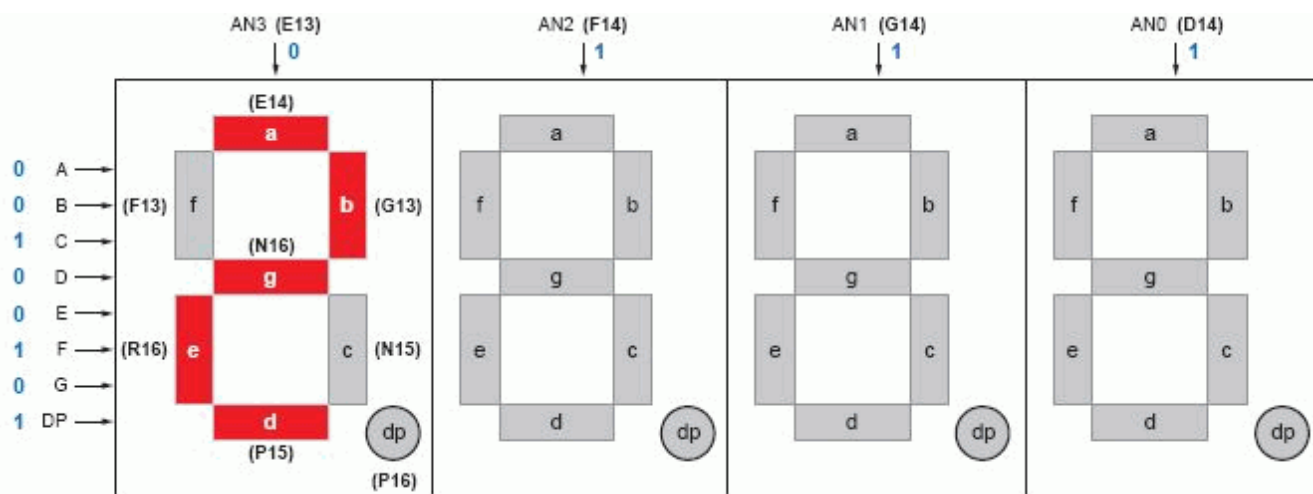
Rys. 1 Interfejs bloku sterowania wyświetlaczem.

Znaczenie portów:

- **clk_i** - na to wejście będzie podawany zegar o częstotliwości 100 MHz.
- **rst_i** - reset asynchroniczny aktywny stanem wysokim '1'. Po wystąpieniu sygnału **rst_i**, powinny się zapalić wszystkie segmenty wyświetlacza.
- **digit_i** - 32-bitowe wejście, każdy bit odpowiada poszczególnemu segmentowi (8 najstarszych bitów dotyczy cyfry AN3, itd).
- **led7_an_o** – wyjście sterujące anodami wyświetlaczy (określa który wyświetlacz się świeci).
- **led7_seg_o** – wyjście sterujące segmentami wyświetlaczy (określa świecące się segmenty).

Wyświetlacz składa się z czterech 7-segmentowych wyświetlaczy LED.

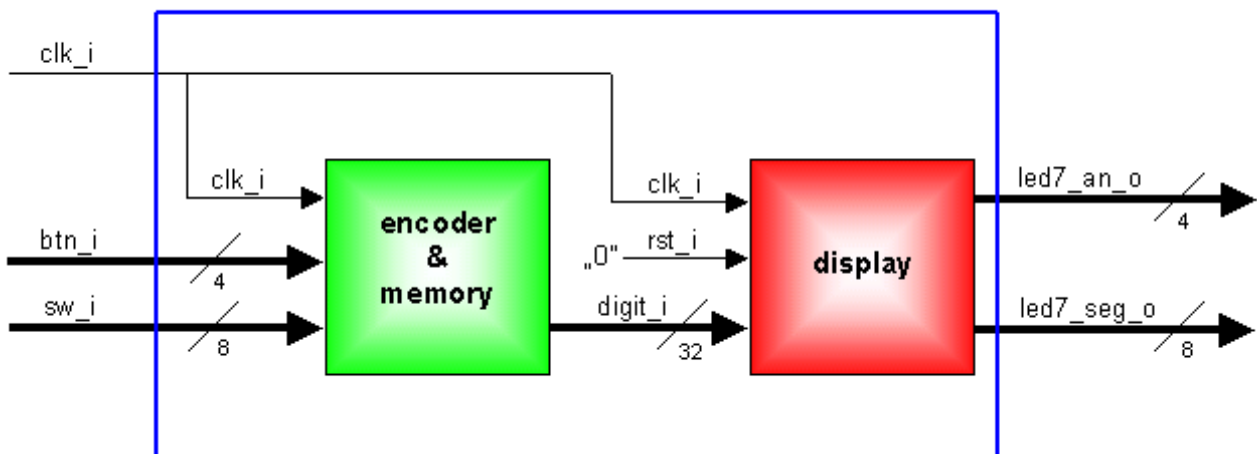
Odpowiednie segmenty poszczególnych wyświetlaczy są ze sobą połączone, więc trzeba zastosować multipleksowanie. Aby zapalić odpowiedni segment należy podać '0' zarówno na cyfrę (końcówka AN~~x~~) jak i na segment (A-G lub DP). Wymagana częstotliwość multipleksowania wynosi 1 kHz (czyli co 1 ms wyświetlacz przełącza się na kolejną cyfrę). Współczynnik podziału systemowej częstotliwości zegarowej (100 MHz) dający w efekcie częstotliwość multipleksowania (1 kHz) powinien być możliwym do zmiany parametrem komponentu display.



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

Następnie należy zrealizować drugi komponent - obsługujący enkoder i pamięć wyświetlacza. Służy on do przetestowania bloku display i powinien zapewniać następujące działanie:

- 1) Przełącznikami **SW3**, **SW2**, **SW1**, **SW0** ustawiamy 4-bitową wartość.
- 2) Naciśnięcie przycisku **BTNL** (w trybie zdalnym **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 **BTNC** (w trybie zdalnym **BTN2**) powoduje zapalenie cyfry **AN2**, **BTNR**(**BTN1**) → **AN1** oraz **BTND**(**BTN0**) → **AN0**. Po naciśnięciu i zwolnieniu 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**. Działanie tej funkcji ma być niezależne od stanu przycisków **BTNx**.



Rys. 3 Schemat blokowy całego układu.

Minimalne wymagania dotyczące symulacji: wykonać symulację naciskania i zwalniania kolejnych przycisków BTN_x. Czas trzymania przycisku powinien wynosić 1ms, natomiast czas zwolnienia 2 ms. W środku przedziału czasu pomiędzy zwolnieniem i naciśnięciem przycisku zmienić stan przełączników SW3-0 na dowolną, ale inną od poprzednich cyfrę heksadecymalną. Stan przełączników SW7-4 ustawić na początku tak, aby świeciły się 2 dowolnie wybrane kropki. Stan tych przełączników należy minimum 2 razy zmienić na inny (w trakcie zmiany przyciski BTN_x powinny być zwolnione). Należy pozwolić pracować symulacji tak długo, aby widoczny był proces wyświetlania się wszystkich wprowadzonych cyfr i zmiany stanu kropek. Przykład tej symulacji został także przedstawiony w formie przebiegów pod koniec prezentacji: [Demonstracja VIVADO](#)

Minimalne wymagania dotyczące weryfikacji na płytce: wprowadzić na wyświetlacz LED numer indeksu zaliczającego studenta (w przypadku grupy – jednego ze studentów z grupy) w postaci heksadecymalnej, przy czym 4 najstarsze cyfry wprowadzić na kolejne wyświetlacze 7-segmentowe, a najmłodszą cyfrę przedstawić za pomocą kropek (od lewej strony najstarszy bit, kropka zapalona – oznacza 1).

Fragment głównego pliku projektowego VHDL z deklaracją sygnałów:

```

entity top is
  Port ( clk_i : in STD_LOGIC;
        btn_i : in STD_LOGIC_VECTOR (3 downto 0);
        sw_i : in STD_LOGIC_VECTOR (7 downto 0);
        led7_an_o : out STD_LOGIC_VECTOR (3 downto 0);
        led7_seg_o : out STD_LOGIC_VECTOR (7 downto 0));
end top;
  
```

Uwaga:

btn_i(3) – przycisk BTNL (BTN3)

btn_i(2) – przycisk BTNC (BTN2)

btn_i(1) – przycisk BTNR (BTN1)

btn_i(0) – przycisk BTND (BTN0)

Plik z ograniczeniami projektowymi dla płytki Nexys-A7 (układ FPGA xc7a100tcsg324-1): [iup6s.xdc](#)

Wersja pliku z ograniczeniami projektowymi przeznaczona dla trybu zdalnego (obsługa przycisków i przełączników wirtualnych): [iup6z.xdc](#)