

Laboratorium #II-2 Rysowanie obiektów na ekranie VGA

- Obsługa ekranu VGA, rysowanie obiektów, tworzenie interfejsów.

Używane elementy: Basys3

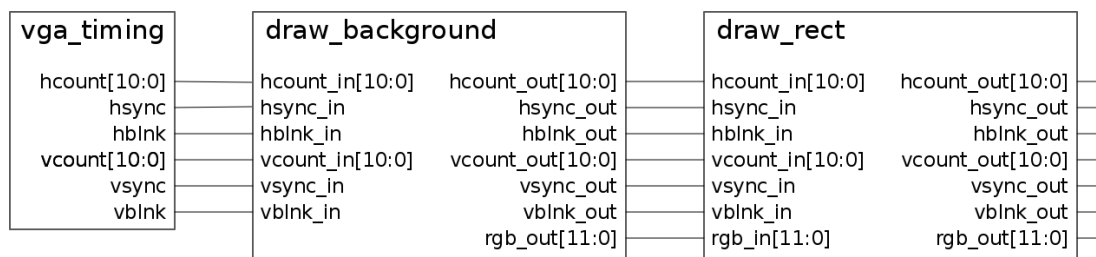
Temat ćwiczenia

Zadaniem do wykonania w ramach ćwiczenia jest napisanie modułu wyświetlającego prostokąt na ekranie VGA.

W poprzednim ćwiczeniu zaprojektowałeś układ generujący przebiegi sterujące do ekranu VGA oraz zmodyfikowałeś moduł rysujący tło. W tym ćwiczeniu dołączysz do nich kolejny moduł, rysujący na ekranie prostokąt. Następnie przygotujesz interfejs do łączenia ze sobą modułów VGA.

Przebieg ćwiczenia

1. Twoim zadaniem jest dodanie kolejnego modułu (**draw_rect**), który będzie rysował na ekranie prostokąt o zadanych kolorze. Wejścia, wyjścia i podłączenie pokazano poniżej.



Moduł w podobny sposób jak wcześniejszy opóźnia sygnały **hcount_out**, **hsync_out**, **hblink_out**, **vcount_out**, **vsync_out** i **vblink_out** o jeden takt zegara. Są one dokładnie tak samo kodowane w języku SystemVerilog. Jedyna różnica polega na generacji sygnału **rgb_out**. Moduł ma nadpisywać wartość koloru, jeżeli pozycja aktualnego piksela wskazywana przez liczniki **hcount_in** i **vcount_in** jest wewnątrz interesującego nas prostokąta; w przeciwnym razie **rgb_out** jest po prostu opóźnionym o takt zegara sygnałem **rgb_in**. Podobnie jak we wcześniejszym przypadku, wszystkie wyjścia mają być rejestrowane.

Moduł ma za zadanie narysowanie na ekranie prostokąta o zadanej pozycji, rozmiarach i kolorze. W związku z tym jako lokalne parametry mają być w module zdefiniowane:

- pozycja X prostokąta,
- pozycja Y prostokąta,
- szerokość prostokąta
- wysokość prostokąta
- kolor, którym prostokąt ma być rysowany.

2. Jak zauważyłeś, bloki rysujące obiekty na ekranie posiadają analogiczne zestawy 7 wejść i 7 wyjść. Wiąże się to z koniecznością tworzenia w module **top_vga** 7 połączeń pomiędzy każdymi modułami. Aby to uprościć, powtarzający się schemat połączeń można zamknąć w interfejsie.

W nowym pliku utwórz zatem interfejs **vga_if**, w którym zdefiniujesz 7 sygnałów: **vcount**, **vsync**, **vblink**, **hcount**, **hsync**, **hblink**, **rgb** oraz przy pomocy konstrukcji **modport** zdefiniujesz kierunki **in** oraz **out**.

W modułach **draw_bg** oraz **draw_rect** zastąp 7 sygnałów vga nowym interfejsem. Dostosuj również moduł **top_vga**.

Wyniki ćwiczenia

Jako wynik ćwiczenia należy:

- zaprezentować działanie programu na następnych zajęciach laboratoryjnych i wyjaśnić słownie zasadę działania. Program powinien wyświetlać na ekranie obrazek z wcześniejszego ćwiczenia oraz dodatkowo dorysowany na nim prostokąt.

- załadować spakowane (ZIP) archiwum projektu na UPEL.

Archiwum powinno zawierać wszystkie pliki źródłowe i skrypty, potrzebne do wygenerowania bitstreamu i przeprowadzenia symulacji. Dodatkowo, w folderze results, powinny się znaleźć: bitstream, warning_summary oraz obrazek tiff wygenerowany przez testbench głównego modułu. Pozostałe pliki (w tym również foldery i pliki ukryte, za wyjątkiem pliku .gitignore) powinny zostać usunięte.

Projekt powinien być napisany zgodnie z zasadami opisanymi w pliku „Zasady modelowania w języku SystemVerilog pod kątem syntezy”, dostępnym na UPEL. Inne style kodowania mogą być stosowane wyłącznie po podaniu źródła.

Nie załadowanie projektu w terminie podstawowym = -1 pkt do oceny

Nie załadowanie projektu w terminie ostatecznym = 0 pkt za ćwiczenie