

Laboratorium #II-1. Rysowanie obiektów na ekranie VGA.

- Obsługa ekranu VGA, rysowanie obiektów.

Używane elementy: Basys3

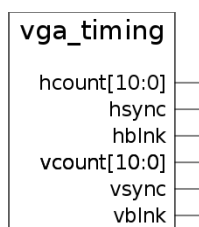
Temat ćwiczenia.

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

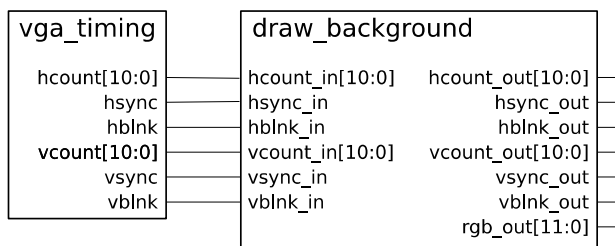
W poprzednim ćwiczeniu zaprojektowałeś układ generujący przebiegi sterujące do ekranu VGA. W tym ćwiczeniu uporządkujesz swój kod, oraz dodasz nowy moduł, dorysowujący na ekranie prostokąt.

Przebieg ćwiczenia

1. W poprzednim ćwiczeniu napisałeś zawartość modułu **vga_timing**, który generował na wyjściu 6 sygnałów sterujących do monitora VGA:



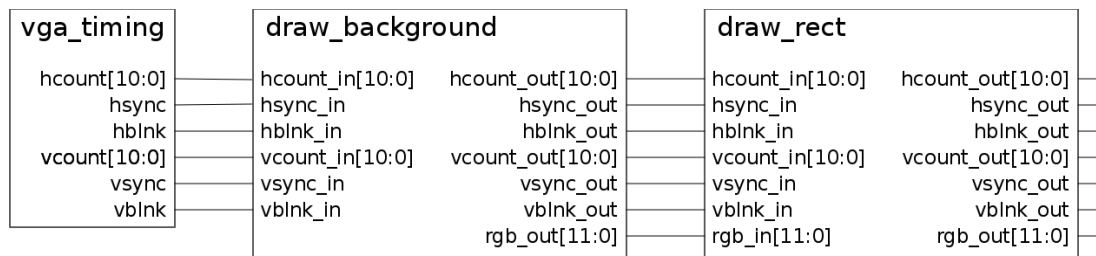
Jak zauważyłeś, bloki proceduralne **always @...** rysujące zawartość ekranu były umieszczone razem z modulem **vga_timing**, w tym samym module **vga_example**. Twoim zadaniem w tym punkcie jest zamknięcie wszystkich bloków proceduralnych rysujących tło, kolorową ramkę, oraz inicjały w jednym wspólnym module **draw_background**, który będzie miał wejścia i wyjścia jak pokazano na schemacie poniżej, i będzie podłączony do modułu **vga_timing**. Na schemacie nie uwzględniono sygnału zegara i resetu.



W module **draw_background** sygnały: **hcount_out**, **hsync_out**, **hblnk_out**, **vcount_out**, **vsync_out** i **vblnk_out** są opóźnionymi o jeden takt zegara odpowiednimi wejściami ***_in**, natomiast **rgb_out[11:0]** łączy w sobie 4-bitowe kolory red-green-blue. Wszystkie te sygnały mają być rejestrowane (są bezpośrednimi wyjściami z rejestrów). Pamiętaj o odpowiednim podłączeniu w module **vga_example** sygnałów **vs**, **hs**, **r**, **g**, **b**.

Po wprowadzeniu tych zmian układ powinien generować taki sam obrazek w symulacji, jak wcześniej, oraz wyświetlać na ekranie ten sam obraz. Nie przechodź do dalszej części ćwiczenia jeżeli nie uzyskasz takich wyników.

2. Drugim zadaniem będzie dodanie kolejnego modułu (**draw_rect**), którego zadaniem będzie dorysowanie na ekranie prostokąta o zadanym 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 tak zegara. Są one dokładnie tak samo kodowane w języku Verilog. 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.

Wyniki ćwiczenia

Jako wynik ćwiczenia należy:

- załadować projekt na UPEL (bez katalogu symulacyjnego, ale z obrazkami tiff generowanymi przez testbench).
- 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.

Kod projektowanego modułu powinien być napisany zgodnie z zasadami opisanymi w pliku „Zasady pisania kodu w języku Verilog”, dostępnego na UPEL.

Wyniki będą ocenione bezpośrednio na kolejnych zajęciach.