

Laboratorium #II-3. Poruszanie obiektami na ekranie VGA.

- Generacja zsynchronizowanych sygnałów zegarowych.
- Podłączenie modułu napisanego w języku VHDL do projektu w języku Verilog.
- Obsługa myszy PS/2.

Używane elementy: Basys3

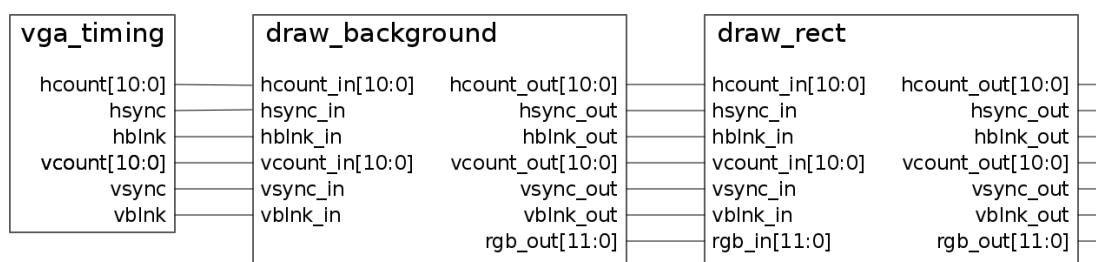
Temat ćwiczenia.

Zadanie do wykonania w ramach ćwiczenia to opracowanie programu, który będzie na ekranie wyświetlał prostokąt poruszany myszą.

W poprzednim ćwiczeniu zaprojektowałeś układ rysujący na ekranie prostokąt. W tym ćwiczeniu rozbudujesz opracowany wcześniej układ tak, aby umożliwić dynamiczne sterowanie pozycją prostokąta na ekranie.

Przebieg ćwiczenia

W poprzednim ćwiczeniu opracowałeś układ, który wyświetlał na ekranie prostokąt na podanym tle. Uproszczony Schemat tego układu wyglądał następująco:



W tym ćwiczeniu dodamy do schematu jeszcze dwa moduły:

- moduł sterujący położeniem rysowanego prostokąta
- moduł obsługujący mysz.

Aby umożliwić dodanie tych dwóch modułów musimy jeszcze wcześniej wymienić na nowy generator sygnału zegara, ponieważ system wyświetlania na ekranie działa z zegarem 40 MHz, natomiast dostępny kontroler myszy działa z zegarem 100 MHz. Drugą rzecz, którą musimy zrobić, to zmodyfikować moduł rysowania prostokąta tak, aby można było mu podawać pozycję prostokąta z zewnątrz.

1. Wymiana modułu generatora zegara.

Zrób kopię zapasową projektu przed wprowadzaniem modyfikacji, jeżeli takiej nie posiadasz.

W projekcie, w pliku **VGA_example.v** znajduje się moduł **MMCME2_BASE** (clk_in_mmcem2) wraz buforami **IBUF** (clk_ibuf), **BUFH** (clk_out_bufh), **BUFGCE** (clk_out_bufgce) oraz **ODDR** (pclk_oddr).

Wszystkie wymienione moduły z wyjątkiem ostatniego (**ODDR**, który służy do wygenerowania kopii sygnału zegara na wyjściu FPGA) zostaną zastąpione przez nowy moduł.

Opisy wymienionych modułów znajdziesz w dokumencie **Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs** (dostępny w dokumentacji Vivado oraz na [www](http://www.xilinx.com)).

1.1. Zakomentuj komentarzem blokowym cztery wymienione moduły.

1.2. Otwórz **IP Catalog** (Project Manager → IP Catalog), wyszukaj **Clocking Wizard** i uruchom go. Wykorzystaj moduł **MMCM** i wygeneruj dwa sygnały zegarowe o nazwach **clk100MHz** i **clk40MHz** i odpowiednich częstotliwościach. Włącz opcję **Safe Clock Startup**. Nazwę sygnału wejściowego ustaw jako **clk**. Zwróć uwagę na to, aby bufor wyjściowe **Drives** dla obydwu sygnałów były tego samego typu **BUFG**.

Po kliknięciu „OK” wybierz „Synthesis Option” **Global**.

1.3. Przeglądnij wygenerowany kod (dwa moduły) i podłącz odpowiedni moduł do pozostałych w module **VGA_example**. Pamiętaj o poprawnym podłączeniu bufora **ODDR**.

1.4. W pliku XDC zakomentuj dotychczasową definicję sygnału zegara **create_clock**. Nie będzie ona potrzebna, ponieważ wygenerowany moduł zawiera już w sobie plik XDC z tą definicją.

1.5. Puść symulację. Wynik powinien być identyczny jak poprzednio.

1.6. Zaprogramuj płytkę i sprawdź, czy układ działa poprawnie.

2. Kontrola pozycji prostokąta przy pomocy myszy.

Zrób kopię zapasową projektu przed wprowadzaniem kolejnych modyfikacji, jeżeli takiej nie posiadasz.

2.1. Dodaj do wejść modułu **draw_rect** dwa wejścia określające pozycję prostokąta. Jeżeli poprawnie zdefiniowałeś parametry w poprzednim ćwiczeniu, to będzie to polegało na zamianie nazw parametrów pisanych dużymi literami na małe oraz dopisaniu dwóch definicji portów wejściowych.

2.2. Dodaj do projektu dwa pliki źródłowe dostępne na UPEL'u łącznie z instrukcją do ćwiczenia: **MouseCtl.vhd** i **Ps2Interface.vhd**. Pliki te zawierają napisany w języku VHDL kontroler obsługujący mysz w standardzie PS/2 i pochodzą z projektu demonstracyjnego dla płytki BASYS3 dostępnego na stronach www firmy Xilinx.

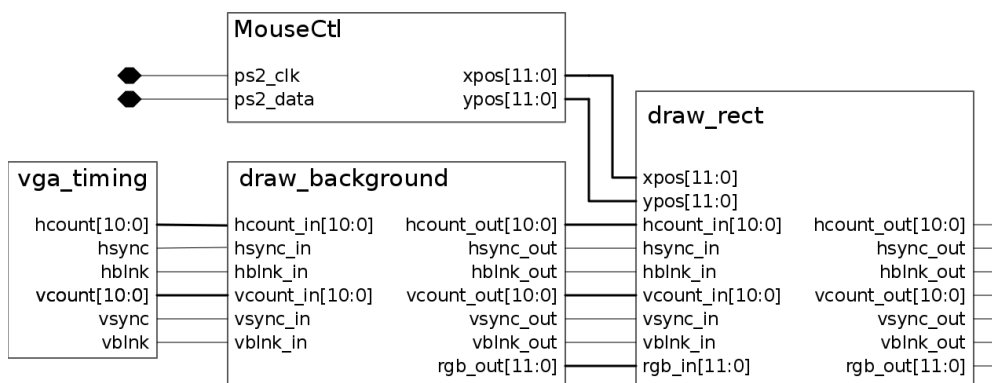
2.3. Przeczytaj nagłówek pliku **MouseCtl.vhd**. Zawiera on opis zaimplementowanego protokołu i działania modułu.

2.4. Dodaj do modułu **VGA_example** porty **ps2_clk** i **ps2_data**. Uwaga: kierunek portów to **inout**.

2.5. Dodaj do pliku XDC definicje podłączeń dla **ps2_clk** (pin **C17**) oraz **ps2_data** (pin **B17**), ze standardową konfiguracją **LVCMOS33**. Aby włączyć rezystory podciągające dla tych wyjść dopisz do pliku XDC następujące linijki:

```
set_property PULLUP true [get_ports ps2_clk]
set_property PULLUP true [get_ports ps2_data]
```

2.6. Dodaj do modułu **VGA_example** moduł **MouseCtl** i podłącz go zgodnie ze schematem poniżej.



Zwróć uwagę, że moduł **MouseCtl** powinien być podłączony do zegara 100 MHz, a pozostałe do 40 MHz. Czy spodziewasz się z tego powodu jakichś problemów?

2.7. Skompiluj i uruchom na płytce program. Powinien on wyświetlać zdefiniowany przez Ciebie prostokąt poruszający się po ekranie zgodnie z ruchami myszy.

2.8. Sprawdź ostrzeżenia w zakładce „Messages” - czy wszystkie rozumiesz?

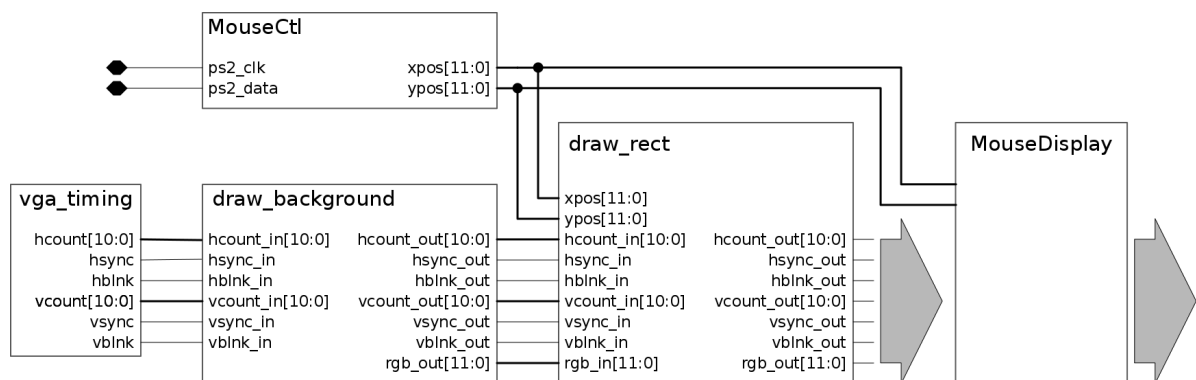
3. Wyświetlanie kursora myszy.

Zrób kopię zapasową projektu przed wprowadzaniem kolejnych modyfikacji, jeżeli takiej nie posiadasz.

3. 1. Dodaj do źródeł plik **MouseDisplay.vhd** udostępniony razem z instrukcją do ćwiczenia. Jest to moduł udostępniany przez firmę Xilinx w projekcie demonstracyjnym dla płytki BASYS3, napisany w języku VHDL. Wykorzystasz go do narysowania na ekranie kursora myszy.

3.2. Przejrzyj plik. Zwróć uwagę, że linie wejściowe RGB i blank są zakomentowane. Usuń w pliku komentarze w odpowiednich miejscach (nie tylko w definicjach portów, ale również w kodzie opisującym stan wyjść).

3.2. Dodaj moduł **MouseDisplay** zgodnie ze schematem poniżej.



Dostosuj odpowiednio wejścia i wyjścia, aby uzyskać zdefiniowaną funkcjonalność. Którego z sygnałów zegara użyjesz w tym module?

3.3. Skompiluj i uruchom na płytce program. Powinien on wyświetlać zdefiniowany przez Ciebie prostokąt poruszający się po ekranie zgodnie z ruchami myszy, z nadpisany kursorem myszy.

3.4. Sprawdź ostrzeżenia w zakładce „Messages” - czy wszystkie rozumiesz?

Wyniki ćwiczenia

Jako wynik ćwiczenia należy:

- załadować projekt na UPEL (bez katalogu symulacyjnego i implementacyjnego).

ZA BRAK ZAŁADOWANEGO PROJEKTU NA UPEL OTRZYMUJEMY -1 pkt. DO OCENY.

- 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.

Każdy zespół będzie odpytany z protokołu PS/2.

UWAGA: Kod projektowanego modułu powinien być napisany zgodnie z zasadami opisanymi w pliku „Zasady pisanie kodu w języku Verilog”, dostępnego na UPEL. Będzie to oceniane!

Wyniki będą ocenione bezpośrednio na kolejnych zajęciach.