

PRZEBIEG ĆWICZENIA

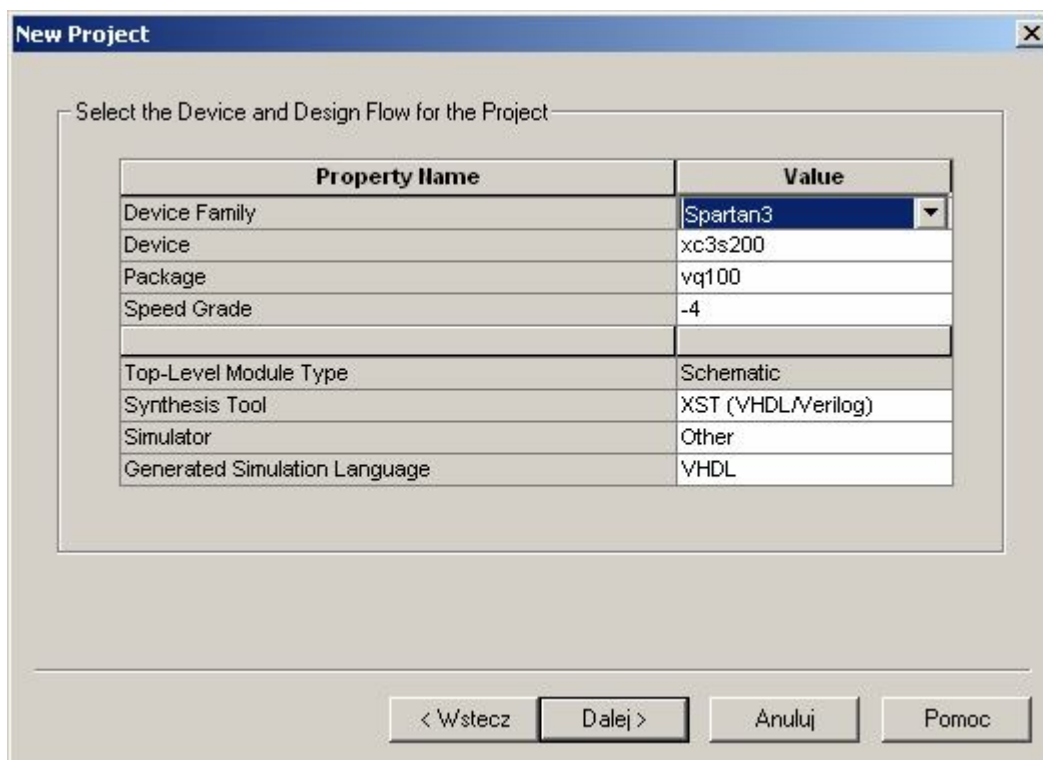
CZĘŚĆ A – projekt układu.

1. Zaprojektować (na papierze) zadany przez prowadzącego licznik asynchroniczny na zadanym typie przerzutników i zamieścić go w protokole.

CZĘŚĆ B – implementacja projektu w strukturze FPGA

(UWAGA: kolejne podpunkty należy realizować bardzo uważnie, ponieważ błędy popełnione na tym etapie mogą być później nie do cofnięcia i trzeba będzie zaczynać od początku).

2. Na podstawie stworzonego w punkcie 1 projektu:
 - 2.1. Uruchomić aplikację **Xilinx - Project Navigator**. Jest to aplikacja firmy Xilinx umożliwiająca projektowanie, symulację i implementację układów logicznych w strukturach układów FPGA.
 - 2.2. Stworzyć nowy projekt (**File -> New Project**). Nadać mu nazwę (**w nazwach NIE używać spacji**) (**Project Name**), wskazać lokalizację (można użyć domyślnej) i wybrać sposób realizacji projektu (**Top-Level Module Type**) – **Schematic**.
 - 2.3. W następnym oknie wybrać rodzaj programowanego układu logicznego. Powinien być to układ wykorzystywany na ćwiczeniu, zgodnie ze screenem (wszystkie widoczne informacje są dostępne na obudowie układu FPGA):



- 2.4. W kolejnym oknie stworzyć nowe źródło (**New Source**) do projektu. Nadać mu nazwę (**File Name**) i wybrać typ – **Schematic**.
- 2.5. Dwa kolejne okna można przeklikać bez zmian. Na tym kończy się wstępna konfiguracja projektu. Na ekranie powinno pojawić się okno projektowania układu. W tym oknie budujemy swój projekt.

2.5.1. Najpierw należy wstawić potrzebne przerzutniki, z menu u góry wybrać pozycję **Add Symbol**, po lewej pojawi się lista dostępnych kategorii, stamtąd wybrać pozycję **Flip_Flop**, a wtedy na dole pojawi się lista dostępnych przerzutników, z których należy wybrać i wstawić do schematu te potrzebne do projektu.

Będą to:

- w przypadku przerzutników typu T – oznaczenie **FTC**
- w przypadku przerzutników typu D – oznaczenie **FDC**
- w przypadku przerzutników typu JK – oznaczenie **FJKC**

Logiczne zero i jedynkę znajdziemy w podmenu **General** – jako **GND** (poziom zera) i **VCC** (poziom jedynki).

Jeżeli dodatkowo potrzebne będą bramki, to można je znaleźć w podmenu **Logic** (najprawdopodobniej potrzebna będzie co najmniej jedna negacja (bramka NOT) oznaczona jako **INV**).

2.5.2. Po wstawieniu wszystkich potrzebnych przerzutników, poziomu zera i jedynki oraz ewentualnie bramek, z menu należy wybrać pozycję **Add I/O Marker** i wstawić wejścia i wyjścia układu (w przypadku przykładowego projektu będą to 2 wejścia (**Add an input marker**) i trzy wyjścia (**Add an output marker**)).

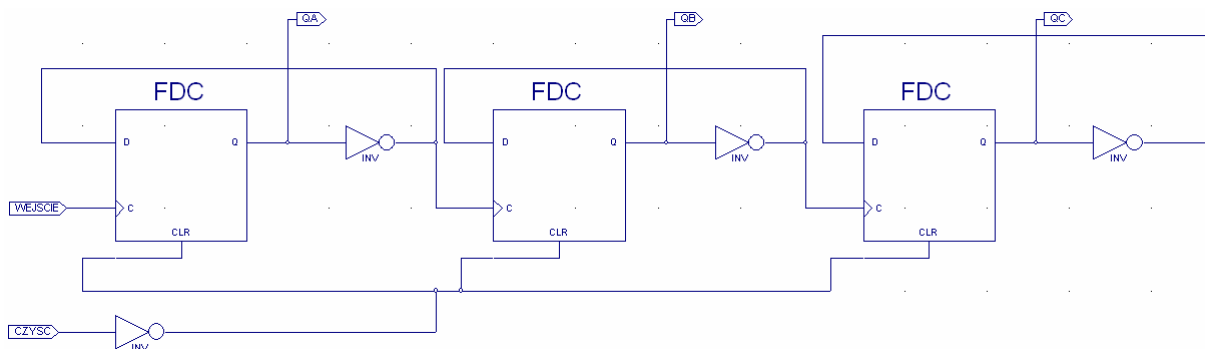
UWAGA: Kolejność jest ważna, NAJPIERW należy wstawiać wejścia/wyjścia w odpowiedniej ilości, a dopiero potem wykonywać połączenia. Wejścia/wyjścia da się wstawiać tylko przy samych wejściach/wyjściach układów, dopiero potem można je przesuwać.

W przypadku projektu realizowanego na laboratorium wejścia też będą dwa, liczba wyjść zależy od liczby bitów licznika (tożsamej z liczbą wykorzystywanych przerzutników).

Nazwy wejść/wyjść można zmieniać dwuklikiem.

2.5.3. Gdy wstawione będą już elementy, wejścia i wyjścia, można wykonać wszystkie potrzebne połączenia (**Add Wire**). Gotowy układ warto jeszcze sprawdzić pod względem formalnym (**Check Schematic**).

Ostateczny wygląd projektu realizującego pokazany wcześniej licznik do 7 na przerzutnikach typu D pokazany jest poniżej. Jak łatwo zauważyć, w oprogramowaniu firmy Xilinx nie występują wyjście negowane z przerzutników i w związku z tym należy je samemu negować bramkami NOT.



Ponadto, wejścia zerujące przerzutników również zostały wyprowadzone na zewnątrz (przez negację). W ten sposób użytkownik będzie miał możliwość np. ręcznego (przyciskiem) wyzerowania układu.

- 2.5.4. Następnym krokiem jest przypisanie wejść i wyjść wykonanego układu do nóżek układu scalonego. W tym celu należy wrócić do okna **Xilinx - Project Navigator** i w oknie po lewej stronie dwukliknąć w opcję **Assign Package Pins**. Powinno otworzyć się okno Xilinx PACE i topologia wykorzystywanego układu FPGA. Po lewej stronie (**Design Object List**) widać wcześniej zaprojektowane wejścia/wyjścia układu.

Nóżki układu przypisać zgodnie z załączoną notą wykorzystywanego w ćwiczeniu zestawu ZL6PLD.

Jako wyjścia ustawić wybrane diody LED (strona 5 noty) – najlepiej po kolei, czyli niech dioda D1 będzie QA, dioda D2 - QB itd.

Jako wejście zerujące (CZYSC) wykorzystać wybrany przycisk klawiatury (strona 6 noty).

Jako wejście zegarowe (WEJSCIE) wykorzystać pin 14.

Piny najłatwiej przypisać wpisując tam bezpośrednio nóżkę układu scalonego, której chcemy przypisać daną funkcję, przykładowo, chcąc przypisać nóżce 14 układu scalonego funkcję wejścia zegarowego należy w kolumnie **Loc** wpisać (z klawiatury) P14 w odpowiednim wierszu (jak na rysunku poniżej). Tak samo należy postąpić przypisując pozostałe sygnały do odpowiednich pinów.

	I/O Name	I/O Direction	Loc	Bank	I/O Std.
	WEJSCIE	Input	P14	BANK	
	QC	Output			
	QB	Output			
	QA	Output			
	CZYSC	Input			

- 2.6. Po przypisaniu pinów można przystąpić do implementacji projektu w strukturze FPGA. W tym celu należy zgłosić prowadzącemu gotowość do zaprogramowania układu.

2.7. Jeżeli wszystko działa, podpisany protokół należy zostawić prowadzącemu i ćwiczenie jest zaliczone.

- 2.8. Ewentualnie nasuwające się wnioski/przemyślenia można zamieścić na odwrocie protokołu.