

Lista zadań nr 4

Zagadnienia

- stosowanie sieci Petriego (ang. *Petri net*) jako narzędzia do modelowania algorytmów sterowania procesami dyskretnymi,
- implementacja algorytmów sterowania w układach cyfrowych FPGA (*Field-Programmable Gate Array*).

Ścieżka projektowa, środowisko laboratoryjne, sprzęt

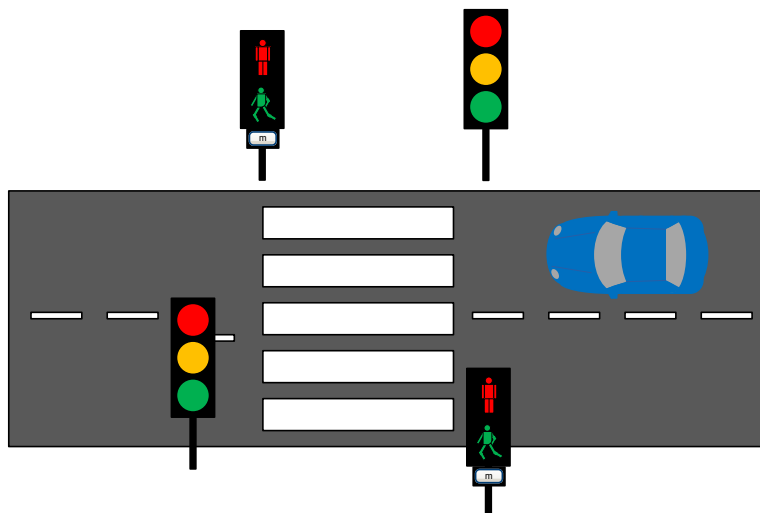
Realizując zadanie skorzystaj ze ścieżki projektowej opisanej w liście zadań nr 2. Środowisko laboratoryjne (*IOPT Tools, Xilinx ISE Design Suite*) oraz sprzęt (*Nexys4 DDR*) także jest opisany w liście zadań nr 2.

Zadanie 1

Wymagania

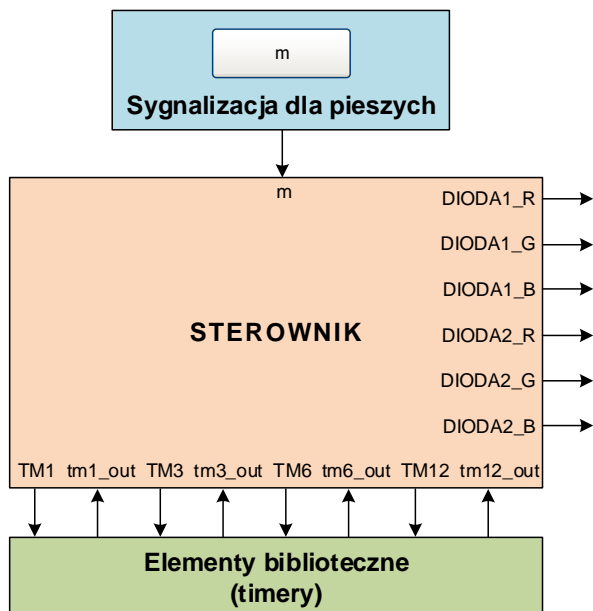
Celem projektu jest utworzenie systemu sterowania sygnalizacją świetlną przejścia dla pieszych (rys. 1). Przebieg sterowania jest następujący: w chwili początkowej świeci się światło zielone dla samochodów i czerwone dla pieszych. Po naciśnięciu przez pieszego przycisku (aktywny sygnał *m*) system uruchamia światło pomarańczowe dla samochodów, a po upływie 3 sek. światło czerwone. W tym samym momencie zapalane jest światło zielone dla pieszych. Po upływie 6 sekund światło zielone dla pieszych zaczyna migać (gaśnie na 1 sekundę a następnie świeci się przez 1 sekundę) i po upływie 3 sekund migania zapalane jest światło czerwone dla pieszych. Po upływie kolejnych 3 sekund zapalane jest światło pomarańczowe dla samochodów (pali się razem ze światłem czerwonym dla samochodów), a po kolejnych 3 sekundach zapalane jest światło zielone dla samochodów. Kolejne włączenie światła pomarańczowego dla samochodów (po wciśnięciu przycisku przez pieszego) nie może nastąpić szybciej niż po upływie 12 sekund.

W modelu użyj sygnałów wyjściowych sterujących dwoma diodami RGB (jedna symbolizować będzie światła dla samochodów a druga dla pieszych). Na przykład, aby dioda 1 zaświeciła się na kolor zielony wartości sygnałów sterujących powinny być następujące: $dioda1_r=0$ (red), $dioda1_g=1$ (green), $dioda1_b=0$ (blue). Analogicznie można uzyskać kolor czerwony: $dioda1_r=1$, $dioda1_g=0$, $dioda1_b=0$. Aby uzyskać kolor pomarańczowy (żółty) należy użyć następujących wartości sygnałów: $dioda1_r=1$, $dioda1_g=1$, $dioda1_b=0$. W przypadku, gdy na sygnalizatorze świetlnym dla samochodów powinno pojawić się światło pomarańczowe (żółte) i jednocześnie czerwone przyjmij, że wyświetlane będzie tylko światło pomarańczowe (żółte). Niełatwo zauważyć, że wszystkie niezbędne w projekcie kolory nie wymagają aktywnego sygnału odpowiedzialnego za kolor niebieski, dlatego zostały one w dalszym opisie pominięte.



Rys. 1

Upływ czasu w systemie realizowany jest w następujący sposób. Sterownik (rys. 2) korzysta z 4 timerów (układów odmierzających zadany czas). Aby uruchomić wybrany timer system musi aktywować (logiczna 1) wyjście odpowiedzialne za uruchomienie timera (np. *TM1*, *TM3*). Timer po uruchomieniu odmierza zadany czas i jego upływanie sygnalizuje aktywnym, odpowiednim sygnałem (wejściowym z punktu widzenia sterownika). Na przykład, aby uruchomić timer odmierzający 3 sekundy, należy uaktywnić wyjście *TM3*. Pojawienie się w systemie aktywnego sygnału *tm3_out* oznacza, że minęły 3 sekundy.



Rys. 2

Wszystkie sygnały występujące w układzie scharakteryzowano w tabeli 1.

Tabela 1. Opis sygnałów układu z zadania 1

Lp.	Nazwa sygnału	Rodzaj sygnału	Źródło	Znaczenie
1.	m	wejściowy	przycisk	Sygnał aktywny oznacza, że pieszy chce przejść na drugą stronę.
2.	DIODA1_R	wyjściowy	sterownik	Sygnał aktywny oznacza, że dioda 1 (RGB) będzie świecić kolorem, którego składową jest kolor czerwony.
3.	DIODA1_G	wyjściowy	sterownik	Sygnał aktywny oznacza, że dioda 1 (RGB) będzie świecić kolorem, którego składową jest kolor zielony.
4.	DIODA2_R	wyjściowy	sterownik	Sygnał aktywny oznacza, że dioda 2 (RGB) będzie świecić kolorem, którego składową jest kolor czerwony.
5.	DIODA2_G	wyjściowy	sterownik	Sygnał aktywny oznacza, że dioda 2 (RGB) będzie świecić kolorem, którego składową jest kolor zielony.
6.	TM1	wyjściowy	sterownik	Sygnał aktywny oznacza uruchomienie timera odmierzającego czas 1 sekundy.
7.	tm1_out	wejściowy	timer	Sygnał aktywny oznacza, że upłynął czas 1 sekundy od momentu uruchomienia timera.
8.	TM3	wyjściowy	sterownik	Sygnał aktywny oznacza uruchomienie timera odmierzającego czas 3 sekund.
9.	tm3_out	wejściowy	timer	Sygnał aktywny oznacza, że upłynął czas 3 sekund od momentu uruchomienia timera.
10.	TM6	wyjściowy	sterownik	Sygnał aktywny oznacza uruchomienie timera odmierzającego czas 6 sekund.
11.	tm6_out	wejściowy	timer	Sygnał aktywny oznacza, że upłynął czas 6 sekund od momentu uruchomienia timera.
12.	TM12	wyjściowy	sterownik	Sygnał aktywny oznacza uruchomienie timera odmierzającego czas 12 sekund.
13.	tm12_out	wejściowy	timer	Sygnał aktywny oznacza, że upłynął czas 12 sekund od momentu uruchomienia timera.

Krok 1 – Modelowanie

Korzystając ze środowiska *IOPT Tools* zamodeluj sieć Petriego przedstawiającą model sterowania sygnalizacją świetlną. Odpowiednio zidentyfikuj i nazwij poszczególne miejsca (np. SCPZ – *samochody światło czerwone*, *piesi światło zielone* itp.) i tranzycje. Użyj dokładnie takich nazw sygnałów jak w tabeli 1.

Krok 2 – Walidacja

Po zaprojektowaniu modelu układu sterowania, wykonaj symulację utworzonej sieci Petriego korzystając z symulatora *IOPT Tools*. Jeżeli wyniki symulacji wykażą, że model zawiera błędy, wróć do kroku pierwszego i popraw go. Jeżeli model przeszedł poprawnie etap symulacji, wyeksportuj jego opis w syntezowalnym języku opisu sprzętu VHDL.

Krok 3 – Synteza

Następnie korzystając z wytycznych prowadzącego utwórz projekt w środowisku *Xilinx ISE Design Suite 14.7*. Parametry projektu pokazano na rys. 3.

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Artix7
Device	XC7A100T
Package	CSG324
Speed	-1
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

Rys. 3

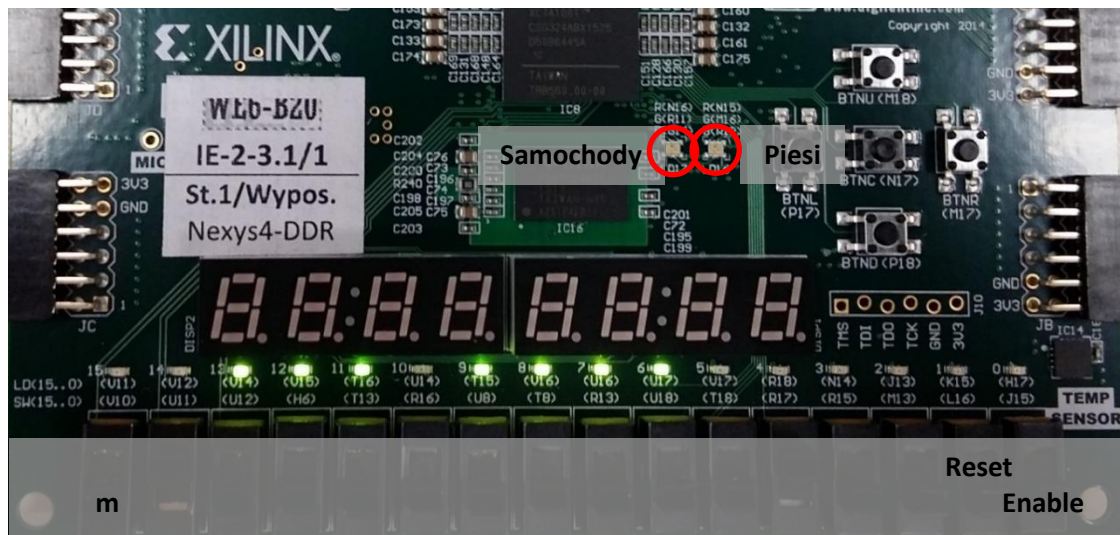
Następnie dodaj do projektu pliki *smart04_zad01_nexys4ddr.ucf* oraz *timers.vhdl* i zgodnie z wytycznymi prowadzącego dokonaj modyfikacji w pliku *nazwaTwojegoProjektu_main.vhdl* na podstawie pliku *smart04_main.vhdl* dostępnego na stronie z materiałami do zajęć (zakładka *SMART*). Kolejnym etapem jest uruchomienie procesu syntezy, implementacji oraz generowania pliku wynikowego (bitstream) zawierającego dane służące do zaprogramowania układu.

Krok 4 – Implementacja

Podłącz do komputera układ *Nexys 4 DDR* i korzystając z narzędzia *IMPACT* wykonaj programowanie układu FPGA za pomocą wygenerowanego w poprzednim kroku bitstreamu.

Krok 5 – Weryfikacja sprzętowa

Zweryfikuj działanie układu przełączając poszczególne przełączniki i obserwując diody RGB (rys. 4).



Rys. 4

W razie trudności lub wątpliwości – pytaj!