

LABORATORIUM Teoria Automatów	
Temat Ćwiczenia: WYKORZYSTANIE TECHNOLOGII FPGA DO PROJEKTOWANIA UKŁADÓW LOGICZNYCH	
Grupa laboratoryjna: 1a, wtorek 11⁰⁰	
L.p	Nazwisko i Imię
1	Aleksandrowicz Maciej
2	Krzyszczuk Michał
3	Marczewski Marcin
Data wykonania ćwiczenia : 07.11.2017r	

Spis treści

1) Wstęp teoretyczny	1
2) Zadanie do wykonania	2
3) Analiza teoretyczna	2
4) Sposób realizacji zadania w praktyce	3
5) Wnioski i obserwacje	6

1) Wstęp teoretyczny

W ćwiczeniu tym zapoznamy się ze sposobem projektowania oraz budowy układów cyfrowych z wykorzystaniem programowalnych układów logicznych typu FPGA firmy XILINX oraz poznamy dostępne środowiska umożliwiające realizację projektu.

Firma XILINX jest producentem sprzętu i oprogramowania umożliwiającego projektowanie i implementację programowalnych układów logicznych. Elastyczność samych układów oraz środowiska programowego umożliwia wykorzystanie różnorodnych technik projektowania.

W celu zdefiniowania zachowania układu FPGA używa się języka opisu sprzętu, takiego jak Verilog lub VHDL. Następnie przy pomocy narzędzi syntezy generuje się listę połączeń, która potem w procesie implementacji jest odwzorowywana w konkretnym układzie. Należy zwrócić uwagę, że proces syntezy dopuszcza tworzenie układów logicznych dowolnych rozmiarów, podczas gdy proces implementacji jest próbą wpisania go do konkretnej programowalnej macierzy bramek, gdzie może zabraknąć zasobów do realizacji zadanej logiki. Do zaprogramowania FPGA służy plik binarny, który zawiera informacje o konfiguracji układu.

Pakiet projektowy zawiera szereg modułów wykorzystywanych w poszczególnych etapach budowy projektu. Przejście od pomysłu do układu docelowego wymaga od użytkownika realizacji kilku etapów projektowania.

Po zdefiniowaniu projektu uzyskuje się plik charakterystyczny dla danego typu edytora (.wir dla Viewlogic, .sch dla ORCAD, .xise dla WebPack itp.). Następnie przy pomocy tzw. Design Manager'a projektant uzyskuje dostęp do kompilatora wraz z pewnymi interfejsami importowo-eksportowymi. Efektem końcowym tego etapu projektowania jest zbiór wynikowy *.bit (tzw. bitstream), który służy jako plik wsadowy (programujący) fizycznego układu.

W trzecim etapie projektant ma możliwość:

- załadowania programu znajdującego się w pliku wynikowym do wybranego układu oraz diagnostykę jego działania,
- weryfikację projektu, a następnie zaprogramowanie układu,
- zaprogramowanie pamięci PROM (lub FLASH), podłączenie jej do układu, a następnie załadowanie programu do układu poprzez specyficzny interfejs szeregowy.

2) Zadanie do wykonania

Należało wykonać kilka przykładowych programów wykorzystujących niektóre narzędzia dostarczone przez FPGA. Zdecydowano o wykonaniu następujących układów:

- 1) Licznik 8 bitowy wyświetlający swoją zawartość binarnie na diodach LED.
- 2) Sumator dwóch 2 bitowych liczb binarnych.
- 3) "Pociąg" - wykonanie automatu z ćwiczenia laboratoryjnego p.t. "Przejazd kolejowy".

3) Analiza teoretyczna

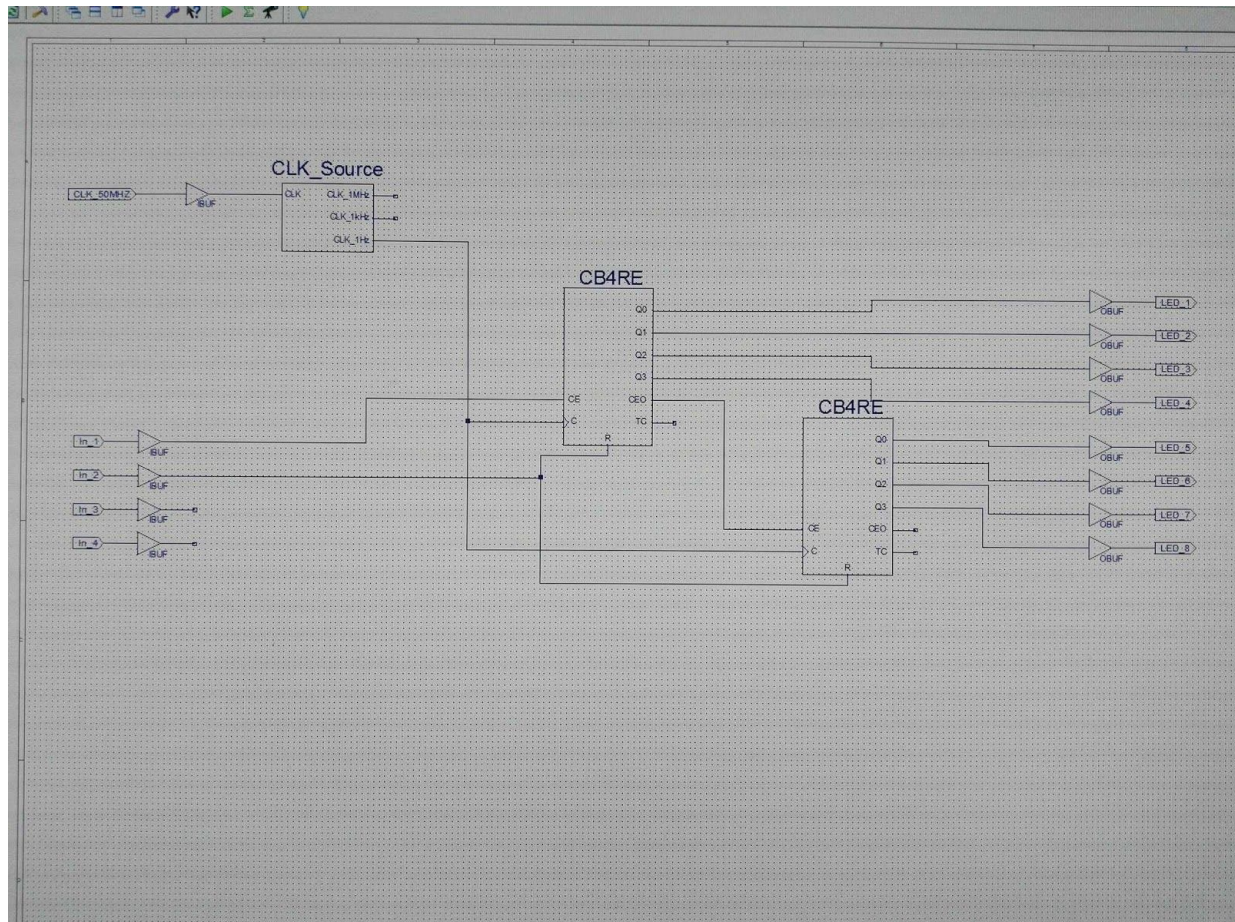
Podczas dyskusji na temat zadania pierwszego, zdecydowano, iż zostaną wykorzystane dwa liczniki 4 bitowe, które zostaną połączone kaskadowo. Za wyświetlacz aktualnego stanu licznika miały posłużyć dostępne na płycie diody LED1-LED8.

Rozpatrując możliwe rozwiązania zadania drugiego, wykorzystano schemat dodawania binarnego "w słupku". Ustalono, że w celu dodania dwóch liczb dwubitowych potrzeba trzech bramek logicznych XOR, a w celu poprawnego wyświetlenia wyniku za pomocą trzech diod LED potrzeba jeszcze kilku bramek AND.

Założeniem zadania trzeciego było wykonanie układu z jak najmniejszej ilości bramek.

4) Sposób realizacji zadania w praktyce

Sygnały wejściowe In_1 ; In_4 skojarzono z przełącznikami SW0 ; SW3, natomiast sygnały wyjściowe LED_1 ; LED_8 z odpowiednimi diodami LED płytki testowej. Do testowania układów synchronicznych można było skorzystać z post-scalera zegara wejściowego, który taktował częstotliwość 50MHz. Można było wybrać jedną z następujących częstotliwości po przeskalowaniu: CLK_1MHz, CLK_1kHz lub CLK_1Hz.



Zdj. 1 Zadania 1 - Licznik 8 bitowy

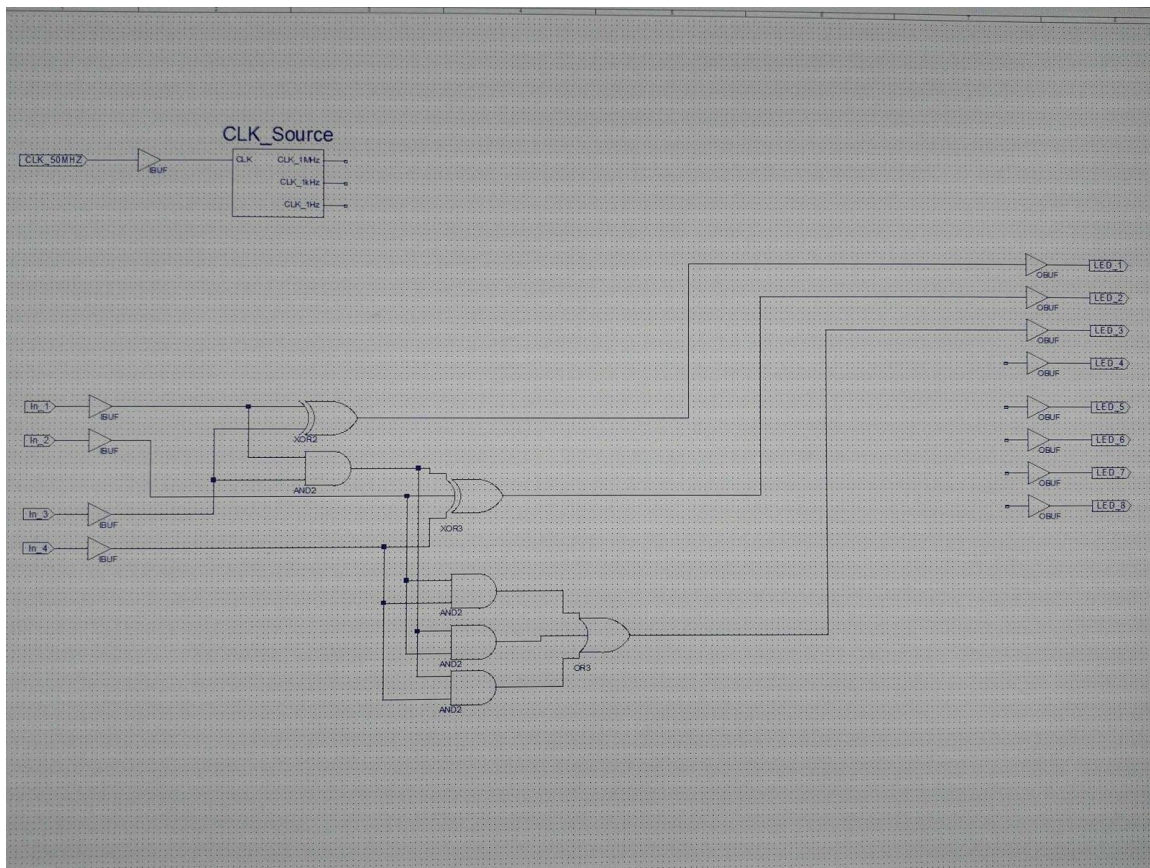
Licznik 8 bitowy został zbudowany z dwóch liczników 4 bitowych. Zostały one połączone kaskadowo - wyjście pierwszego licznika (które wysyła stan wysoki w przypadku przepełnienia licznika) zostało podłączone do wejścia zegarowego drugiego licznika.

Za pomocą przełącznika SW0 można było kontrolować, czy zbudowany licznik 8 bitowy będzie zliczać kolejne takty z modułu CLK_Sourse. Przełącznik SW1 służył do zerowania licznika za pomocą wejść RESET w obydwóch licznikach 4 bitowych.

Po zaprogramowaniu układu FPGA powyższym schematem oraz włączeniu przełącznika SW0, na płytce zaświecały się oraz gasły diody LED, symbolizujące binarne liczenie w górę.

Następnie, zmienialiśmy częstotliwości taktowania przepinając różne sygnały z CLK_Source i obserwowaliśmy jak szybko zmieniają się świecące diody.

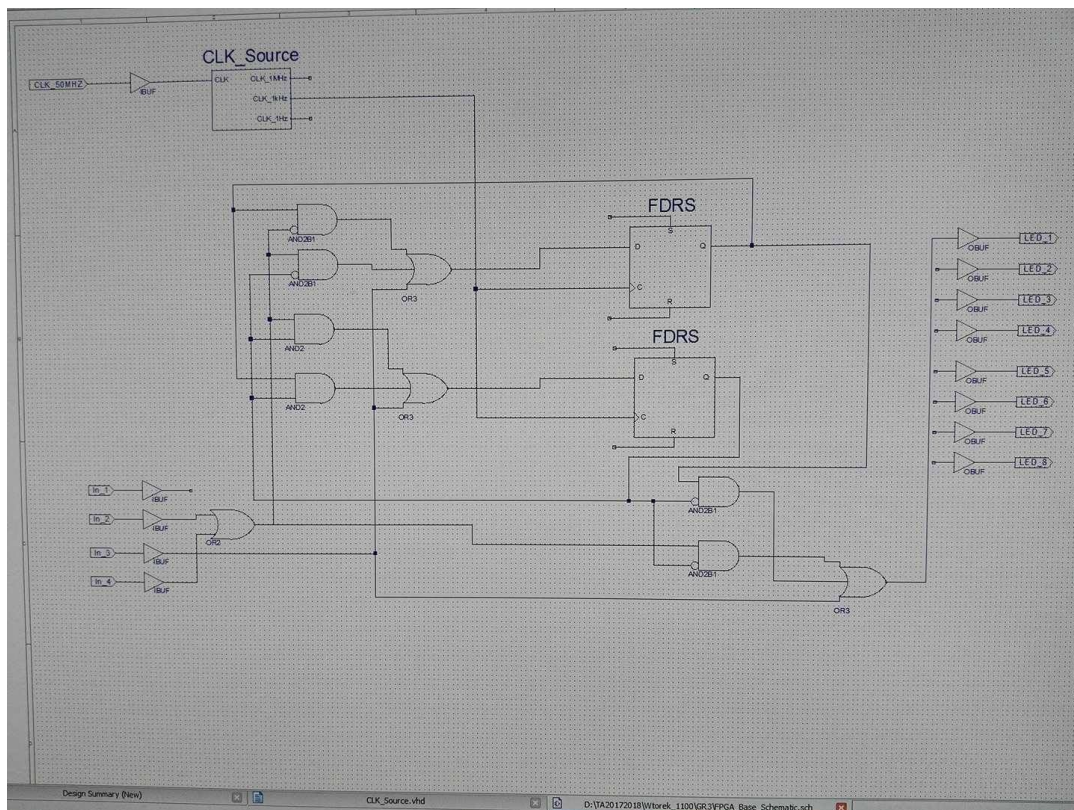
Następnie, wykonaliśmy sumator dwóch liczb 2 bitowych, który został zrealizowany za pomocą bramek logicznych. Sercem układu są bramki XOR, które realizują działanie "modulo 2".



Zdj. 2 Zadanie 2 - Sumowanie binarne dwóch liczb 2 bitowych.

Przełączniki SW0 i SW1 służyły do kodowania liczby pierwszej, a SW2 i SW3 do kodowania liczby drugiej. Wynik był wyświetlany na trzech pierwszych diodach LED. Po uruchomieniu sprawdziliśmy wszystkie możliwe kombinacje dodawania i były one wykonane poprawnie.

Ostatnim "programem", który został zapisany w pamięci FPGA był schemat realizujący automat z ćwiczenia o przejeździe kolejowym. Mając do dyspozycji szereg różnych układów i bramek logicznych, mogliśmy zrealizować układ w inny sposób, niż tylko za pomocą bramek NAND. Wykorzystano tutaj 2 przerzutniki synchroniczne typu D, które służyły za elementy pamiętające aktualny stan automatu.



Przełączniki od SW1 do SW3 służyły za kolejne wejścia układu A,B i C. Wyjście układu było obserwowane na pierwszej diodzie. W sumie do budowy układu wykorzystaliśmy 10 bramek logicznych i 2 przerzutniki. Automat działał poprawnie.

5) Wnioski i obserwacje

- Wszystkie zadania zostały wykonane poprawnie.
- Tworzenie prostych rozwiązań z pomocą FPGA niekoniecznie jest dobrą alternatywą do napisania prostego programu dla mikrokontrolera. Wynika to z faktu, że “programy” FPGA są realizowane współbieżnie. W większości przypadków przetwarzanie sekwencyjne wraz obsługą przerwań będzie wystarczająca.
- Obserwując działanie dwóch liczników 4 bitowych połączonych ze sobą kaskadowo, zauważono, że pierwszy licznik spełnia podobne działanie jak moduł CLK_Source. Można wysunąć hipotezę, że moduł CLK_Source jest swoistym licznikiem, który przepełniając się spełnia swoje działanie jako dzielnik częstotliwości.
- Wykorzystanie układu FPGA do zbudowania automatu z ćwiczenia laboratoryjnego “Przejazd kolejowy” pozwoliło na wykorzystanie mniejszej ilości bramek, skrócenie czasu montażu. Ponadto, pozwoliło to również na szybsze wykrycie i wyeliminowanie wszystkich błędów wynikających z wykonania błędnych połączeń.
- Układ sterujący przejazdem kolejowym został uproszczony dzięki wykorzystaniu bramek XOR.
- Realizując układ dodawania binarnego początkowo występowały błędy wynikające z błędnego dodania bitu przeniesienia z pierwszej pozycji do drugiej pozycji. Dzięki możliwościom FPGA poprawienie błędów mogło zostać zrealizowane bez wnikliwej analizy teoretycznej.

Układy FPGA sprawdzają się wszędzie tam, gdzie potrzebne jest niezwykle szybkie wykonanie jakiegoś algorytmu. Używane są między innymi w cyfrowym przetwarzaniu sygnałów, lotnictwie i wojsku czy też podczas prototypowania układów wyspecjalizowanych (ASIC).

Ciekawostką jest, że układy FPGA firmy Xilinx zostały użyte podczas misji na Marsa zakończonej lądowaniem łazików Spirit i Opportunity.