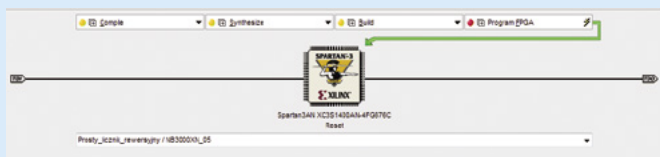


Programowanie układów FPGA w Altium Designer

Synteza i implementacja projektu

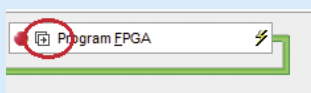
W poprzednich częściach cyklu utworzyliśmy od podstaw projekt dla układu FPGA w środowisku Altium Designer. Następnie został on zweryfikowany, a w kolejnym etapie przeprowadziliśmy konfigurację. W tym artykule opiszemy, jak wykonać kompilację, syntezę a następnie implementację projektu.



Rysunek 1. Proces przetwarzania danych „Flow Control” w środowisku Altium Designer



Rysunek 2. Brak oprogramowania narzędziowego Xilinx



Rysunek 3. Bez względu na wykonanie wszystkich etapów

Gdy mamy już kompletny, sprawdzony projekt, to znaczy – bez żadnych błędów elektrycznych czy składniowych, możemy rozpocząć wykonanie syntezy projektu, czyli przetworzenie plików

źródłowych na format pośredni, na tak zwane pliki netlisty, z których można będzie wykonać implementację – wygenerować pliki wynikowe dla dowolnie wybranej technologii. Ostatnim etapem będzie zaprogramowanie układu FPGA z wykorzystaniem wygenerowanych wcześniej plików. Cały ten proces przetwarzania danych – począwszy od kompilacji plików źródłowych po zaprogramowanie urządzenia fizycznego – jest przeprowadzany w oknie *Devices*.

Po pierwsze, należy upewnić się, że okno *Devices* jest aktywnym oknem Altium Designer (*View* -> *Devices View*). Następnie trzeba sprawdzić, czy wybrano opcję *Live* i czy wskaźnik połączenia z płytą zewnętrzną wskazuje *Connected* (jest wyświetlany na zielono). Jeśli płyta NanoBoard będzie prawidłowo dołączona do komputera, to w oknie *Devices* będzie wyświetlany między innymi kanał transmisyjny JTAG składający się z łańcucha układów konfigurowalnych – w tym przykładzie jest to jeden układ Xilinx SpartanAN. Do każdego takiego układu są przyporządkowane kolejne etapy procesu przetwarzania danych (*flow control*), co pokazano na **rysunku 1**.

Możliwość wykonania syntezy i implementacji będzie możliwa gdy zostaną spełnione dwa dodatkowe warunki:

- Będzie zainstalowane oprogramowanie producenta FPGA w naszym przypadku oprogramowanie firmy Xilinx np.: WEB Pack (pisałmy o tym w pierwszym

artykule w EP 12/2012). Brak odpowiedniego oprogramowania narzędziowego jest sygnalizowany komunikatem jak na **rysunku 2**. Może zdarzyć się, że nie zobaczymy właściwej ikony układu FPGA (jak to widać u nas na rysunku – Spartan 3AN). Wtedy konieczne będzie zainstalowanie brakujących bibliotek za pomocą menu *DXP* -> *Plug-ins and updates*.

- Z listy wyboru umieszczonej pod symbolem układu FPGA mamy możliwość wyboru naszego projektu i związanej z nim konfiguracji. Takie powiązanie istnieje, gdy projekt ma przypisaną konfigurację zawierającą plik ograniczeń, w którym poprawnie zdefiniowano typ zastosowanego układu FPGA. Była o tym mowa w poprzednim artykule.

Mając możliwość skorzystania z automatycznej konfiguracji, nazwy kolejnych konfiguracji powstają zgodnie z szablonem *Project* -> *Configuration*. W naszym projekcie będzie to *Prosty_licznik_rewersyjny/NB3000XN_05* (rys. 1).

Proces *Flow Control* (kontrola przepływu) składa się z czterech odrębnych etapów. Pliki wyjściowe z poprzedniego etapu są automatycznie pobierane do kolejnego etapu. Chociaż cały proces można uruchomić klikając bezpośrednio na przycisk *Program FPGA* (przycisk ostatniego etapu), to (szczególnie przy nowym projekcie) skończyć z każdego etapu po kolei.

Można uruchomić wszystkie etapy bez względu na ich aktualny status, klikając na ikonę strzałki znajdującej się po lewej stronie przycisku (**rysunek 3**).

- Aby uruchomić proces kompilacji projektu, należy kliknąć na przycisk *Compile*. Ten pierwszy etap ma za zadanie skompilowanie dokumentów źródłowych w ramach projektu FPGA. Jeśli projekt obejmowałby procesory softwarowe, to odpowiadające im oprogramowanie byłoby kompilowane na tym etapie. W naszym przykładowym, nieskomplikowanym projekcie nie używamy procesora, więc kompilacja będzie się sprowadzała tylko do weryfikacji plików źródłowych pod kątem poprawności elektrycznej i składniowej. Zauważmy jednak, że zgodnie z poprzednim artykułem ten projekt był już sprawdzany, więc nie powinien mieć błędów! Komunikaty kompilatora można śledzić na panelu zatytułowanym *Messages*. Każdy etap, który zakończy się pomyślnie, zmieni kolor świecenia powiązanego z nim wskaźnika na zielony.

Po poprawnie wykonanej kompilacji, należy kliknąć na przycisk *Synthesize*. Podczas syntezy, wszystkie dokumenty źródłowe (arkusze schematu) są tłumaczone do pośrednich plików VHDL, które następnie są syntetyzowane do uniwersalnej netlisty EDIF. Wszystkie pliki są kopiowane do odpowiedniego katalogu wyjściowego (**rysunek 4**). Należy przypomnieć, że domyślnym programem do przeprowadzenia syntezy jest syntezer dostarczany przez Altium. W wyniku jego działania otrzymujemy uniwersalne pliki netlisty, które posłużą do wykonania implementacji w dowolnej technologii wspieranej przez Altium.

Jeżeli cały proces syntezy zakończy się powodzeniem, zostanie utworzony katalog *Generated [ConfigurationName]*, w którym zostaną umieszczone pliki:

- EDIF (*Prosty_licznik_rewersyjny.edf*),
- VHDL (*Prosty_licznik_rewersyjny.vhd*),
- w katalogu Report plik z ostrzeżeniami z ostatniej syntezy (*Prosty_licznik_rewersyjny.log*).

Podgląd kolejnych etapów syntezy jest możliwy w oknie *Messages*.

Teraz należy kliknąć na przycisk *Build*. Na tym etapie Altium Designer wywołuje narzędzia *Place&Route* (**rysunek 5**) konkretnego producenta FPGA (w naszym wypadku Xilinx). Każdy z następujących pięciu etapów cząstkowych będzie działał kolejno:

- 1) **Translate Design**: przetwarza pliki netlisty top level EDIF z powiązanymi modułami, do podstawowego formatu *Native Generic Database* (NGD).
- 2) **Map Design To FPGA**: mapowanie projektu w FPGA.
- 3) **Place and Route**: wykonuje niskopoziomowe rozmieszczenie logiki i jej logiczne połączenie, operacja wykonywana jest iteracyjnie i czas trwania tego etapu jest znaczący.
- 4) **Timing Analysis**: wykonuje analizę czasową projektu, zgodnie z wszelkimi ograniczeniami czasowymi, które zostały zdefiniowane. Jeśli nie są określone – jak to jest w naszej prostej konstrukcji – zostaną użyte wartości domyślne.
- 5) **Make Bit File**: generuje plik o rozszerzeniu *.bit* służący do programowania pamięci konfiguracyjnej w układzie FPGA.

Ogólne informacje i komunikaty z przebiegu implementacji są dostępne w oknie *Messages*. Bardziej szczegółowe informacje dotyczące poszczególnych etapów, kolejnych iteracji są dostępne w panelu *Output* (dostępnym pod przyciskiem *System* w prawym dolnym rogu ekranu).

Gdy kolejne etapy zostaną zakończone z powodzeniem, zostanie wyświetlone okno (*Results Summary*, **rysunek 6**) zawierające podsumowanie całego procesu implementacji. Okno zawiera informacje odnośnie do wykorzystania zasobów układu FPGA w urządzeniu docelowym, jak również wszelkie informacje czasowe odnoszące się do tego projektu. Po zapoznaniu się z tymi informacjami, okno należy zamknąć.

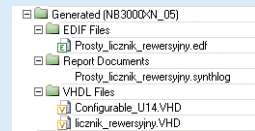
Ostatnim etapem będzie zaprogramowanie układu FPGA. W tym celu trzeba kliknąć na przycisk *Program FPGA*. Uruchamia to proces zapisu układu FPGA plikami powstałymi w trakcie implementacji. Zwykle są to pliki o rozszerzeniu *.bit*. Proces przebiega z użyciem kanału transmisyjnego JTAG. Pasek postępu jest wyświetlany na pasku stanu środowiska Altium Designer (**rysunek 7**).

Gdy projekt zostanie załadowany, opis pod ikoną układu FPGA zostanie zmieniony z *Reset* na *Program-*

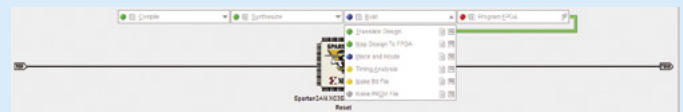
med, natomiast na płycie zaświeci się na zielono dioda sygnalizacyjna „Program”. Upewnia nas to, że projekt został załadowany do FPGA.

W celu przetestowania właśnie wykonanego programu, należy użyć przełączników DIP-switch na płycie NanoBoard w następujący sposób:

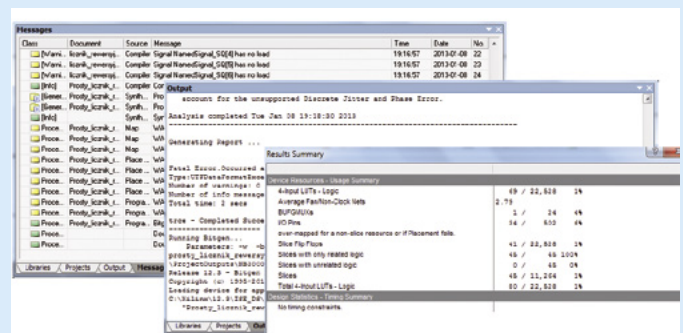
- *Przełącznik nr 8*: przełączenie w pozycję ON powoduje rozpoczęcie zliczania przez licznik, jeżeli był on poprzednio zatrzymany i/lub pulsowanie diod LED od lewej do prawej.
- *Przełącznik nr 7*: przełączenie w pozycję ON powo-



Rysunek 4. Pliki wygenerowane na etapie syntezy



Rysunek 5. Przebieg implementacji – w trakcie Place&Route



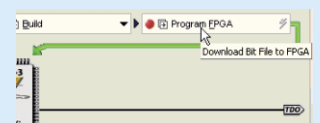
Rysunek 6. Kolejne raporty implementacji

duje rozpoczęcie zliczania przez licznik, jeżeli był on poprzednio zatrzymany i/lub pulsowanie diod LED od prawej do lewej.

- *Przełącznik nr 6*: przełączenie do pozycji ON, aby zatrzymać licznik. Licznik zostanie zatrzymany pod warunkiem, że przełączniki 7 i 8 są wyłączone (OFF). Naciśnięcie przycisku „DAUGHTER BD TEST/RESET” zamontowanego na płycie NanoBoard powoduje wyłączenie diod LED.

Na koniec i dla dalszych potrzeb, należy zapamiętać bieżący projekt.

Można zauważyć, że diody LED znajdujące się na płycie NanoBoard świecą jednakowo i nie można dostrzec efektu przemiatania światła charakterystycznego dla licznika pierścieniowego. Dzieje się tak dlatego, że układ jest taktowany zegarem o zbyt wysokiej częstotliwości. Częstotliwość referencyjna *Clk_Ref* na płycie NanoBoard wynosi aż 20 MHz. Aby dostrzec zapalające się kolejno diody LED musimy podzielić częstotliwość sygnału zegarowego przez 1000000. Uzyskamy wtedy częstotliwość taktującą 20 Hz. Zmiany z taką częstotliwością są już zauważane przez ludzkie oko.



Rysunek 7. Rozpoczęcie programowania FPGA

Podsumowanie

W kolejnym artykule rozbudujemy projekt dodając dodatkowy arkusz, na którym będzie umieszczony dzielnik zegarowy a jednocześnie przy tej okazji pokażemy jak stworzyć projekt hierarchiczny.

Wojciech Pietrasina
Evatronix SA