

LiveDesign w praktyce, część 4



Przykładowy projekt licznika Johnsona w obecnej wersji został przygotowany i skonfigurowany pod kątem implementacji na płycie uruchomieniowej NanoBoard NB-1 firmy Altium. W kolejnych częściach cyklu pokażemy, w jaki sposób zmienić konfigurację projektu, aby dostosować go do innej platformy sprzętowej, np. płyty z zestawu LiveDesign Evaluation lub naszej własnej płyty ewaluacyjnej.

Płyta NanoBoard nie jest sprzętem powszechnie dostępnym dla amatorów, jednak zastosowane podejście jest celowe i ma pokazać elastyczność systemu DXP 2004, który pozwala na bardzo swobodny wybór platformy sprzętowej. Zasady tworzenia i uruchamiania projektu są niezmiennie, a dostosowanie go do konkretnej konfiguracji sprzętowej sprowadza się do wyboru właściwego chipu FPGA i powiązaniu sygnałów we/wy z otoczeniem sprzętowym układu programowalnego. Szczegóły już za miesiąc, a teraz prześledzimy proces programowania kości FPGA i uruchamiania projektu na płycie NanoBoard NB-1 z układem Xilinx Spartan IIE. Ponieważ nasz projekt nie wykorzystuje żadnego „soft-procesora”, wystarczy nam oprogramowanie Protel 2004, nawet w wersji ewaluacyjnej. Nexar 2004 jest niezbędny tylko przy projektowaniu systemów mikroprocesorowych na FPGA.

Programujemy układ FPGA...

...korzystając z okna *Devices*, które uruchamiamy, wybierając z menu *View>Devices*. Widok urządzeń, pokazany na ilustracji poniżej, pozwala na przejście całego procesu (od lewej do prawej), wymaganego do załadowania programu do układu FPGA. Kolejne przyciski odpowiadają za poszczególne etapy:

- *Compile* – uruchamia kompilację projektu i związaną z tym kon-

Po wstępnych częściach cyklu, w których powstawał przykładowy projekt FPGA, przyszedł czas na jego uruchomienie. W bieżącym odcinku pokażemy, jak uruchomić i przetestować projekt, korzystając z oprogramowania Altium DXP 2004 i płyty uruchomieniowej NanoBoard, która – przypominamy – była jedną z nagród w naszych konkursach.

trołę błędów,

- *Synthesize* – uruchamia proces syntezy, który w efekcie tworzy listę połączeń w formacie EDIF,
- *Build* – uruchamia szereg zadań, realizowanych przez narzędzia dedykowane dla danego układu programowalnego, które przekształcają plik EDIF na Rys. 17. tzw. *bit file*, czyli

kodeks programujący układ FPGA,

- *Program FPGA* – załadowanie kodu programującego do chipu na płycie, w naszym przykładzie do układu Xilinx Spartan IIE. Możemy uruchomić cały proces przetwarzania projektu, jednym kliknięciem myszki w przycisk oznaczony *Program FPGA*. To wygodny sposób, ale w celach poznawczych, przejdźmy kolejne etapy, krok po kroku.

Należy zacząć od sprawdzenia sprzętu. Jeśli płyta NanoBoard jest włączona i mamy zaznaczoną opcję *Live*, kontrolka *Connected* powinna być koloru zielonego. Oznacza to, że system komunikuje się z płytą uruchomieniową. Klikamy przycisk *Compile*. Jeśli kompilacja przebiegnie prawidłowo, czerwona kontrolka zmieni kolor na zielony. Jeśli w trakcie kompilacji pojawi się okno *Messages* z komunikatami o błędach, należy wrócić do schematów, poprawić błędy, zapisać pliki i uruchomić kompilację ponownie.

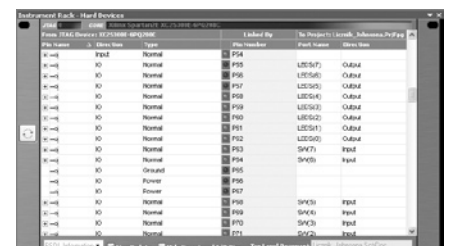
Klikamy przycisk *Synthesize*. Jeśli proces syntezy przebiegnie prawidłowo, pojawi się folder o nazwie *Generated* (nazwa_konfiguracji), w którym przechowywane są wygenerowane pliki EDIF, VHDL oraz plik *.log* (dziennik syntezy). Czerwona kontrolka na przycisku *Synthesize* zmieni



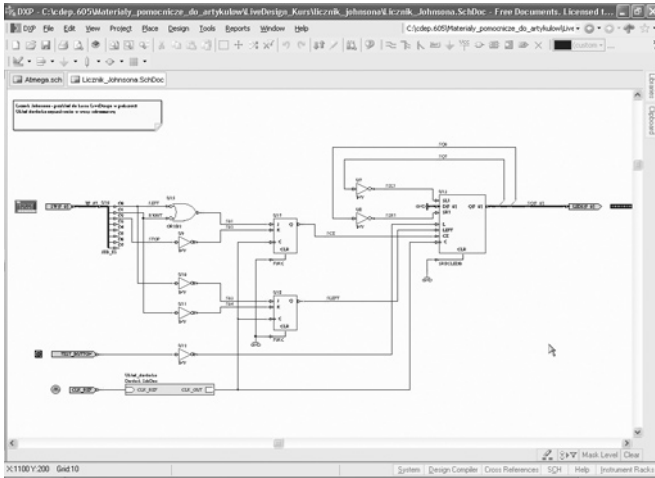
kolor na zielony, a pod ikoną układu Spartan IIE pojawi się nazwa projektu z nazwą konfiguracji NB_SpartanIIE, użytą w przykładzie.

Podczas syntezy, dokumenty źródłowe (schematy) są konwertowane na pośrednie pliki VHDL, które są następnie syntezowane do formatu EDIF, odpowiedniego dla narzędzi do implementacji (*Place & Route tools*) producentów układów FPGA. Błędy wykryte podczas syntezy wynikają z błędów w plikach pośrednich, więc należy wrócić do plików źródłowych i wyeliminować problemy. Klikając dwukrotnie informację o błędzie w oknie *Messages*, możemy zobaczyć miejsce wystąpienia błędu w plikach źródłowych i pośrednich plikach VHDL.

Klikamy przycisk *Build*, który uruchamia szereg procesów, generujących ostatecznie tzw. *bit file*, który można załadować do kości FPGA. Na



Rys. 18.



Rys. 19.

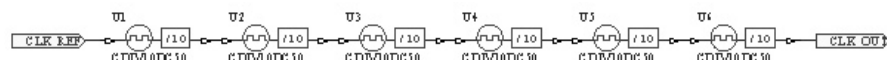
tym etapie DXP 2004 wykorzystuje oprogramowanie producenta układu programowalnego, np. ISE WebPACK Xilinx'a. Ten etap trwa dość długo, a przejścia kolejnych procesów sygnalizowane są zmianą koloru kontrolki z czerwonego na zielony. Zakończenie całego etapu jest sygnalizowane zmianą koloru kontrolki na przycisku *Build* na zielony. Informacje i komunikaty generowane podczas tego etapu możemy przejrzeć w oknie panelu *Output*. Klikamy przycisk Program FPGA, aby w kilka sekund załadować kod programujący do układu FPGA na płytce.

Od tego momentu, możemy na żywo obserwować działanie naszego projektu w sprzęcie. Zestaw przełączników DIP na płytce NanoBoard, posłuży nam do sterowania pracą licznika. Pstrykając odpowiednie przełączniki, powodujemy:

- 1 – uruchamia przesuwanie światła na linii LED w lewo,
- 2 – uruchamia przesuwanie światła na linii LED w prawo,
- 3 – zatrzymuje licznik.

Naciśnięcie przycisku *Test/Reset* poniżej liniiki LED na płytce, powoduje wyzerowanie licznika.

Jak się okazuje, trudno zaobserwować prawidłowe działanie naszego licznika Johnsona w tym stanie projektu. Wszystkie kontrolki LED świecą równocześnie. Tylko reakcja programu na przyciski pozwala sądzić, że projekt działa. Równoczesne świecenie całej liniiki LED spowodowane jest zegarem 50 MHz, którym domyślnie jest taktowany układ. Musimy spowolnić zegar milion razy, aby



Rys. 20.

zobaczyć sekwencyjne świecenie kontrolki LED. Dalej dodamy dzielnik częstotliwości do schematu, ale najpierw spróbujemy sprawdzić działanie projektu za pomocą instrumentu *Hard Devices*.

Instrumenty wirtualne...

...to odpowiedniki rzeczywistych przyrządów kontrolno-pomiarowych, jakimi posługujemy

się do badania tradycyjnych układów cyfrowych. Różnica polega na tym, że przyrządy wirtualne pracują wewnątrz układu FPGA i poprzez panele dostępne na ekranie komputera, pozwalają badać sygnały wewnątrz układu.

System DXP2004 oferuje bibliotekę liczącą kilkanaście różnych instrumentów wirtualnych, m.in. analizatory logiczne, generator i licznik częstotliwości, które możemy umieścić wprost na schemacie i korzystać z nich w czasie uruchamiania projektu. W dalszej części kursu pokażemy przykłady wykorzystania takich instrumentów, umieszczonych w projekcie.

Natomiast teraz wystarczy nam panel *Hard Devices*, czyli specjalny instrument, który pozwala odczytać sygnały na wyprowadzeniach układu FPGA. Ponieważ kontrolki LED na płytce NanoBoard podłączone są wprost do układu FPGA, obserwując poziomy sygnałów na odpowiednich wyprowadzeniach, zobaczymy jak świecą diody. Sygnały w panelu *Hard Devices* wyświetlane są znacznie wolniej, niż częstotliwość taktowania płyty NanoBoard, więc będzie można zobaczyć sekwencyjne przełączanie LED-ów bez spowalniania zegara.

Instrument *Hard Devices* uruchamiamy, klikając dwukrotnie w ikonę Spartan-IIIE w oknie *Devices*. Pojawi się panel oznaczony *Instrument Rack - Hard Devices*, pokazany na **rys. 17**.

Należy zwrócić uwagę, żeby kratka *Live Update* była zaznaczona, ponieważ tylko wtedy możemy obserwować zmiany sygnałów na bieżąco. Jeśli suwakiem przesuniemy widok na

List. 1. Opis VHDL realizujący funkcję dzielnika częstotliwości

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity clock_divider is
port (
    CLK_REF : in std_logic;
    CLK_OUT : out std_logic
);
end entity;

architecture RTL of clock_divider is
begin
    process(CLK_REF)
        variable i : integer range 0
        to 999999;
    begin
        if rising_edge(CLK_REF) then
            if i = 0 then
                CLK_OUT <= '1';
                i := 999999;
            else
                CLK_OUT <= '0';
                i := i - 1;
            end if;
        end if;
    end process;
end architecture;
    
```

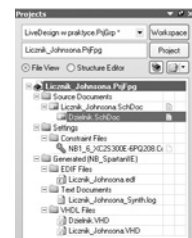
wyprowadzenia P55...P62, zobaczymy błyskanie ikon związanych z portami LEDS(0)...LED(7). Podobnie możemy zaobserwować zmiany sygnałów w reakcji na manipulacje przełącznikami DIP-switch na płytce. Lokalizację właściwych pinów ułatwia opis w kolumnie *Port Name*, który odpowiada nazwom portów umieszczonych na naszym schemacie. Kolumna *Direction* wskazuje kierunek wyprowadzenia – wyjście lub wejście.

Rozbudowujemy układ...

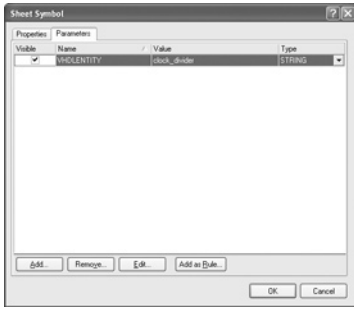
...dodając część spowalniającą jego działanie – dzielnik częstotliwości, który zaprojektujemy w formie dodatkowego arkusza schematu.

Ponieważ płyta NanoBoard jest taktowana domyślnie zegarem 50 MHz, będziemy potrzebować sześciu dzielników przez 10 (element CDIV10DC50 w bibliotece) w naszym liczniku Johnsona, aby spowolnić przełączanie kontrolki LED tak, żeby dało się go zaobserwować naocznie. Dzielnik wykonamy w formie pod-arkusza, aby pokazać w jaki sposób tworzyć projekty hierarchiczne dla FPGA.

Otwieramy schemat *Licznik_Johnsona.SchDoc* (**rys. 18**) i wstawiamy do niego symbol arkusza, który będzie reprezentował arkusz podrzędny z układem dzielnika częstotliwości, jak



Rys. 21.



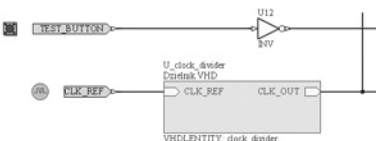
Rys. 22.

na ilustracji poniżej. Symbol arkusza wstawiamy, wybierając z menu polecenie *Place>Sheet Symbol*. Następnie naciskamy od razu klawisz TAB, aby w oknie właściwości elementu wpisać oznaczenie np. Układ dzielnika oraz nazwę pliku, pod jaką będzie zapisany arkusz, np. Dzielnik.SchDoc. Klikamy OK, aby zamknąć okno dialogowe, następnie klikamy na arkuszu miejsce położenia elementu i kolejny raz, ustawiając jego rozmiar. Symbol arkusza wstawiamy w miejsce połączenia biegnącego od portu CLK_REF, które uprzednio trzeba przerwać, korzystając z polecenia w menu *Edit>Break Wire*.

Symbol arkusza należy wyposażyć w dwa wejścia, które oznaczamy CLK_REF i CLK_OUT. Wejścia do arkusza dodajemy, wybierając z menu *Place>Add Sheet Entry*. Należy określić ich typ (*I/O Type*), odpowiednio na wejście (*Input*) i wyjście (*Output*). Można to zrobić na etapie wstawiania naciskając klawisz TAB, który otwiera okno właściwości obiektu.

Następnie tworzymy arkusz podrzędny, wybierając z menu *Design>Create Sheet from Symbol*. Ustawiamy kursor nad nowym symbolem arkusza i klikamy w niego. Pojawia się okno dialogowe *Confirm* zapytaniem, czy odwrócić kierunki wejść/wyjść. Należy kliknąć *No*. Zostanie utworzony nowy plik schematu i pojawi się na ekranie z widocznymi portami CLK_REF i CLK_OUT, które zostały dodane automatycznie. Teraz musimy umieścić sześć elementów dzielnika CDIV10DC50, wybierając je z biblioteki FPGA Generic.IntLib, jak widać na rys. 19.

Następnie kompilujemy projekt, aby sprawdzić, czy nie występu-



Rys. 23.

ją błędy. W razie potrzeby usuwamy błędy i kompilujemy projekt ponownie. Po kompilacji możemy zobaczyć hierarchiczną strukturę projektu, zaglądając do panelu *Projects* (rys. 20). Jak widać, arkusz dzielnika (Dzielnik.SchDoc), występuje jako podrzędny w stosunku do schematu *Licznik Johnsona.SchDoc*.

Przechodzimy do widoku *Devices* i uruchamiamy proces programowania FPGA, klikając ostatni przycisk *Program FPGA*. Po uruchomieniu licznika, będzie można zobaczyć przemieszczanie się światła na linijce LED. Przełączniki 1 i 2 z zestawu DIP switch umożliwiają zmianę kierunku zliczania, odpowiednio z prawej do lewej i z lewej do prawej. Przełącznik numer 3 zatrzymuje licznik.

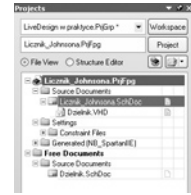
Można również w języku VHDL...

... opisać część projektu, która zastąpi dzielnik częstotliwości, zaprojektowany wcześniej w formie schematu. Wspomnieliśmy na początku kursu o tym, że system DXP 2004 pozwala korzystać z różnych plików źródłowych – tak ze schematów, jak również kodu w językach opisu sprzętu VHDL oraz Verilog. W jednym projekcie można łączyć różne rodzaje opisu z zastrzeżeniem, że zawsze schemat pełni rolę głównego pliku projektu, umieszczonego najwyżej w hierarchii.

Na list. 1 znajduje się fragment kodu VHDL, który w projekcie będzie obniżał częstotliwość zegara milion razy. Plik VHDL zostanie powiązany z głównym schematem licznika Johnsona za pomocą symbolu arkusza, który zastąpi wcześniejszą wersję dzielnika, zbudowaną w formie schematu.

Klikamy prawym klawiszem myszki nazwę projektu FPGA w panelu *Project* i z menu podręcznego wybieramy polecenie *Add New to Project* i dalej *VHDL Document*. Otworzy się edytor tekstowy, za pomocą którego możemy przepisać kod VHDL pokazany wyżej. Następnie, korzystając z polecenia *Save*, zapisujemy nowy plik np. pod nazwą *Dzielnik.VHD* w katalogu, gdzie znajdują się pozostałe pliki projektu.

Teraz utworzymy symbol arkusza z naszego pliku VHDL, aby zastąpić nim poprzednią wersję *Dzielnik.SchDoc*. Mając otwarty schemat *Licznik Johnsona.SchDoc*, tworzymy nowy symbol arkusza, wybierając z menu *Design>Create Sheet Symbol from Sheet*. Pojawia się okno dialogowe



Rys. 24.

Choose Document to Place, w którym wskazujemy plik *Dzielnik.VHD*, utworzony przed chwilą i zamykamy okno, klikając OK. Pojawi się symbol arkusza, który umieszczamy na schemacie zaraz poniżej symbolu arkusza *Dzielnik.SchDoc*. Możemy jeszcze upewnić się, klikając dwukrotnie nowy symbol arkusza, czy w oknie dialogowym *Sheet Symbol*, na zakładce *Parameters*, został dodany parametr *VHDLEntity*, jak widać na ilustracji poniżej. Opcja *Visible*, obok parametru, powinna być zaznaczona (rys. 21).

Usuujemy ze schematu symbol arkusza *Dzielnik.SchDoc*, a w jego miejsce przesuwamy nowo utworzony *Dzielnik.VHD* (rys. 22). Należy zwrócić uwagę, czy połączenia są doprowadzone prawidłowo, a następnie zapisujemy schemat, wybierając z menu polecenie *Save*.

Usuujemy plik *Dzielnik.SchDoc* z projektu, klikając prawym klawiszem myszki nazwę dokumentu w panelu *Projects* i wybierając z menu polecenie *Remove from Project*. Nie będzie już potrzebny, ponieważ zastąpiliśmy go plikiem VHDL. Zapisujemy projekt FPGA, wybierając z menu polecenie *Save*.

W panelu *Projects* widać (rys. 23), że w strukturze hierarchicznej projektu, plik *Dzielnik.SchDoc* został zastąpiony przez plik *Dzielnik.VHD*.

Pozostało jeszcze sprawdzić, czy nasz projekt działa. Przechodzimy do widoku *Devices* i klikamy ostatni przycisk *Program FPGA*, aby uruchomić kolejne etapy przetwarzania projektu, od kompilacji, przez syntezę, budowę pliku *bit file*, skończywszy na zaprogramowaniu chipu FPGA. Jeśli wszystko przebiegło prawidłowo, można uruchomić licznik Johnsona, używając przełączników DIP na płycie, jak poprzednio.

W kolejnej części cyklu pokażemy, w jaki sposób przenieść projekt na inną platformę sprzętową. Zarówno, korzystając z innego modułu FPGA na płycie *NanoBoard*, jak i modyfikując projekt do pracy na płycie uruchomieniowej z zestawu *LiveDesign Evaluation* firmy *Alti*um.

Grzegorz Witek, Evatronix