

Procesory Nios II w układach FPGA (2)

Projektowanie systemu cyfrowego w układzie programowalnym



W poprzedniej części kursu przedstawiono możliwości implementacji systemów cyfrowych w układach programowalnych. Głównym elementem takiego systemu jest przeważnie procesor typu softcore. Przykładem jest opisywany procesor Nios II, którego rdzeń oraz peryferia można dowolnie kształtować w układzie FPGA. W artykule przedstawiono proces projektowania i implementacji systemu cyfrowego z procesorem Nios II przy użyciu środowiska projektowego firmy Altera. Projekt został tak przygotowany, aby z jego pomocą można było uruchomić prostą aplikację programową.

Projektowanie i implementacja w układzie FPGA zintegrowanego systemu cyfrowego (*System-On-Chip* – SoC) z procesorem Nios II wymaga kilkustopniowego procesu projektowego. Można go podzielić na 3 etapy: projekt systemu cyfrowego, utworzenie i kompilacja projektu w układzie FPGA oraz opracowanie programu sterującego. Firma Altera udostępnia oprogramowanie, dla każdego z tych kroków:

Quartus II – środowisko programistyczne służące do tworzenia i implementacji projektów dla układów programowalnych firmy Altera. W programie Quartus II łączy się moduły projektowanego systemu cyfrowego, definiowane są również połączenie z blokami I/O układu FPGA i jego wyprowadzeniami. Quartus II umożliwia również kompilację i syntezę projektu oraz generuje pliku konfiguracyjny układu FPGA.

SOPC Builder – program narzędziowy uruchamiany bezpośrednio ze środowiska projektowego Quartus II. Jest to rozbudowany kreator systemu cyfrowego budowanego w oparciu o magistralę Avalon. W programie SOPC Builder łączone są między sobą bloki IP takie jak: procesory, peryferia komunikacyjne, bloki pamięci, itp.

Overclocking Nios II

Ze względu na planowaną rozbudowę systemu cyfrowego, w przykładowych projektach stosowany jest zewnętrzny sygnał zegarowy 50 MHz. W układach Cyclone II dostępne są pętle PLL, które umożliwiają powielenie częstotliwości sygnału wewnątrz układu FPGA. Autorowi udało się utworzyć projekt z procesorem Nios II/e taktowanym sygnałem zegarowym o częstotliwości 125 MHz. Maksymalna częstotliwość sygnału zegarowego zależeć będzie w znacznym stopniu od zajętości zasobów.

Nios II EDS – środowisko programistyczne dla procesorów Nios II opracowane na bazie bezpłatnego IDE Eclipse.

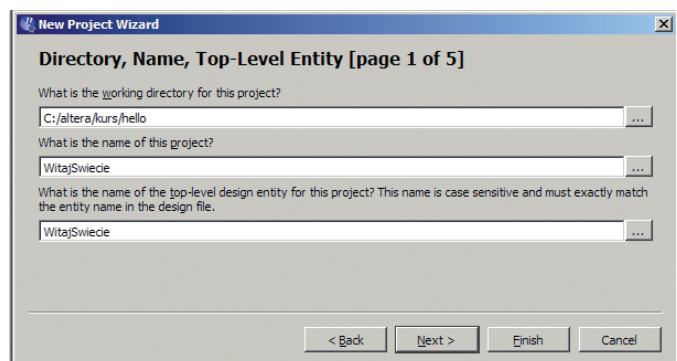
Budowanie własnego systemu SoC

Projektowanie systemu cyfrowego w układach programowalnych firmy Altera należy rozpocząć od utworzenia nowego projektu w programie Quartus II. Opisywane w artykule przykłady były tworzone przy użyciu najnowszej wersji tego środowiska oznaczonej numerem 10.

Po uruchomieniu programu Quartus II należy wybrać *Create a New Project* z okna powitalnego lub z menu *File* opcję *New Project Wizard*. W kolejnych krokach (**rysunek 3**) kreatora należy wskazać miejsce na dysku, gdzie mają być zapisywane pliki projektu oraz podać nazwę projektu (w przykładzie *WitajSwiecie*) i nazwę głównej encji projektowej (głównego modułu projektu, który będzie zawierał wszystkie inne moduły projektu). Należy pominąć krok dodawania własnych bibliotek, gdyż nie będą używane w tym przykładzie. W kroku 3. należy podać, dla jakiego układu programowalnego przeznaczony jest projekt. Po wybraniu odpowiedniego układu z listy należy zakończyć pracę kreatora klikając w przycisk *Finish*.

Nowo utworzony projekt nie zawiera żadnych plików. Głównym plikiem przykładowego projektu będzie plik edytora schematów (rozszerzenie *.bdf*). Aby dodać nowy plik edytora schematów do projektu, należy wybrać opcję *New File* z menu, a następnie wybrać *Block Diagram/Schematic File* w oknie dialogowym (**rysunek 4**). Utworzony plik należy zapisać pod nazwą głównej encji projektowej (w przykładzie *WitajSwiecie.bdf*). Plik ten zawiera składowe moduły projektu.

Do projektu układu FPGA należy dodać system cyfrowy z procesorem Nios II. Jak już wspomniano, służy do tego program SOPC Builder.



Rysunek 3. Utworzenie nowego projektu

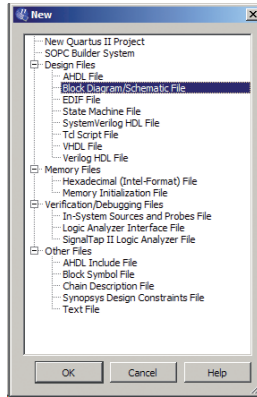
der, który uruchamiany jest poprzez wybranie ikony lub wybierając opcję *New File*, a następnie *SOPC Builder System*.

Przy pierwszym uruchomieniu (lub gdy poprzedni projekt nie został zapisany) należy podać nazwę nowego systemu cyfrowego (w przykładzie będzie to *nios_ep*) oraz wybrać język HDL, w którym zostaną wygenerowane pliki projektu. W kursie przykłady zostały przygotowane w języku Verilog, jednak nic nie stoi na przeszkodzie, aby w ramach jednego projektu używane były równocześnie pliki Verilog i VHDL.

Okno programu SOPC Builder (rysunek 5) jest podzielone na 4 główne sekcje:

- biblioteka bloków IP,
- wybór układu programowalnego i sygnały zegarowe systemu,
- mapa połączeń między blokami składowymi,
- okno informacyjne.

Głównym elementem systemu jest procesor Nios II. Należy wybrać go z biblioteki bloków IP (znajduje się on w grupie *Processors*) i nacisnąć przycisk *Add*. Przed dodaniem procesora do systemu zostanie uruchomiony kreator. Można w nim wprowadzić ustawienia konfiguracyjne, takie jak wersja rdzenia, rodzaj pamięci czy wybór własnych instrukcji. W zakładce *Core Nios II* należy wybrać wersję *Nios II/e* (rysunek 6). Jest to najprostsza wersja procesora Nios II, więc większość



Rysunek 4. Wybór pliku edytora schematów

ustawień opcjonalnych będzie nieaktywna. Przed zakończeniem pracy kreatora należy upewnić się, że w zakładce *JTAG Debug Module* wybrany jest debugger *Level 1*. Uaktywni to możliwość debugowania kodu programu oraz przesłania kodu programu do pamięci procesora.

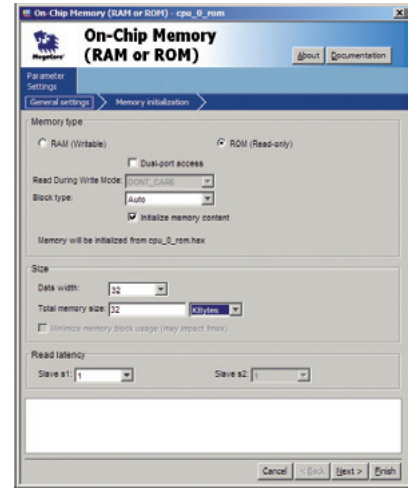
Po zakończeniu pracy kreatora przyciskiem *Finish*, w sekcji mapy połączeń programu SOPC Builder wyświetlony będzie nowo dodany procesor (widoczny na rysunku 5). W oknie informacyjnym widoczne są dwa komunikaty z podpowiedziami projektowymi (*To Do*) oraz ostrzeżenia (*Warning*). Informują one, że procesor nie ma zdefiniowanego adresu pamięci spod którego ma zacząć wykonywanie programu (*reset vector*) oraz adresu z wektorem przerwań (*exception vector*). Aby rozwiązać ten problem należy do procesora dołączyć pamięć RAM oraz ROM.

W bibliotece bloków IP dostępnych jest wiele kontrolerów pamięci zewnętrznych. Na potrzeby pierwszego projektu zostanie użyta pamięć wbudowana układu Cyclone II. Do projektowanego systemu należy dwukrotnie dodać pamięć On-Chip Memory (RAM or ROM). Ustawienia pamięci ROM przedstawiono na rysunku 7. Pojemności pamięci RAM i ROM to odpowiednio: 8 i 32 kB. Dobrą praktyką jest przyjęcie nazw pamięci tak, aby zawierały one nazwę procesora, dla którego są przeznaczone (np. w projekcie ustawiono nazwy *cpu_0_rom* i *cpu_0_ram*) lub były nazwą pełnionej funkcji (np. *buffer_ram*).

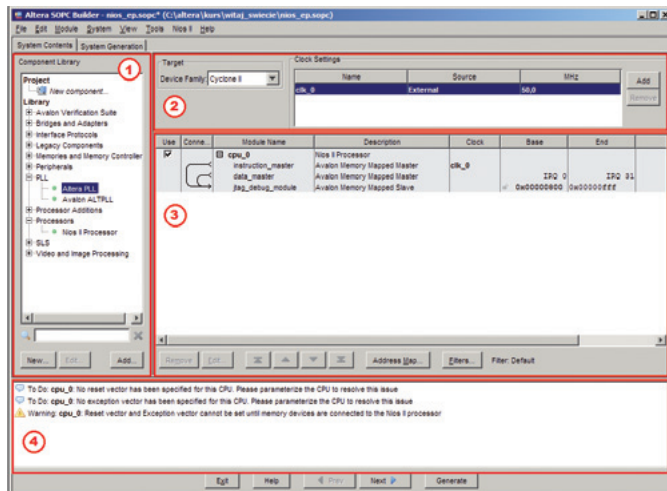
Po dodaniu pamięci w oknie informacyjnym pojawiła się informacja o błędnym adresie początkowym pamięci *cpu_0_rom*. Należy ją zamienić zgodnie z sugestią programu na *0x0008000*. W tym celu należy dwukrotnie kliknąć w kolumnie *Base* pole w wierszu pamięci *cpu_0_rom*. Następnie należy podać adresy startowe procesora *cpu_0*. W tym celu należy wywołać kreator konfiguracji modułu poprzez dwukrotne kliknięcie jego nazwy w oknie mapy połączeń (lub wybierając przycisk *Edit*). W zakładce *Core Nios II* należy ustawić pamięć *cpu_0_rom* jako pamięć dla wektora przerwań i obsługi wyjątków. Przesunięcie adresu *offset* należy pozostawić domyślne.

Do systemu należy dodać jeszcze timer systemowy (*Interval Timer*), który jest wymagany przez niektóre funkcje z warstwy programowej HAL procesorów Nios II. Jest on umieszczony w zakładce *Peripherals > Microcontroller Peripherals*. W ustawieniach bloku IP należy wybrać predefiniowane ustawienia *Full-Featured* oraz okres 1 ms (rysunek 8). Dodany timer należy nazwać *sys_clk_timer*.

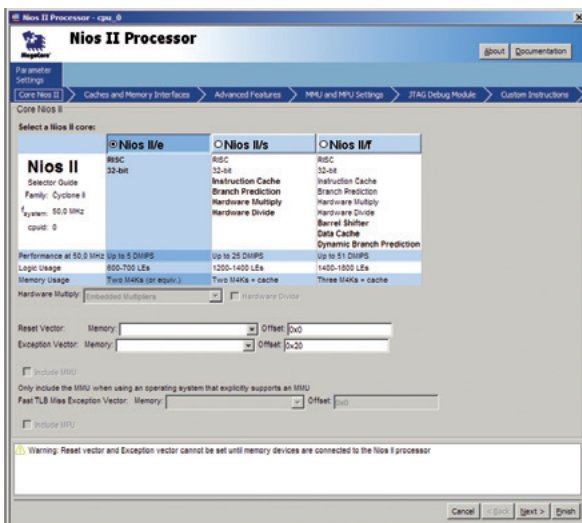
Do systemu należy też dodać blok *System ID* (*Pe-*



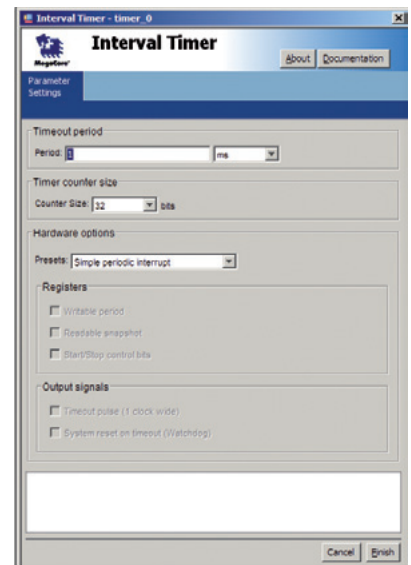
Rysunek 7. Ustawienia konfigurujące kontrolera pamięci wbudowanej układu FPGA



Rysunek 5. Okno programu SOPC Builder



Rysunek 6. Wybór wersji procesora Nios II



Rysunek 8. Ustawienia konfigurujące modułu timera

Use	Conn	Module Name	Description	Clock	Base	End
✓		cpu_0	Nios II Processor	clk_0		
		instruction_master	Avalon Memory Mapped Master		IP0_0	IP0_31
		data_master	Avalon Memory Mapped Master		0x00000000	0x00000fff
		jtag_debug_module	On-Chip Memory (RAM or ROM)		0x00000200	0x00003fff
✓		cpu_0_ram	Avalon Memory Mapped Slave	clk_0		
		s1	On-Chip Memory (RAM or ROM)		0x00000000	0x0000ffff
✓		cpu_0_rom	Avalon Memory Mapped Slave	clk_0		
		s1	On-Chip Memory (RAM or ROM)		0x00000000	0x0000ffff
✓		sys_clk_timer	System Clock	clk_0		
		int	Interrupt Controller		0x00000000	0x0000001f
✓		sysid	System ID Peripheral	clk_0		
		control_slave	Avalon Memory Mapped Slave		0x00000020	0x00000027
✓		jtag_uart	JTAG UART	clk_0		
		avalon_jtag_slave	Avalon Memory Mapped Slave		0x00000028	0x0000002f

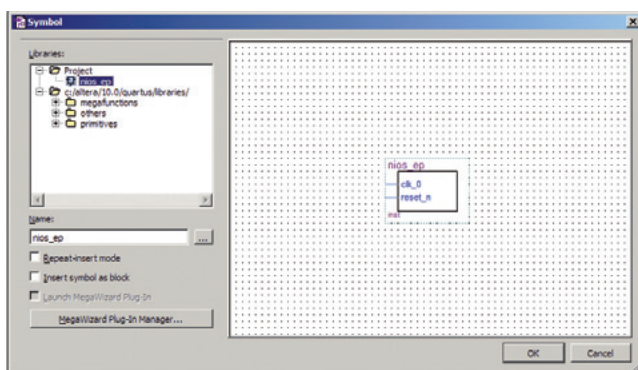
Rysunek 9. Mapa połączeń w projektowanym systemie cyfrowym

ipherals > Debug and Performance). Przechowuje on unikalny dla danego systemu cyfrowego ciąg liczb, który jest tworzony na etapie generowania plików systemu w programie SOPC Builder. Korzystając z niego narzędzia programowe firmy Altera. Zapobiegają one wgraniu kodu programu utworzonego dla innego systemu cyfrowego. Blok ten należy nazwać sysid.

Tak skonfigurowany projekt nie jest w pełni funkcjonalny, gdyż nie zdefiniowano żadnych urządzeń wejścia-wyjścia. Obsługa periferiów zewnętrznych zostanie omówiona w kolejnych częściach kursu. Na potrzeby pierwszego projektu należy dodać do systemu blok IP JTAG UART (Interface Protocols > Serial). Umożliwia on komunikację z systemem Nios II poprzez interfejs JTAG.

W miarę dodawania elementów do systemu cyfrowego można zauważyć, że w systemie występują dwie linie połączeń, których początki znajdują się przy procesorze Nios II, a końce w kolejnych modułach systemu. Każda z linii należy do osobnego modułu nadrzędnego (master) i reprezentuje możliwość komunikacji z modułami podrzędnymi (slave). Procesor Nios II ma dwie linie, gdyż ma on osobne połączenia dla magistrali danych i instrukcji.

Każdy moduł dołączony do magistrali Avalon ma przydzielony adres początkowy oraz końcowy. Zajętość przestrzeni adresowej danego modułu będzie zależała od liczby rejestrów danego modułu, a w przypadku pamięci – jej pojemności. Adresy początkowe można ustalać niemalże dowolnie, ze zwróceniem uwagi, aby adres końcowy nie nakładał się na przestrzeń adresową innego modułu. Na rysunku 9 przedstawiono mapę połączeń między modułami projektowanego systemu. W kolumnach Base i End widoczne są adresy początkowe i końcowe modułów.

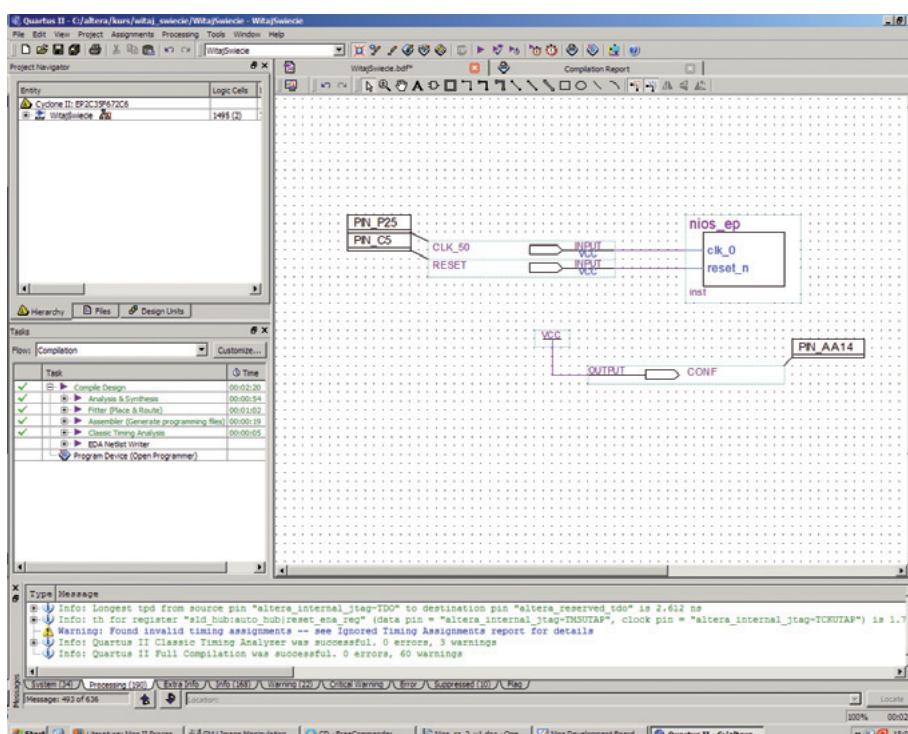


Rysunek 10. Dodawanie systemu cyfrowego do schematu w środowisku Quartus II

Jeżeli w projekcie występują jakiegokolwiek konflikty adresów, nie będzie możliwe wygenerowanie plików systemu. Aby wygenerować pliki systemu cyfrowego dla środowiska Quartus II należy przejść do zakładki System Generation i wybrać przycisk Generate. Po zakończeniu tworzenia plików należy zamknąć program. Należy zauważyć, że po wygenerowaniu systemu w programie SOPC builder można rozpocząć prace nad oprogramowaniem dla Procesora Nios II, gdyż program Nios II EDS do utworzenia bibliotek systemowych wymaga jedynie pliku z opisem wygenerowanego systemu (w przykładzie będzie to nios_ep.sopcinfo)

Tak utworzony system można dodać do projektu w edytorze schematowym. Dla systemu zostaje utworzony symbol graficzny, który można umieścić na schemacie podobnie jak elementy standardowe (rysunek 10).

Do zaprojektowanego systemu cyfrowego nios_ep należy dodać sygnały wejściowe (rysunek 11): sygnał zegarowy o częstotliwości 50 MHz (CLK_50) oraz sygnał z przycisku zerowania mikrokontrolera (RESET). W projekcie należy umieścić dodatkowo sygnał wyjściowy CONF dołączony na stałe do napięcia zasilania. Jest to podyktowane umieszczeniem



Rysunek 11. Schemat blokowy projektowanego systemu w programie Quartus II

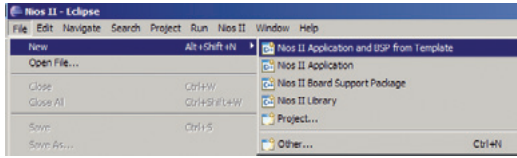
na płytce układem CPLD MAX7256, który pełni funkcję układu zarządzającego procesem konfigurowania układu FPGA Cyclone II. Gdy na wejściu układu występuje napięcie 0 V, to włączany jest mechanizm konfigurowania układu FPGA obrazem zapisanym w pamięci Flash. Układ ten może służyć m.in. do zdalnej zmiany projektu uruchamianego w układzie FPGA, czyli aktualizacji warstwy sprzętowej.

Gotowy projekt należy następnie skompilować i utworzyć plik dla pamięci konfiguracyjnej docelowego układu FPGA. Służy do tego przycisk Compile Design.

Wstęp do programowania Nios II

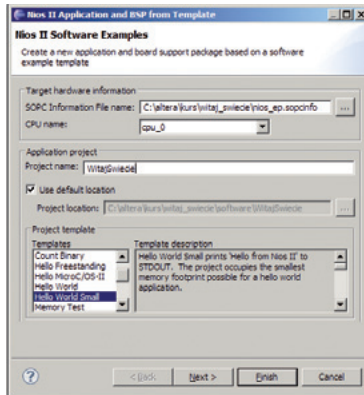
Firma Altera oferuje zintegrowane środowisko programistyczne Nios II EDS dla swoich procesorów typu softcore. Do programu Nios II EDS dołączane są: biblioteka programowa Nios II HAL, przykładowe programy oraz system operacyjny MicroC/OS-II. Nios II HAL jest biblioteką ułatwiającą programiście korzystanie z zasobów sprzętowych procesora Nios II.

Każdy program przygotowany dla procesora Nios II w środowisku EDS zawiera projekt aplikacji, opcjonalne projekty bibliotek



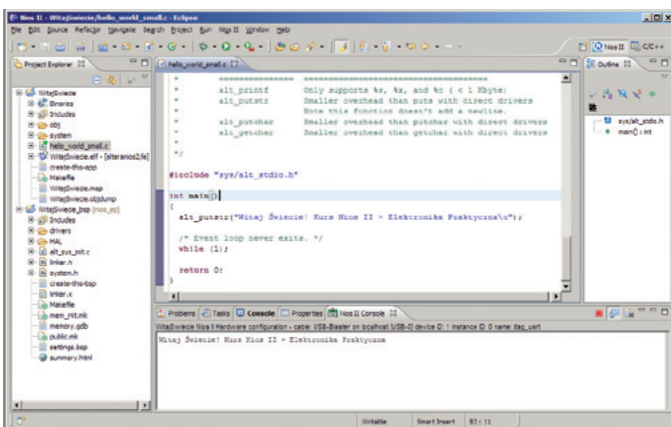
Rysunek 12. Wybór kreatora nowego projektu

programowych użytkownika oraz projekt BSP. W projekcie aplikacji przechowywane są pliki źródłowe tworzonego programu oraz generowany automatycznie plik *makefile* dla kompilatora. Projekt BSP zawiera m.in. biblioteki programowe HAL, opcjonalną bibliotekę standardową *newlib C* i sterowniki programowe. Projekt BSP jest związany z jednym procesorem z projektem systemu cyfrowego utworzonego w kreatorze SOPC Builder. Jeden projekt BSP może być używany w wielu programach. Możliwości biblioteki HAL oraz szczegóły implementacji programów dla procesora Nios II zostaną opisane w kolejnych częściach kursu. Na potrzeby uruchomienia przykładowej aplikacji dla utworzonego wcześniej projektu zostanie użyty przykładowy projekt dostępny w środowisku Nios II EDS.

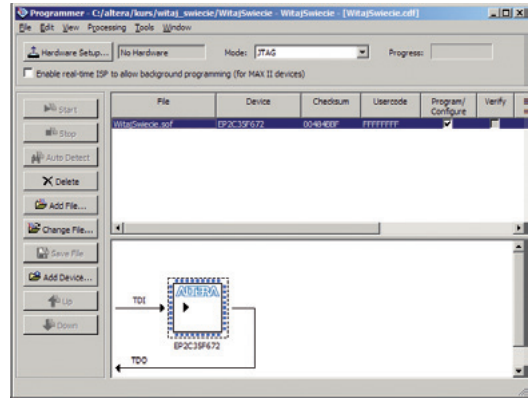


Rysunek 13. Okno wyboru przykładowego projektu

Aby utworzyć nowy program dla procesora Nios II należy wybrać opcję *Nios II Application and BSP from Template* z menu *File > New* (rysunek 12). W oknie dialogowym wyboru przykładowego projektu (rysunek 13) należy zaznaczyć przykład *Hello World Small*. Jest to przykładowy program korzystający ze standardowego wyjścia *STDOUT* do wypisania wiadomości. Środowisko Nios II EDS automatycznie wybierze jeden z dostępnych w systemie urządzeń znakowych – w przykładowym projekcie będzie to *JTAG UART*. Program ten korzysta z alterowskich odpowiedników funkcji *printf()*, *putstr()*, *putchar()* oraz *get_char()* z biblioteki standardowej. Funkcje z biblioteki *alt_stdio* korzystają bezpośrednio z zasobów sprzętowych i obsługują tylko część opcji ich odpowiedników. Uzyskano dzięki temu kilkukrotne zmniejszenie objętości kodu wynikowego programu. Oprócz nazwy projektu należy wskazać plik z opisem systemu cyfrowego utworzony przez kreator SOPC Builder. W opisywanym przykładzie plik *nios_ep.sopcinfo* (rysunek 13) znajduje się w głównym katalogu projektu. Jeżeli w systemie cyfrowym jest więcej niż jeden procesor Nios II, należy wybrać z listy ten, dla którego tworzony jest projekt.

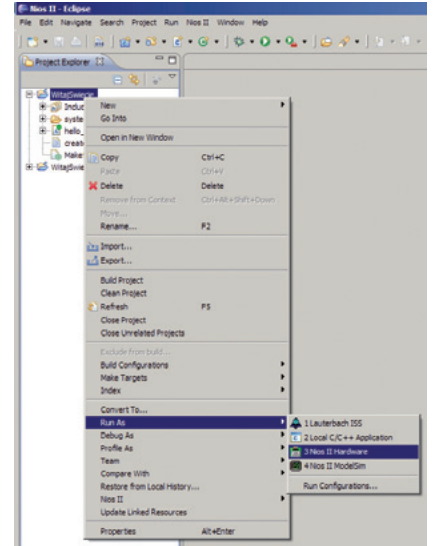


Rysunek 14. Okno środowiska programistycznego Nios II EDS




Rysunek 15. Widok okna programu Altera Programmer

Na rysunku 14 przedstawiono widok środowiska Nios II EDS z otwartym projektem *WitajSwiecie*. W głównej części okna programu znajduje się edytor kodu z otwartym plikiem źródłowym. Program zawiera tylko kilka linijek kodu: dołączenie pliku nagłówkowego *alt_stdio.h*, główną funkcję programu *main()* oraz wywołanie funkcji *alt_putstr()* która wysłała zdefiniowany ciąg znaków poprzez interfejs *JTAG UART*. Interfejs *JTAG UART* znakomicie spełnia funkcję terminala serwisowego oraz kanału zwrotnego podczas debugowania kodu programu. Komunikaty wysyłane przez ten interfejs są wyświetlane w konsoli Nios II (na rysunku 12 znajduje się poniżej edytora kodu).



Rysunek 16. Uruchomienie kodu programu w procesorze Nios II

Przed uruchomieniem kodu programu w procesorze Nios II należy wgrać do pamięci konfigurującej układ FPGA projekt systemu cyfrowego. Służy do tego programator firmy Altera. Można go uruchomić poprzez wybranie ikony  w środowisku Nios II EDS lub w programie Quartus II. Na rysunku 15 przedstawiono widok uruchomionego programu Altera Programmer. Program automatycznie wybierze odpowiedni plik konfigurujący.

Aby uruchomić program w procesorze Nios II należy wybrać opcję *Run As > Nios II Hardware* z menu *Run* lub z menu podręcznego projektu (rysunek 16). Przy pierwszym uruchomieniu uruchomiony zostanie proces kompilacji wszystkich plików projektu programu oraz projektu BSP co może zająć nawet kilkanaście minut. Przy ponownych zmianach kodu programu nie będzie potrzebna kompilacja projektu BSP, więc proces kompilacji trwa znacznie krócej.

Podsumowanie

W artykule przedstawiono podstawy tworzenia systemów cyfrowych przy użyciu programów projektowych firmy Altera. Opisa- no również tworzenie nowych projektów programów dla procesora Nios II na podstawie przykładowych programów udostępnionych przez firmę Altera. W następnej części kursu zostanie opisany bardziej szczegółowo proces tworzenia i debugowania oprogramowania dla procesorów Nios II.

Maciej Gołaszewski, EP
maciej.golaszewski@ep.com.pl