

Tajniki (zmiennoprzecinkowej) arytmetyki w FPGA

Układy FPGA nie są „same z siebie” wdzięczną platformą do wykonywania zaawansowanych – z natury rzeczy – obliczeń zmiennoprzecinkowych. Nie wynika to jednakże z ich ograniczonych zasobów logicznych, ale z konieczności „ręcznego” budowania w dostępnych sprzęcie bloków realizujących potrzebne obliczenia. Problem ten zauważyli inżynierowie z firmy Altera i żeby uprościć życie konstruktorom udostępnili (bezpłatnie!) w ramach pakietu Quartus II konfigurowalne biblioteki IP core’ów, dzięki którym m.in. realizacja zaawansowanych operacji arytmetycznych przestała stanowić problem.

System Quartus II jest zintegrowanym, łatwym w obsłudze, bardzo przyjaznym narzędziem projektowym, wyposażonym we wszystkie niezbędne edytory, kompilatory i syntezery oraz wspomniane na początku biblioteki z rdzeniami IP, które w oryginalnej

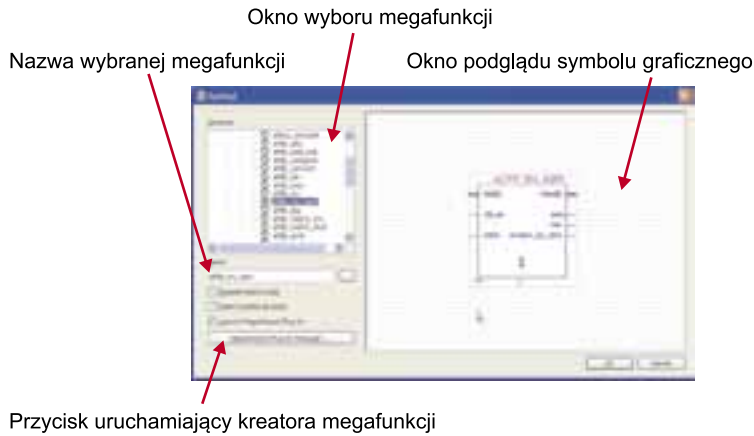
nomenklaturze Altery są nazywane „megafunkcjami” lub LPM (*Library Parametrized Module*). Listę megafunkcji dostępnych bezpłatnie można przejrzeć otwierając okno *Symbol* (pokazane na **rysunku 1**). W artykule skupimy się na megafunkcjach związanych

Dodatkowe informacje:
Testy sprzętowe niezbędne do opracowania artykułu przeprowadzono na zestawie Cyclone IV GX Transceiver Starter Kit udostępnionym przez firmę **EBV Elektronik**, tel. 71 342 29 44 oraz 22 640 23 55, www.ebv.com

z wykonywaniem obliczeń arytmetycznych, przede wszystkim zmiennoprzecinkowych, których nazwy zaczynają się od ALTFP_ (np. ALTFP_MATRIX_INV, ALTFP_EXP czy ALTFP_LOG).

Wygodne, przejrzyste i bezpłatne

Standardowo w pakiecie Quartus II znajduje się 13 megafunkcji zmiennoprzecinkowych (**tabela 1**), które mogą być przez projektanta w pewnym zakresie modyfikowane, dzięki czemu można je dostosować do wymogów



Rysunek 1. Okno *Symbol* z wykazem dostępnych bibliotek i elementów w nich zawartych

realizowanego projektu. Wszystkie operują na danych zmiennoprzecinkowych w formacie IEEE-754 z uwzględnieniem wartości specjalnych (w tym *Quiet* i *Signaling NaN*), w formatach pojedynczej i podwójnej precyzji, a także



Rysunek 2. Pierwsze okno kreatora z możliwością wyboru języka opisu generowanego bloku

pojedynczej-rozszerzonej (szerokość słowa bitowego mieści się w przedziale od 43 do 64 bitów). Dzięki takiemu rozwiązaniu możliwe jest



Rysunek 3. Drugie okno kreatora umożliwia wybór formatu danych wejściowych i wyjściowych oraz liczby taktów opóźnienia generowania wyniku (jak na rysunku 4)

zoptymalizowanie zasobów wykorzystywanych przez implementowany moduł funkcjonalny w zależności od rzeczywistych potrzeb aplikacji – megafunkcje zawierają wyłącznie parametryzowane opisy modułów, które są każdorazowo syntezowane w projekcie. W odróżnieniu od zapisów w IEEE-754, megafunkcje Altery obsługują tylko jeden typ zaokrąglania wyników (*round-to-nearest-even*), najczęściej stosowany w typowych obliczeniach zmiennoprzecinkowych.

Ze względu na stopień skomplikowania megafunkcji realizujących obliczenia zmiennoprzecinkowe, są to bloki synchroniczne, wymagające taktowania sygnałem zegarowym. Wykorzystując możliwości układów FPGA (duże zasoby logiczne) większość obliczeń realizują one równolegle, ale zazwyczaj liczba taktów zegarowych niezbędnych do wykonania kompletnego cyklu obliczeniowego jest większa niż 1. Powoduje to pojawienie się wyniku obliczeń na wyjściach modułu z pewnym opóźnieniem, które użytkownik może (jako jeden z parametrów) w pewnym zakresie modyfikować. Parametryzacji przez użytkownika podlegają także niektóre wejścia (jak na przykład zerujące, włączające sygnał taktujący) oraz wyjścia (jak *overflow*, *zero* czy *nan*), co także pozwala zoptymalizować zasoby zajmowane w FPGA przez implementowany blok.

Przykład konfiguracji megafunkcji

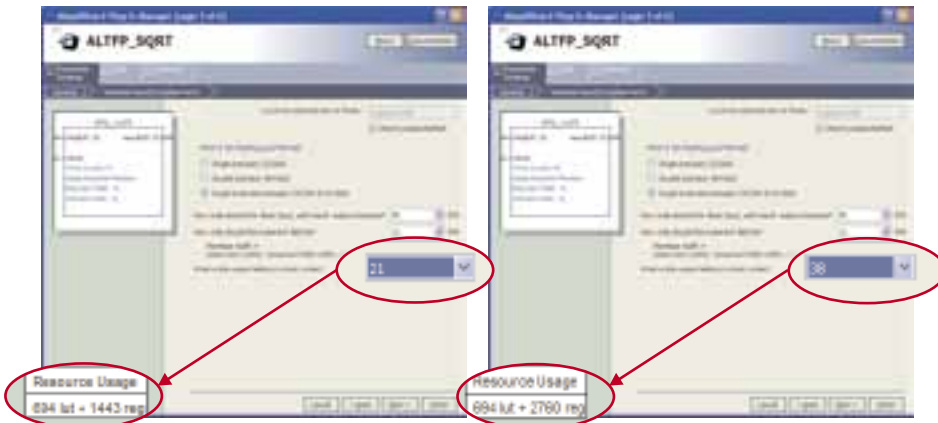
Parametryzację prezentowanych megafunkcji można przeprowadzić za pomocą „ręcznego” wywołania funkcji z parametrami podanymi w wywołaniu – co jest metodą dość żmudną i błędogenną – lub za pomocą kreatora wbudowanego w pakiet Quartus II (*MegaWizard Plug-in Manager*). Kreator można uruchomić ręcznie za pomocą przycisku widocznego w dolnej lewej części rysunku 1 lub automatycznie (przy zaznaczonej opcji *Launch MegaWizard Plug-in Manager*) po wybraniu megafunkcji, która będzie modyfikowana.

Po wybraniu megafunkcji i zatwierdzeniu wyboru przyciskiem OK, jest wyświetlane pierwsze okno kreatora (rysunek 2), w którym można wybrać język opisu generowanego bloku oraz lokalizację tego pliku. W kolejnym kroku (rysunek 3) projektant może ustalić typ docelowego układu FPGA (jeżeli ma być inny niż zadeklarowany w projekcie), format danych wejściowych i wyjściowych, a także – opcjonalnie – liczbę bitów tworzących słowo wejściowe. Na tym etapie można ustalić także liczbę taktów zegarowych, po których wynik pojawia się na wyjściu bloku, co determinuje także zajmowane przez niego zasoby logiczne (rysunek 4).

W kolejnym kroku można wybrać dodatkowe wejścia i wyjścia implementowanego bloku, co odbywa się za pomocą wygodnego w obsłudze menu wyświetlanego w oknie kreatora (rysunek 5).

Tabela 1. Zmiennoprzecinkowe megafunkcje dostarczane bezpłatnie w pakiecie Quartus II

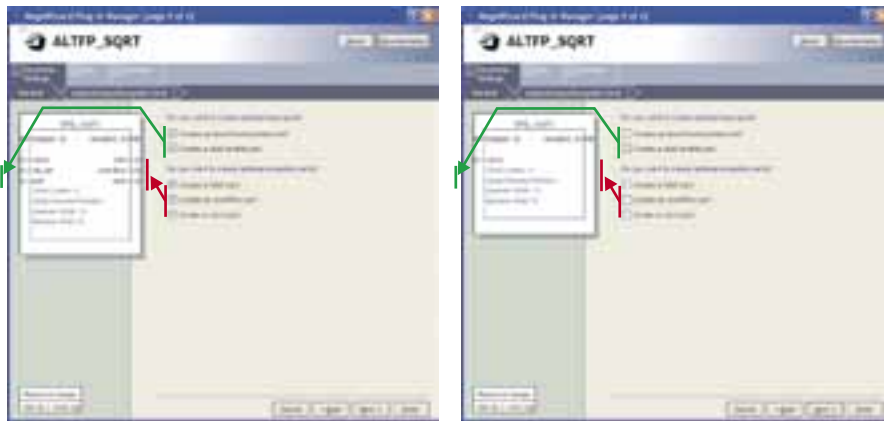
Nazwa	Opis
Operatory zmiennoprzecinkowe	
ALTFP_ADD_SUB	Dodawanie i odejmowanie
ALTFP_DIV	Dzielenie
ALTFP_MULT	Mnożenie
ALTFP_SQRT	Pierwiastkowanie stopnia 2
Zaawansowana algebra	
ALTFP_EXP	Oblicza eksponentę wartości wejściowej
ALTFP_INV	Oblicza wartość inwersyjną wartości wejściowej
ALTFP_INV_SQRT	Oblicza wartość inwersyjną pierwiastka stopnia 2 wartości wejściowej
ALTFP_LOG	Logarytm naturalny
Inne działania zmiennoprzecinkowe	
ALTFP_ABS	Oblicza wartość całkowitą wartości wejściowej
ALTFP_COMPARE	Porównuje liczby zmiennoprzecinkowe
ALTFP_CONVERT	Konwertuje wartości: <ul style="list-style-type: none"> – całkowite do postaci zmiennoprzecinkowej – zmiennoprzecinkowe do postaci całkowitej – stałoprzecinkowe do postaci zmiennoprzecinkowej – zmiennoprzecinkowe do postaci stałoprzecinkowej – zmiennoprzecinkowe do innych wariantów zmiennoprzecinkowych
Operacje na macierzach	
ALTFP_MATRIX_INV	Oblicza inwersję macierzy o wymiarach do 64×64
ALTFP_MATRIX_MULT	Oblicza iloczyn dwóch macierzy



Rysunek 4. Modyfikacja liczby cykli zegarowych znajduje silne odbicie w zasobach zajmowanych przez implementowany blok

Przedostatni krok działania kreatora (rysunek 6) zapewnia wygodny dostęp do dokumentacji implementowanej megafunkcji (przycisk *Documentation*), jej skróconego opisu (*Abort* – rysunek 7) z wykazem rodzin FPGA, w których może być ona implementowana. Na tym etapie można zażądać także wygenerowania *netlisty*, która pozwoli z dość dużą dokładnością oszacować parametry czasowe (w tym maksymalną częstotliwość taktowania) bloku zaimplementowanego w wybranym układzie FPGA.

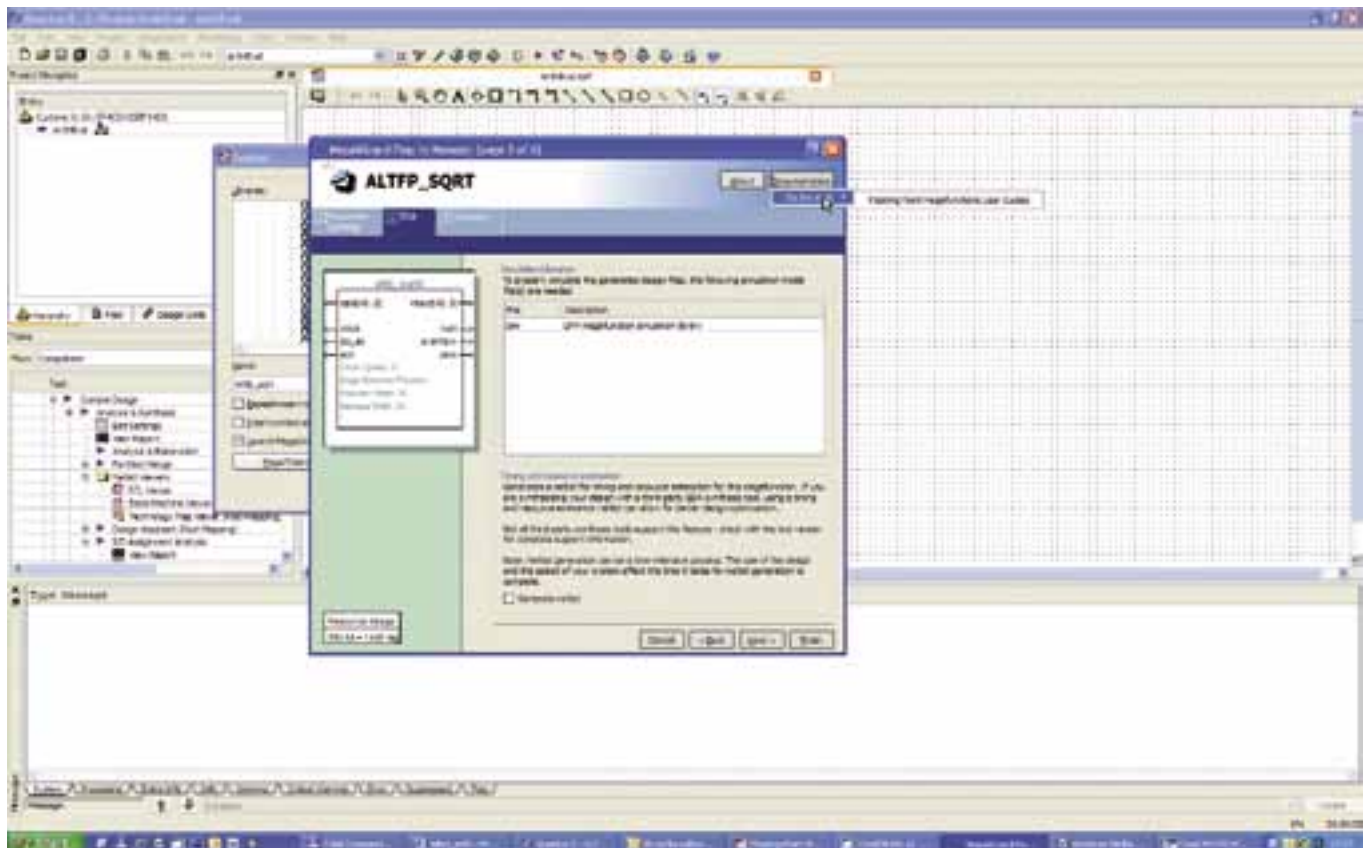
Na rysunku 8 pokazano ostatnie okno kreatora megafunkcji, w którym wyświetlono podsumowanie wyboru generowanych plików (niektóre z nich można zmienić), po



Rysunek 5. W kolejnym kroku pracy kreatora można wybrać dodatkowe wejścia i wyjścia implementowanego bloku, co skutkuje zmianą wyglądu symbolu graficznego elementu



Rysunek 7. Skrócony opis cech implementowanego bloku logicznego



Rysunek 6. W przedostatnim kroku działania kreatora można zażądać wygenerowania *netlisty*, która pozwoli z dość dużą dokładnością oszacować parametry czasowe implementowanego bloku



Rysunek 8. W ostatnim oknie kreatora widać podsumowanie wyboru generowanych plików implementacyjnych

zatwierdzeniu których w katalogu projektu tworzony jest nowy blok funkcjonalny (rysunek 9), który można wykorzystać zarówno w edytorze graficznym jak i w opisie tekstowym, dzięki automatycznie wygenerowanemu opisowi komponentu (listing 1).

Podsumowanie

Biblioteki z predefiniowanymi opisami bloków funkcjonalnych są niezwykle wygodnym sposobem skrócenia czasu realizacji projektu w FPGA, podobnie jak ma to miejsce w przypadku bibliotek wykorzystywanych przez programistów piszących na mikrokontrolery lub systemy mikroprocesorowe. Przykłady megafunkcji pokazanych w artykule ilustrują łatwość zaawansowanego operowa-

nia za pomocą FPGA na liczbach zmiennoprzecinkowych oraz macierzach takich liczb, co jest wstępem do zaawansowanej filtracji i cyfrowej obróbki sygnałów, czego przykłady pokażemy w jednej z kolejnych EP.

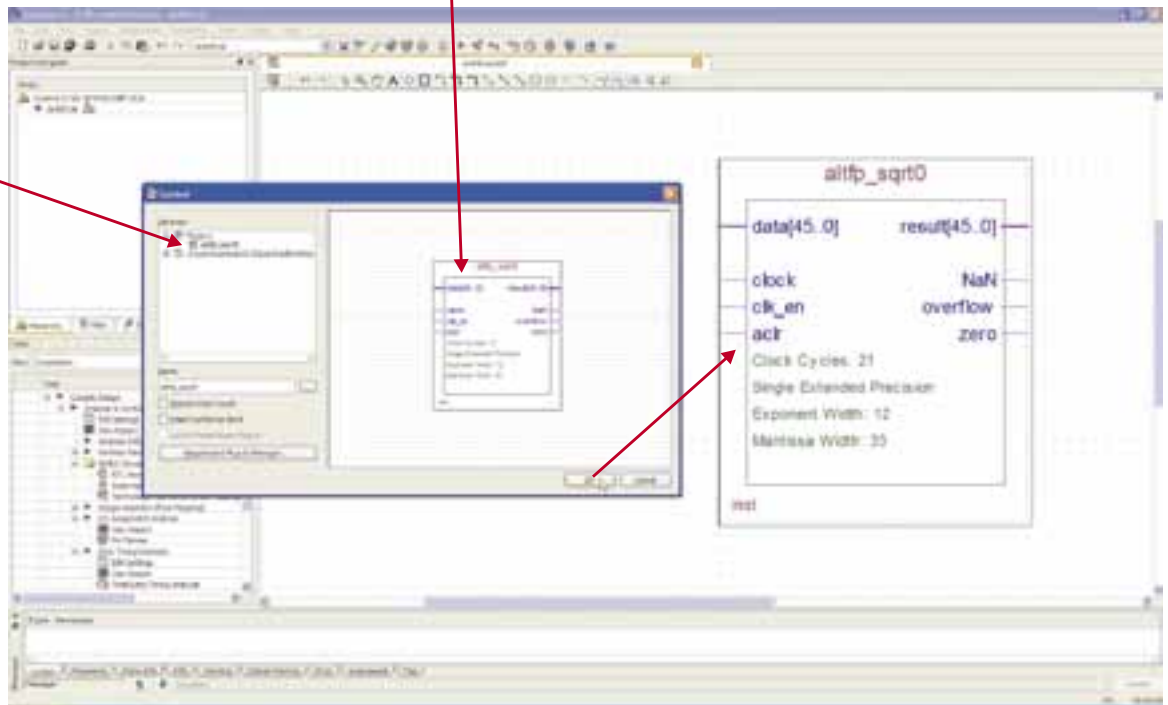
Piotr Zbysiński, EP
piotr.zbysinski@ep.com.pl

Listing 1. Sygnały wykreowanego komponentu w opisie VHDL

```
altfp_sqrt0_inst : altfp_sqrt0 PORT
MAP
(
  aclr          => aclr_sig,
  clk_en       => clk_en_sig,
  clock        => clock_sig,
  data         => data_sig,
  nan          => nan_sig,
  overflow     => overflow_sig,
  result       => result_sig,
  zero        => zero_sig
);
```

Symbol graficzny wykreowanego symbolu można wykorzystać w edytorze schematów

Wykreowana megafunkcja jest automatycznie dodawana do bibliotek projektu



Rysunek 9. Widok listy elementów bibliotecznych po wygenerowaniu własnej wersji bloku oraz jego symbolu graficznego na planszy edytora schematów

R E K L A M A

RK-SYSTEM
www.rk-system.com.pl

Profesjonalne narzędzia dla elektroników i programistów

- uniwersalne programatory układów scalonych
- analizatory stanów logicznych
- oscyloskopy cyfrowe
- systemy do wyważania i pomiaru drgań
- oprogramowanie CAD, CAM, CAE
- emulatorzy, symulatory, debuggery dla różnych rodzin procesorów
- kompilatory C/C++ dla różnych rodzin procesorów
- szkolenia w zakresie FPGA, VHDL
- narzędzia na procesory sygnałowe DSP
- projektujemy, produkujemy, szkolimy, dystrybuujemy

15-825 Grodzisk-Nieś, ul. Cieszańskiego 20, tel. (522) 734 30 39, 792 05 19, fax (522) 734 30 37

FAISONANCE
IAR SYSTEMS