Oleg MASLENNIKOW [1], **Piotr RATUSZNIAK** [1], **Anatoli SERGIYENKO** [2]
[1] POLITECHNIKA KOSZALIŃSKA, WYDZIAŁ ELEKTRONIKI I INFORMATYKI
[2] POLITECHNIKA KIJOWSKA, WYDZIAŁ TECHNIKI OBLICZENIOWEJ I INFORMATYKI

Generator opisów VHDL bloków operacyjnych działających w arytmetyce ułamkowej

Dr hab. inż. Oleg MASLENNIKOW

Pracuje na stanowisku profesora nadzwyczajnego na Wydziale Elektroniki i Informatyki, Politechnika Koszalińska. Jego zainteresowania naukowe to akceleracja obliczeń, architektura komputerów, systemy czasu rzeczywistego.

e-mail: oleg@ie.tu.koszalin.pl

Mgr inż. Piotr RATUSZNIAK

Pracuje na stanowisku asystenta na Wydziale Elektroniki i Informatyki, Politechnika Koszalińska. Jego zainteresowania naukowe to projektowanie wyspecjalizowanych architektur równoległych, optymalizacja z wykorzystaniem algorytmów genetycznych, bazy danych oraz programowanie.

e-mail: ratusz@ie.tu.koszalin.pl

Streszczenie

W niniejszej pracy przedstawiono generator opisów VHDL potokowych bloków operacyjnych działających w arytmetyce ułamkowej (RFA) i przeznaczonych do implementacji w nowoczesnych układach FPGA, mających wbudowane bloki mnożące i/lub DSP. Badania autorów świadczą o mniejszej złożoności sprzętowej jednostek arytmetycznych RFA, wykonujących operacje dodawania i/lub mnożenia i/lub dzielenia w porównaniu z analogicznymi jednostkami operującymi na liczbach stałoprzecinkowych (przy zachowaniu wymaganej dokładności i wydajności obliczeń). Podstawowymi parametrami generatora są: rodzaj operacji arytmetycznej, szerokość danych wejściowych i wyjściowych oraz liczba stopni w potoku.

Słowa kluczowe: systemy czasu rzeczywistego SoC, arytmetyka ułamkowa RFA, język opisu sprzętu VHDL, generator IP Core, układy reprogramowalne FPGA, bloki DSP.

Rational fraction arithmetic unit IP-core generator

Abstract

In this paper, the IP-core generator is proposed, which produces the VHDL description of the arithmetic units operating in rational fraction arithmetic (RFA). Due to RFA, the hardware complexity of the new arithmetic units, which must perform for example the addition or multiplication or division operations, is much lower in comparison with complexity of the similar fixed-point arithmetic units (with the same precision and performance). The architectures of the target RFA units are pipelined and are adapted to the internal structure of the modern reconfigurable devices (like to Xlinx Virtex 4 or Altera Sratix II devices), and use the built-in 18-bit multipliers or DSP blocks. The main tuned parameters of the proposed soft-generator are the type of arithmetic operation, for example addition, multiplication, division, square rooting, RFA to fixed-point format conversion (see tab. 2), the input and output data width, as well as the number of the pipeline stages in the target arithmetic unit.

Keywords: system on chip, rational fraction arithmetic, VHDL, FPGA, DSP.

Dr Anatoli SERGIYENKO

Pracuje na stanowisku starszy pracownik naukowy na Politechnice Kijowskiej, Wydziale Informatyki i Techniki obliczeniowej (Ukraina). Jego zainteresowania naukowe to architektura komputerów, systemy czasu rzeczywistego oraz równoległe i potokowe przetwarzanie danych.

e-mail: aser@comsys.ntu-kpi.kiev.ua

1. Wprowadzenie

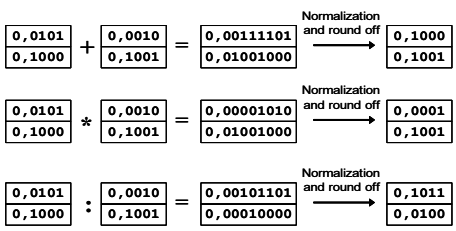
Nowoczesne reprogramowalne układy cyfrowe FPGA zawierają setki tysięcy rekonfigurowalnych komórek [1], wbudowane bloki pamięci, dziesiątki a nawet setki wbudowanych bloków mnożących lub bloków DSP, jak również wbudowane rdzenie mikroprocesorów. Tak rozwinięta technologia VLSI pozwala na umieszczenie w pojedynczym układzie reprogramowalnym całego systemu jednoukładowego SoC (*ang.* System-on-Chip). Trudnym zadaniem staje się jednak efektywne zagospodarowanie tak dużych zasobów sprzętowych [1], tj. zaprojektowanie systemu SoC bezbłędnie i w rozsądnym czasie, przy równoczesnym zachowaniu pożądanej wydajności i funkcjonalności układu oraz minimalnym poborze mocy. W związku z tym, nowoczesne tendencje w projektowaniu systemów jednoukładowych są ukierunkowane m. in. na wykorzystanie gotowych projektów (IP-core) dla poszczególnych bloków systemu [2, 3] oraz na automatyzację procesu projektowania i weryfikacji projektu na wszystkich poziomach [4]. Nowoczesne narzędzia programowe do syntezy pozwalają projektantowi zdecydować o wykorzystaniu (lub nie) wbudowanych bloków DSP, mnożących, pamięci BRAM nawet bez konieczności znajomości ich budowy i zasad działania. Jednak w przypadku optymalizacji projektowanych jednostek przetwarzających np. pod względem wydajności (która w dużym stopniu zależy od maksymalnej możliwej częstotliwości działania systemu) należy brać pod uwagę budowę i zasady działania wbudowanych bloków lub wykorzystywać gotowe opisy jednostek przetwarzających i bloków operacyjnych, wybierając te opisy z istniejących bibliotek lub z odpowiednich generatorów IP-core. Autorzy niniejszej pracy w ramach badań nad wykorzystaniem arytmetyki ułamkowej RFA (*ang.* Rational Fraction Arithmetic) w równoległych i potokowych jednostkach przetwarzających SoC przeznaczonych do realizacji w nowoczesnych układach FPGA postanowili opracować własny generator soft-core, generujący opisy w języku VHDL podstawowych bloków operacyjnych, działających w arytmetyce RFA. Wyniki testowania otrzymanych modeli VHDL bloków RFA świadczą o ich ponad dwukrotnie mniejszej złożoności sprzętowej w porównaniu do analogicznych bloków operacyjnych działających na liczbach stałoprzecinkowych (przy zachowaniu wymaganej dokładności i wydajności obliczeń).

2. Arytmetyka ułamkowa

Algorytmy algebry liniowej, mimo swojej względnie dużej złożoności obliczeniowej, są coraz częściej stosowane w systemach cyfrowego przetwarzania sygnałów, szczególnie w przypadkach zaawansowanego przetwarzania sygnałów o dużej rozdzielczości. Nowoczesne układy FPGA z wbudowanych blokami DSP mogą być efektywną platformą sprzętową dla realizacji takich systemów. Jednak realizacja większości algorytmów algebry liniowej

wymaga wykonania, oprócz podstawowych operacji arytmetycznych jak np. mnożenie i dodawanie, także operacji dzielenia, a niekiedy nawet pierwiastkowania. W przypadku realizacji operacji mnożenia i dodawania w arytmetyce stałoprzecinkowej w układach FPGA, zwykle są wykorzystywane wbudowane w te układy bloki mnożące (lub bloki DSP). Problem stanowi efektywna implementacja operacji dzielenia, podczas której w/w bloki nie mogą być wykorzystane [6]. Mianowicie, realizacja szybkich bloków dzielenia, jak również bloków operacyjnych działających na liczbach zmiennoprzecinkowych, wymaga ogromnych zasobów układu reprogramowalnego [5]. W celu przezwyciężenia w/w problemu autorzy proponują, w przypadku realizacji jednostek arytmetycznych wykonujących m. in. operacje dzielenia w układach reprogramowalnych, wykorzystywać w tych jednostkach arytmetykę ułamkowa [6]. Jednym z powodów zastosowania arytmetyki ułamkowej jest właśnie efektywna implementacja operacji dzielenia, która sprowadza się do wykonania dwóch operacji mnożenia, które są efektywnie realizowane w oparciu o wbudowane bloki mnożące lub bloki DSP. Ponadto złożoność bloku mnożącego w arytmetyce ułamkowej jest od 2 do 4 razy mniejsza, a jego opóźnienie - nawet 2 razy mniejsze w porównaniu do odpowiedniego bloku mnożącego działającego w arytmetyce stałoprzecinkowej. Porównanie złożoności sprzętowej bloku dzielenia n-bitowych ułamków z blokiem dzielenia dwóch $2n$-bitowych liczb w formacie stałoprzecinkowym wypada jeszcze lepiej, ponieważ sprowadza się ono w RFA do wykonania dwóch operacji mnożenia liczb n-bitowych. Zaletą wykorzystania arytmetyki ułamkowej na platformach FPGA jest również dokładność wykonywanych obliczeń. W arytmetyce ułamkowej każda liczba jest przedstawiona w postaci ułamku wymiernego a/b, gdzie a i b są n-bitowe liczby całkowite, $b \neq 0$. Z tego powodu dla przechowania liczb potrzebne są $2n$-bitowe komórki pamięci i rejestry. Jednak dokładność takiej $2n$-bitowej liczby jest nie mniejsza (a często większa) od dokładności zapisu tejże liczby w $2n$-bitowym formacie stałoprzecinkowym (jak i zmiennoprzecinkowym) [6, 7]. Również w celu porównania dokładności arytmetyki ułamkowej i stałoprzecinkowej, członkowie zespołu badawczego opracowali 2 modele VHDL generatora funkcji $\sin x$ i $\cos x$. Pierwszy generator działał na 32-bitowych danych stałoprzecinkowych i generował na wyjściu sinusoidę, której amplituda jednak ciągle się zmniejszała z powodu błędów zaokrągleń, i po 47 tys. iteracji okazała się równa zero (zamiast 1). Drugi generator działał na 20-bitowych ułamkach w arytmetyce RFA i generował sinusoidę, której amplituda po 256 tys. iteracjach wyniosła 1,029 (zamiast 1) [8].

Arytmetyka ułamkowa posiada również wady, które jednak, zdaniem autorów, można w dużym stopniu zredukować. Pierwszą wadą jest to, że ponieważ licznik i mianownik w jednostkach przetwarzających RFA są przedstawione w postaci liczb całkowitych, po wykonaniu, np. operacji mnożenia dwóch ułamków n-bitowych a/b i c/d powstaje ułamek $2n$-bitowy $a \cdot c \,/\, (b \cdot d)$, tj. liczba bitów w otrzymywanych wynikach będzie ciągle wzrastać. Wyeliminować tę wadę można, zdaniem autorów, poprzez normalizację wyniku i jego zaokrąglenie do ułamku n-bitowego po wykonaniu każdej operacji, przy czym normalizacja w tym przypadku odbywa się poprzez przesunięcie w lewo licznika i mianownika o jednakową liczbę bitów (rys. 1).

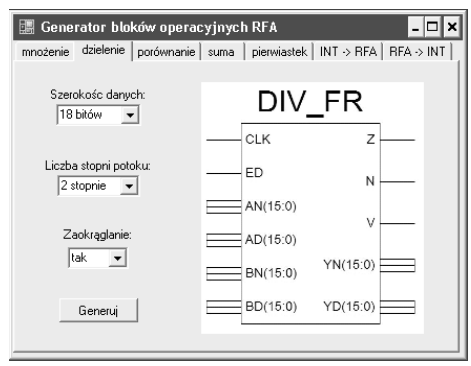


Rys. 1. Przykłady realizacji operacji normalizacji i zaokrąglenia
Fig. 1. Examples of normalization and round off operations

Kolejną wadą arytmetyki ułamkowej jest dość skomplikowana realizacja operacji dodawania (odejmowania), która wymaga wykonania 3 operacji mnożenia. Jednak trudno sobie wyobrazić jednostkę arytmetyczno-logiczna (ALU), która jest przeznaczona do realizacji wyłącznie operacji dodawania. Badania autorów świadczą o tym, że jednostki arytmetyczno-logiczne (ALU) RFA, mają znacznie mniejszą złożoność sprzętową, w porównaniu do odpowiednich ALU działających w arytmetykach stało i zmiennoprzecinkowych [9]. Na przykład zaprojektowana przez autorów jednostka RFA, przeznaczona do wykonania operacji typu dodawanie, mnożenie, mnożenie z dodawaniem, dzielenie oraz dzielenie z dodawaniem na n-bitowych ułamkach w kodzie uzupełnieniowym do dwóch zawiera pięć n-bitowych bloków mnożących, jeden n-bitowy sumator i dwa bloki normalizacji wyniku. Wyniki implementacji zaprojektowanego 35-bitowego ALU w układzie Virtex4 świadczą o ponad trzykrotnie mniejszej liczbie wykorzystanych bloków CLB w porównaniu do znanych ALU stałoprzecinkowych wykonujących podobny zbiór operacji z analogiczną dokładnością (przy porównywalnej maksymalnej częstotliwości zegara systemowego).

3. Generator bloków operacyjnych

Prezentowany generator tworzy opis w języku VHDL jednego z następujących (wybranych przez użytkownika) bloków operacyjnych działających w arytmetyce ułamkowej: mnożenia, dzielenia, dodawania, porównania, pierwiastkowania, konwersji liczb z formatu stałoprzecinkowego do formatu ułamkowego i odwrotnie. Wszystkie modele VHDL w/w bloków operacyjnych mogą być wygenerowane dla standardowych wartości szerokości danych wejściowych i wyjściowych 24 i 32 bitów oraz (uwzględniając możliwości wbudowanych bloków mnożących i bloków DSP w układach FPGA firmy Xilinx) 18 i 35 bitów. Wprowadzono również możliwość wyboru liczby stopni potoku dla projektowanego bloku operacyjnego, co ma duży wpływ na maksymalną częstotliwość działania jednostki ALU, tj. na jej wydajność, co jest szczególnie ważne przy wykorzystywaniu projektowanych bloków w systemach czasu rzeczywistego. Generator umożliwia również włączenie operacji zaokrąglania wyników cząstkowych po wykonaniu operacji normalizacji. Interfejs użytkownika zaprojektowanego generatora bloków operacyjnych przedstawiony jest na rys. 2. Generator został napisany w języku C# z wykorzystaniem środowiska Visual Studio 2008 oraz platformy Microsoft . NET Framework 3.5, która jest niezbędna do jego uruchomienia.



Rys. 2. Interfejs graficzny zaprojektowanego generatora bloków operacyjnych.
Fig. 2. Graphic interface of designed operations blocks generator

4. Parametry wygenerowanych modeli VHDL bloków operacyjnych RFA

Proces syntezy i implementacji bloków operacyjnych RFA został przeprowadzony dla układów FPGA Xilinx4 xc4vSX35-12 w środowisku Xilinx ISE 9.2.04i oraz FPGA Altera StratixII Eps215F672C5 w środowisku Quartus II 7.2 SP2 Web Edition.

Proces syntezy ukierunkowano na optymalizację szybkości działania projektowanych jednostek z automatycznym wykorzystywaniem bloków DSP. W celu porównania wyników warto podkreślić, że pojedynczy blok DSP dla wymienionego układu Xilinx Virtex4 zawiera dwa 18-bitowe bloki mnożące [2], natomiast bloki DSP w układach Altera Stratix II zawierają cztery takie bloki mnożące [3]. Tabele tab. 1. oraz tab. 2. przedstawiają najważniejsze parametry projektowanych bloków operacyjnych, mianowicie liczbę wykorzystanych komórek CLB (dla układu Xilinx'a) lub ALUT(dla układu Altery), liczbę wykorzystanych bloków DSP oraz maksymalną częstotliwość działania.

Tab. 1. Parametry wygenerowanych modeli bloków operacyjnych dla układu xc4vSX35-12
Tab. 1. The main parameters of the RFA processing units implemented in the Xilinx xc4vSX35-12 device

Nazwa modułu	Liczba CLB slices + 18-bitowych bloków DSP przy szerokości danych (bit)				Max częst., MHz przy szer. danych			
	18	**24**	**32**	**35**	**18**	**24**	**32**	**35**
ADD_FR	200 +3	311 +12	372 +12	528 +12	241	170	166	149
CMP_FR	7 +2	70 +8	80 +8	84 +8	484	181	181	181
MUL_FR	188 +2	195 +8	273 +8	402 +8	220	181	181	176
DIV_FR	188+2	195 +8	273 +8	402 +8	220	181	181	176
SQRT_FR	544 +5	724 +20	959 +20	1262 +20	270	150	147	146
INT2FR	96	122	167	197	384	327	311	326
FR2INT	555	921	1579	1827	191	173	153	148

Tab. 2. Parametry wygenerowanych modeli bloków operacyjnych dla układu Eps215F672C5
Tab. 2. The main parameters of the RFA processing units implemented in the Altera Eps215F672C5 device

Nazwa modułu	Liczba bloków ALUT + 9-bitowych bloków DSP przy szer. danych (bit)				Max częst., MHz przy szer. danych			
	18	**24**	**32**	**35**	**18**	**24**	**32**	**35**
ADD_FR	226 +6	303 +24	396 +24	500 +24	172	146	137	125
CMP_FR	45 +4	62 +16	82 +16	90 +16	270	227	183	169
MUL_FR	132 +4	175 +16	224 +16	352 +16	225	220	196	174
DIV_FR	132 +4	175 +16	224 +16	352 +16	225	220	196	174
SQRT_FR	414 +10	532 +40	1228 +40	1680 +40	162	124	124	121
INT2FR	83	114	190	212	289	268	215	208
FR2INT	629	1026	1773	2042	138	117	102	90

Na podstawie powyższych wyników można wywnioskować, że w obu układach FPGA wykorzystanych zostaje identyczna liczba wbudowanych 18-bitowych bloków mnożących zawartych w blokach DSP (blok DSP w rodzinie Virtex4 zawiera 2 18-bitowe bloki mnożenia, w rodzinie StratixII - 4 takie bloki), natomiast dla układu Virtex4 uzyskano wyższe częstotliwości pracy, niż dla układu StratixII. Warto zauważyć, że blok dzielenia zajmuje identyczne zasoby co blok operacji mnożenia, gdyż w arytmetyce ułamkowej operacja dzielenia sprowadza się do operacji mnożenia (przez odwrotność drugiego operandu operacji). Powyższe dane mogą być pomocne przy wyborze rodziny i modelu układu FPGA, w którym będzie realizowany projektowany system. Dodatkowo autorzy zbadali, jaki wpływ na liczbę wykorzystanych bloków CLB i częstotliwość pracy jednostek arytmetycznych RFA ma wykorzystanie wbudowanych w układ Xilinx xc4vSX35-12 bloków DSP48. Zgodnie z przewidywaniami (wyniki porównania są przedstawione w tab. 3) wykorzystanie wbudowanych bloków DSP powoduje wykorzystanie znacznie mniejszej liczby bloków CLB oraz zwiększenie maksymalnej częstotliwości pracy bloku operacyjnego RFA.

Tab. 3. Parametry 18-bitowych bloków operacyjnych FRA implementowanych w układzie xc4vSX35-12 z i bez wykorzystania wbudowanych bloków DSP
Tab. 3. The main parameters of the 18-bit RFA processing units implemented in the Xilinx xc4vSX35-12 device with and without built-in DSP blocks.

Nazwa modułu	Stopnie potoku	CLB slices + l. bloków DSP	Max częst. (MHz)
ADD_FR	5	200+3 866+0	241 140
CMP_FR	4	7+2 479+0	484 140
MUL_FR	4	188+2 592+0	220 140
DIV_FR	4	188+2 592+0	220 140
SQRT_FR	48	544+5 1171+0	270 142

5. Podsumowanie

Testowanie generatora IP-core wykazało, że on z pewnością może być wykorzystany przy projektowaniu jednostek przetwarzających RFA dla systemów jednoukładowych przeznaczonych do realizacji w układach FPGA. Wykorzystanie generatora pozwala skrócić czas projektowania w/w jednostek i zwiększyć jego jakość (bezbłędność), jak również zwiększyć jakość projektowanych systemów (zmniejszając złożoność sprzętową lub zwiększając maksymalną częstotliwość pracy jednostek przetwarzających). Obecnie autorzy pracują nad poszerzeniem funkcji generatora IP-core o generowanie opisów bloków wielofunkcyjnych (ALU) oraz o wybór rodziny układu FPGA, dla którego będzie generowany opis VHDL projektowanej jednostki. To pozwoli na dalszą optymalizację projektowanych bloków poprzez uwzględnienie architektur bloków DSP specyficznych dla wybranych rodzin układów (zwłaszcza dla różnej liczby stopni potoku dla projektowanego bloku operacyjnego). W planach dalszej pracy nad generatorem jest również rozszerzenie zestawu dostępnych operacji arytmetycznych, np. o operację mnożenia z akumulacją wyniku.

Praca została wykonana w ramach grantu Ministerstwa Nauki i Szkolnictwa Wyższego N515 002 32/0176.

6. Literatura

[1] C. Berthet. Going Mobile: The Next Horizon for Multi-million Gate Designs in the Semi-Conductor Industry. Proc. IEEE Conf. DAC'2002, pp. 375-378.
[2] M. Keating, P. Bricaud. Reuse Methology Manual For System-on-a-Chip Design. Kluwer Academic Publishers, 1999
[3] C. Fields. Design reuse strategy for FPGAs. Xcell jornal, Xilinx, 2000, pp. 40-42.
[4] L. Rizzatti. How to achieve design productivity increases using architectural synthesis. EDAVision Magazine - January 2002.
[5] K.D. Underwood, K. S. Hemmert. Closing the Gap: CPU and FPGA Trends in sustained Floating Point BLAS Performance. Proc. IEEE Symp. Field Programmable Custom Computing Machines, FCCM 2004.
[6] B. K. P. Horn. Rational Arithmetic for Minicomputers. Software – Practice and Experience, Vol. 8, 1978, pp. 171-176.
[7] P. Kornerup, D. W. Matula. Finite-precision rational arithmetic: an arithmetic unit. IEEE Transactions on Computers, C-32, 1983, pp. 378-388.
[8] O. Masennikow, N. Maslennikow, P. Pawłowski, W. Khadzhynov, A. Sergiyenko. Realizacja w układach FPGA jednostek operacyjnych działających w arytmetyce ułamkowej. Materiały VI Krajowa Konferencja Elektroniki, Darłówko Wschodnie, 2007.
[9] O. Maslennikow, P. Ratuszniak, A. Sergiyenko. Implementation of Cholesky LLT-decomposition algorithm in FPGA-based rational fraction parallel prosessor. MIXDES 2007, Ciechocinek, Poland, June 23-27, 2007.

Artykuł recenzowany