Paweł RUSSEK, Kazimierz WIATR
AKADEMIA GÓRNICZO-HUTNICZA W KRAKOWIE, KATEDRA ELEKTRONIKI, ACK „CYFRONET"

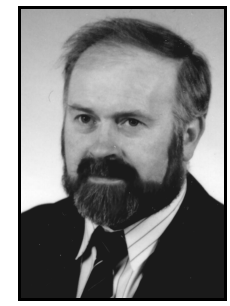
Potokowa realizacja operacji pomnóż i dodaj dla argumentów zmiennoprzecinkowych podwójnej precyzji

Dr inż. Paweł RUSSEK

Ukończył studia na wydziale Elektrotechniki, Automatyki i Elektroniki AGH Kraków (1994), dr nauk technicznych (2003). Jest adiunktem w Katedrze Elektroniki AGH i pracownikiem ACK „Cyfronet". Prowadzone prace badawcze dotyczą sprzętowej akceleracji obliczeń przy pomocy architektur dedykowanych, zagadnień realizacji obliczeń przy użyciu rekonfigurowalnego sprzętu oraz wykorzystania układów reprogramowalnych w obliczeniach naukowych i technicznych wielkiej skali.

e-mail: russek@agh.edu.pl

Prof. dr hab. inż. Kazimierz WIATR



Studia AGH Kraków (1980), dr nauk technicznych (1987), dr habilitowany (1999) i profesor (2002). Profesor zwyczajny na Akademii Górniczo-Hutniczej oraz Dyrektor Akademickiego Centrum Komputerowego Cyfronet AGH. Prowadzone prace badawcze dotyczą komputerowego sterowania procesami, systemów wizyjnych, systemów wieloprocesorowych, układów programowalnych, rekonfigurowalnych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń.

e-mail: wiatr@agh.edu.pl

Streszczenie

Operacja pomnóż i dodaj to fundament realizacji obliczeń numerycznych we współczesnej nauce i technice. Możliwość szybkiej realizacji tej operacji ma zasadnicze znaczenie dla efektywności systemu obliczeniowego. Obok techniki przyśpieszania obliczeń polegającej na równoległej ich realizacji duże znaczenie i zastosowanie ma również technika przetwarzania potokowego. Zwiększa ona przepustowość modułów obliczeniowych wydłużając opóźnienie. W przypadku operatora pomnóż i dodaj zastosowanie techniki potokowej ze względu na pętle sprzężenia zwrotnego w ścieżce danych napotyka pewne problemy. W pracy zaprezentowano sposób potokowej realizacji operacji pomnóż i dodaj oraz wyniki jej implementacji w FPGA dla argumentów zmiennoprzecinkowych podwójnej precyzji.

Słowa kluczowe: układy FPGA, obliczenia dużej złożoności, architektury dedykowane.

Pipeline implementation of multiply and accumulate double precision floating point operation

Abstract

Multiply and accumulate operation is a foundation of contemporary numerical computation in science and technology. Ability for its fast execution is crucial for performance of computing system. In computing acceleration beside parallel processing technique also pipelining has an important role as a way to increase system throughput. In a case of multiply-and-accumulate (MAC) operation there is a problematic issue that comes from the feedback loop necessary in MAC architecture. In this paper double precision MAC pipeline architecture is proposed and FPGA implementation results presented.

Keywords: FPGA, supercomputing, custom computing machines.

1. Wprowadzenie

Celem realizowanych przez autorów prac jest przyśpieszenie obliczeń naukowo-technicznych realizowanych w Akademickim Centrum Komputerowym „Cyfronet" AGH. Przyśpieszaniem objęte będą obliczenia realizowane przez pakiet „Gaussian", który jest wykorzystywany do numerycznej symulacji właściwości cząsteczek chemicznych. Złożoność prowadzonych symulacji wymusza użycie maksymalnych technicznie dostępnych mocy obliczeniowych. Superkomputery dostępne w „Cyfronet" oferują wydajność rzędu 1 TFLOP teoretycznej mocy obliczeniowej. Pomimo to, moce te są przez użytkowników centrum obliczeniowego szybko zużywane. Wynika to przede wszystkim z faktu, że w praktyce złożoność obliczeniowa podejmowanych problemów jest dostosowywana do aktualnie dostępnych mocy komputerów tak, aby czasy uzyskiwanych wyników były racjonalne.

Fakty te skłoniły autorów artykułu do podjęcia prac nad rozwojem alternatywnych technik obliczeniowych opartych o technologię układów rekonfigurowalnych, których wykorzystanie w dziedzinie obliczeń wielkiej skali praktycznie nie było do tej pory podejmowane. Realizowana akceleracja polega na opracowaniu dedykowanej architektury sprzętowej: specjalizowanego koprocesora sprzętowego o architekturze przygotowanej do wydajnego realizowania operacji arytmetycznych specyficznych dla algorytmów realizowanych przez „Gaussian". Platformą do realizacji tej architektury są układy reprogramowalne FPGA. Dzięki ich zastosowaniu możliwe jest zredukowania kosztu NRE, który tradycyjnie towarzyszy wszelkim opracowaniom sprzętowym.

W przypadku obliczeń realizowanych przez pakiet „Gaussian" realizowane algorytmy w dużej części sprowadzają się do rozwiązywania problemów z zakresu algebry macierzy. Są to typowe problemy obliczeniowe, wspólne dla wielu zagadnień naukowo technicznych, takie jak: rozwiązywanie układów równań, poszukiwanie wektorów i wartości własnych macierzy. Wydajne algorytmy obliczeniowe do rozwiązywania tych zadań są w obszarze zainteresowań informatyki praktycznie od czasu pojawienia się pierwszych elektronicznych maszyn liczących. W chwili obecnej obowiązującym standardem jest dekompozycja problemów numerycznych na problemy podstawowe i uzyskiwanie akceleracji metodą optymalizacji czasu realizacji elementarnych procedur. W dziedzinie algebry macierzy elementarną operacją jest mnożenie macierzy, a obowiązującym standardem biblioteka BLAS (ang. Basic Linear Algebra Subroutines)[4]. Praktycznie każda architektura komputerowa posiada optymalizowaną pod względem wydajności bibliotekę procedur BLAS. Wykorzystanie szybkich procedur mnożenia macierzy dostarczanych przez BLAS przez algorytmy wyższego poziomu zapewnia wydajne czasowo obliczenia. W związku z powyższym zasadne wydaje się podjęcie próby akceleracji obliczeń „Gaussian" poprzez akcelerację procedur BLAS.

Należy także podkreślić, że duża dynamika danych wartości pojawiających się w obliczeniach kwantowo-chemicznych wymaga ich realizowania przy wykorzystaniu reprezentacji zmiennoprzecinkowej podwójnej precyzji (64 bit).

Podstawowym operatorem mnożenia macierzy, które ma być implementowane w układach rekonfigurowalnych jest operacja pomnóż i dodaj: MAC (ang. multiply-and-accumulate). Jej realizacja jest podstawowym warunkiem późniejszej realizacji algorytmu mnożenia macierzy. W celu uzyskania maksymalnej przepustowości projektowanej architektury operatory MAC muszą pracować w trybie potokowym, z wydajnością jednej operacji MAC na jeden takt zegara. Realizacja takiego założenie napotkała na problemy, których możliwe rozwiązanie prezentuje artykuł.

2. Prace pokrewne

Alternatywnym podejściem jest podniesienie mocy obliczeniowej centrum obliczeniowego poprzez zwiększanie liczby węzłów obliczeniowych z procesorami ogólnego zastosowania. Jednak rozwiązania te wymagają poniesienia ogromnych kosztów

i w praktyce stosowane jest jedynie przez bardzo nieliczne ośrodki. Realizacja wspomnianego koprocesora w technologii FPGA pozwoli na podniesienie mocy obliczeniowej systemu przy poniesieniu konkurencyjnych w stosunku do superkomputerów kosztów.

Inne popularne podejście to obliczenia typu GP-GPU (ang. General-Purpose Computation Using Graphics Hardware) [3]. Problematyka realizacji operacji splotu jest powszechna i jest podstawową operacją podczas różnego rodzaju operacji przetwarzania grafiki. Masowość wykorzystania sprzętowych akceleratorów grafiki (np.: procesora Cell [8]) jest gwarantem ich niskiej ceny, a zatem dostępności taniego i szybkiego rozwiązania sprzętowego do zastosowań naukowych. Rozwiązanie takie jest dobre, jeżeli budujemy instalacje nakierowane na obliczenia jednego typu i wystarczające są obliczenia pojedynczej precyzji. W warunkach, w których istotna jest uniwersalność oferowanych rozwiązań i ich elastyczność, FPGA wydają się lepszym wyborem. Inwestycje poniesione na wprowadzenie technologii układów rekonfigurowalnych będą służyć szerszej grupie użytkowników poprzez opracowanie innych odpowiednich dla nich architektur sprzętowych.

Układy FPGA dają ponadto możliwość czerpania korzyści z wielokrotnego ich programowania. Możliwe jest zatem zastosowanie tej samej struktury krzemowej do przyśpieszania kilku następujących po sobie zadań poprzez ciągłą podmianę konfiguracji zgodnie z aktualnie potrzebną strukturą sprzętu.

Podobne prace w dziedzinie układów rekonfigurowalnych były już podejmowane, jednak nie polegały na implementacji w konkretnym środowisku obliczeniowym[1] lub nie obejmowały realizacji operatorów podwójnej precyzji [2, 5].

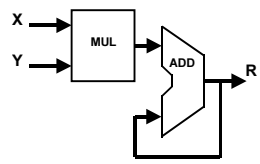
3. Platforma RASC

Kluczowe w architekturze sprzętowej dedykowanej do akceleracji sprzętowej obliczeń jest zapewnienie kanału przesyłu danych pomiędzy pamięcią główną systemu, a pamięcią lokalną koprocesora sprzętowego. W wielu rozwiązaniach akcelerator sprzętowy dla systemu komputerowego dołączany jest jako karta rozszerzeń wejścia-wyjścia. W praktyce okazuje się, że takie rozwiązania rzadko dają dobre rezultaty. W przypadku, kiedy mamy do czynienia z systemem wyposażonym w procesor o bardzo szybkiej magistrali systemowej, okazuje się, że realizacja obliczeń w układach FPGA przestaje być atrakcyjna, jeżeli do czasu obliczeń dodamy czas potrzebny na przesłanie danych obliczeniowych z pamięci głównej do pamięci na karcie z układem FPGA. Między innymi dlatego w technologii FPGA sięgnięto do rozwiązania, w którym procesor znajduje się na jednym półprzewodniku z logiką rekonfigurowaną tak, aby wymiana danych pomiędzy procesorem a koprocesorem FPGA nie konsumowała czasu, czyniąc koprocesor nieprzydatnym. Innym podejściem, stosowanym w systemach wielkich mocy obliczeniowych, jest mocne zintegrowanie koprocesora rekonfigurowalnego z magistralą systemową procesora (np.: Cray XD1). Takie rozwiązanie zaproponowała firma SGI w swoich systemach Altix [6]. System Altix to system typu SMP, w którym poszczególne procesory dysponując swoją pamięcią lokalną w węźle mają możliwość szybkiego dostępu do pamięci pozostałych procesorów za pośrednictwem wydajnego łącza NUMALink. NUMALink pozwala na transfer danych o przepustowości 6,4GB/s. Za pośrednictwem tego łącza do systemu Altix może być dołączony moduł z logiką rekonfigurowalną: RASC (ang. Reconfigurable Application Specific Computing).

Moduł RASC wyposażony jest w dwa układy FPGA serii Virtex4 oferujące 200 tysięcy makro-komórek logicznych oraz 64kb dwuportowej pamięci RAM stanowią wystarczającą ilość zasobów do tego, aby zaimplementować funkcje i operatory zmiennoprzecinkowe podwójnej precyzji. Ponadto na karcie RASC znajduje się 128MB pamięci QDR RAM stanowiącej dla modułów FPGA pamięć drugiego poziomu (pamięć pierwszego poziomu to wspomniana wcześniej pamięć w strukturze FPGA). Wymiana danych pomiędzy FPGA i pamięcią QDR odbywa się poprzez dwa kanały 128-bitowe pracujące z częstotliwością 200MHz.

4. Architektura potokowa MAC

Klasyczną architekturę sprzętową typu MAC przedstawia rysunek 1.



Rys. 1. Ogólny schemat architektury MAC
Fig. 1. Overall MAC architecture concept diagram

Układ składa się z jednego układu mnożącego i jedne układu dodającego. Sprawdza się on przy ciągłym, strumieniowym dostępie do danych wejściowych, kiedy w każdym takcie zegara na wejściu pojawia się nowa para współczynników. Wtedy równocześnie realizowana jest operacja mnożenia i dodawania. Architektura taka działa poprawnie przy założeniu, że operacja dodawania wykonywana jest w jednym takcie zegara. Kiedy jednak opóźnienie wyjścia względem wejścia wynosi więcej niż jeden takt zegara sytuacja komplikuje się z dwóch powodów:

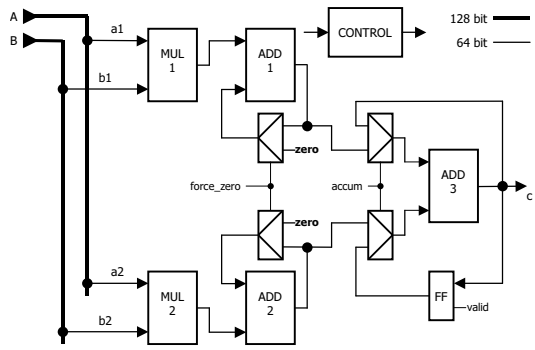
- Należy zapewnić właściwy start układu. Na początku przez liczbę taktów zegara równą opóźnieniu sumatora w pętli sprzężenia zwrotnego powinna pojawiać się wartość zero.
- Po zakończeniu wprowadzania mnożonych wektorów wejściowych należy zapewnić właściwe zsumowanie wyników pośrednich zgromadzonych w wewnętrznym potoku sumatora.

Ponieważ nasz projekt wymaga maksymalnej dostępnej przepustowości, potokowy charakter zastosowanych układów mnożących i dodających jest oczywisty. Dodatkowo przy ciągłym przetwarzaniu napływających danych wejściowych maksymalizowanie przepustowości kosztem opóźnienia układu wydaje się racjonalne. Dodatkowym założeniem projektowym jest, aby dla zapewnienia lepszej wydajności kolejne wektory były mnożone przez układ jeden po drugim, bez żadnych przerw w ich wprowadzaniu.

Jak już wspomniano, RASC zapewnia 128-bitowy interfejs pamięci zewnętrznej. Jest tak, ponieważ częstotliwość pracy NUMALink przy transferze danych z pamięci operacyjnej systemu jest większy niż możliwa do obsłużenia przez układ FPGA. Aby zapewnić pełną obsługę strumienia danych, FPGA operuje magistralą 128-bitową przy dwukrotnie obniżonej częstotliwości taktowania w stosunku do NUMALink. Ponieważ projektowany układ MAC ma operować na danych 64 bitowych, współpraca z dwukrotnie szerszą magistralą danych wymaga dodatkowego dostosowania.

Zrealizowaną architekturę spełniającą postawione wymagania przedstawia rysunek 2.

Ponieważ w każdym takcie zegarowym na wejściu układu MAC pojawiają się dwie pary współczynników wejściowych, które równocześnie muszą być pomnożone, układ zawiera dwa układy mnożące MUL. Każdy układ mnożący posiada odpowiadający mu układ sumatora, w którym sumowane są wyniki mnożenia. W MUL1 mnożone są współczynniki nieparzyste, a w MUL2 parzyste. Następnie ADD1 sumuje iloczyny parzyste, a ADD2 nieparzyste. W pętli sprzężenia zwrotnego ADD1 i ADD2 pojawia się układ multipleksera przy pomocy którego można wymusić wartość zero na wejściu odpowiedniego sumatora. Zapewnia on prawidłowy start procesu sumowania tzn. wprowadza zero na wejście sumatora przez liczbę taktów zegara równą opóźnieniu sumatora, do momentu zapełnienia potoku sumatora danymi ważnymi dla aktualnie mnożonych wektorów.



Rys. 2. Schemat blokowy potokowej architektury MAC z 128-bitowym interfacem wejściowym
Fig. 2. A block diagram of pipelined architecture of 128-bit interface MAC

Rola sumatora ADD3 jest podwójna. Po pierwsze zapewnia on dodanie do siebie wyników sumowania iloczynów parzystych i nieparzystych. Po drugie przy jego pomocy następuje zsumowanie sum częściowych danych zgromadzonych w wewnętrznych potokach sumatorów ADD1 i ADD2 na zakończenie mnożenia pary wektorów. Ponieważ ADD3 ma identyczną budowę jak pozostałe sumatory (jest to budowa potokowa) nadal pozostaje problem zsumowania danych pozostających w jego potoku wewnętrznym na zakończenie mnożenia. Jednak w przeciwieństwie do ADD1 i ADD2, ADD3 nie pracuje w trybie ciągłym. Na jego wejściu pojawiają się dane tylko w fazie kończącej mnożenie dwóch wektorów. W czasie pomiędzy tymi kończącymi fazami możliwe jest cykliczne dodanie do siebie danych z potoku wewnętrznego ADD3 zatrzaskując w odpowiedni sposób dane w FF przy pomocy sygnału „valid". Multipleksery na wejściach ADD3 służą właśnie do przełączenia sumatora z trybu sumowania iloczynów parzystych i nie parzystych w tryb sumowania danych z potoku.

5. Wyniki implementacji

Przedstawioną architekturę układu MAC zrealizowaną przy wykorzystaniu operatorów zmiennoprzecinkowych dostarczonych jako moduły IPCore przez firmę Xilinx. Są to moduły pracujące zgodnie ze standardem kodowania danych zmiennoprzecinkowych IEEE-754. Parametry zastosowanych modułów przedstawia tabela 1.

Tab. 1. Parametry zastosowanych sumatorów i mnożarek
Tab. 1. Utilized multipliers' and adders' parameters

	Mnożarka 1	Mnożarka 2	Sumator
Logic Slice	1 335	559	851
DSP 48	0	16	0
Opóźnienie (takty zegara)	21	18	12

Zastosowany układ FPGA Virtex4 [7] oferuje 89 088 elementów logicznych Logic Slice. Elementy te to podstawowe komórki służące implementacji funkcji cyfrowych. Ponadto Virtex oferuje 96 specjalizowanych elementów DSP48 realizujących funkcje MAC typu stałoprzecinkowego. Istnieje możliwość zastosowania tego typu elementów do realizacji układów mnożących zmiennoprzecinkowych, co pozwala na zaoszczędzenie części logiki uniwersalnej. W związku z tym do budowy układów MAC można użyć obu typów mnożarek. Przedstawiona w tabeli 1 „Mnożarka 1" jest implementowana przy pomocy samej tylko logiki uniwersalnej, a „Mnożarka 2" z zastosowaniem bloków specjalizowanych DSP48.

Również układy MAC można zrealizować w dwóch wariantach (z DSP48 i bez). Odpowiednie zapotrzebowanie na zasoby przedstawia tabela 2.

Tab. 2. Zapotrzebowanie na zasoby układów MAC
Tab. 2. The MAC's resource utilization

	Logic	DSP48
MAC	5 480	
MAC (DSP48)	3 908	16

Z przedstawionych parametrów wynika, że w pojedynczy układzie Virtex4LX200 można teoretycznie zrealizować 20 takich układów MAC. Jednak w praktyce eksperyment pokazał, że realizacja takiej liczby układów nie jest możliwa. Ponadto liczba układów MAC równocześnie umieszczonych w strukturze reprogramowalnej ma bezpośredni wpływ na możliwą do osiągnięcia częstotliwość pracy układu. Przy implementacji równocześnie 10 układów MAC, co oznacza 50% zużycie dostępnych zasobów logicznych można uzyskać częstotliwość pracy zegara równą 50MHz. Należy podkreślić, że teoretyczna częstotliwość pracy dostarczonych przez Xilinx IPCores to ponad 250MHz.

6. Podsumowanie

Zrealizowana architektura, oraz otrzymane wyniki implementacji wielu takich układów pokazały, że przy mnożeniu zmiennoprzecinkowym podwójnej precyzji przy pomocy układu reprogramowalnego możliwe jest uzyskanie 2 GFLOP mocy obliczeniowej. Przede wszystkim jest to moc obliczeniowa otrzymana dla zegara o bardzo małej w porównaniu z procesorami ogólnego stosowania częstotliwości zegara. Oznacza to oczywiście energetyczną atrakcyjność rozwiązania. Co prawda w przypadku obliczeń zmiennoprzecinkowych podwójnej precyzji FPGA nie dystansuje innych rozwiązań tak jak to ma miejsce w zastosowaniach o stałym przecinku, to jednak wynik jest atrakcyjny, a rozwiązanie warte rozważenia zwłaszcza w przypadku kiedy system jest wyposażony w układ reprogramowalny wykorzystywany równocześnie do innego typu obliczeń. Z drugiej strony dla obliczeń zmiennoprzecinkowych pojedynczej precyzji można oczekiwać dwukrotnie lepszych parametrów.

7. Literatura

[1] Yong Dou, S. Vassiliadis, G. K. Kuzmanov, G. N. Gaydadjiev, "64-bit floating-point FPGA matrix multiplication", International Symposium on Field Programmable Gate Arrays archive Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays
[2] Ling Zhuo, Viktor K. Prasanna, "High Performance Linear Algebra Operations on Reconfigurable Systems" Proceedings of the 2005 ACM/IEEE conference on Supercomputing SC '05
[3] K. Fatahalian, J. Sugerman, P. Hanrahan, "Computation: Understanding the efficiency of GPU algorithms for matrix-matrix multiplication" August 2004 Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware HWWS '04
[4] J. J. Dongarra, Jeremy Du Croz, Sven Hammarling, I. S. Duff , "A set of level 3 basic linear algebra subprograms" March 1990, ACM Transactions on Mathematical Software (TOMS), Volume 16 Issue 1, Publisher: ACM Press
[5] Viktor K. Prasanna, "Energy-Efficient Computations on FPGAs", May 2005, The Journal of Supercomputing, Volume 32 Issue 2, Publisher: Kluwer Academic Publishers
[6] Silicon Graphics, "SGI® RASC™ RC100 Blade, Dramatic Application Speed-up with Next Generation Reconfigurable Compute Technology", www.sgi.com
[7] Xilinx, "Virtex-4 User Guide", www.xilinx.com
[8] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, D. Shippy, "Introduction to the Cell multiprocessor", IBM Journal of research and development, Published online September 7, 2005

Artykuł recenzowany