

Paweł Russek*, Kazimierz Wiatr*

Perspektywy przyśpieszenia obliczeń w instalacjach o wielkich mocach obliczeniowych za pomocą układów logiki rekonfigurowalnej

1. Wprowadzenie

Pojęcie obliczeń typu RC (*Reconfigurable Computing*) upowszechniło się w środowisku inżynierów i naukowców zainteresowanych rozwojem i wykorzystywaniem układów rekonfigurowalnych FPGA w latach 90. W tym okresie wprowadzono pierwsze pojęcia związane z tym sposobem prowadzenia obliczeń oraz pojawiły się pierwsze aplikacje z nim związane. Był to okres, w którym pojemność dostępnych układów rekonfigurowalnych wynosiła kilkadziesiąt tysięcy bramek logicznych. Jednak mimo to układy te z powodzeniem spełniały zadania obliczeniowe i wymagania stawiane przez entuzjastów obliczeń wspieranych przez rekonfigurowalną logikę.

Mylne jest przypuszczenie, że każde urządzenie wykorzystujące układ rekonfigurowalny jako część swojej cyfrowej struktury może być klasyfikowane jako system z rodziny RC. Nie było bowiem intencją twórców pomysłu na rekonfigurowalne systemy obliczeniowe ograniczenie się jedynie do prostego zastępowania w zastosowaniach aplikacyjnych układów typu MPGA (*Mask Programmable Gate Arrays*) układami FPGA (*Field Programmable Gate Arrays*). Zamiana taka sprowadzałaby się do zastąpienia układów *semi-custom* ASIC programowalnych w procesie technologicznym za pomocą zaprojektowanej dla indywidualnych potrzeb odbiorcy maski, układami programowanymi za pomocą rekonfigurowalnego elementu logicznego (*reconfigurable element*). Podstawową zaletą takiej zamiany jest oczywiście obniżenie kosztów stosowanego układu, dzięki jego pełnej uniwersalności i temu, że programowanie realizowane jest samodzielnie przez użytkownika. Choć powyższy proces zastępowania innych technologii cyfrowych przez technologię FPGA w ostatnim okresie przybiera na sile, zwłaszcza dzięki pojemnościom oferowanym przez współczesne układy rekonfigurowalne (2005 r.: Xilinx XC4VLX200 FPGA – 20 milionów równoważnych bramek cyfrowych) oraz znacznemu poprawieniu się ich parametrów czasowych przy

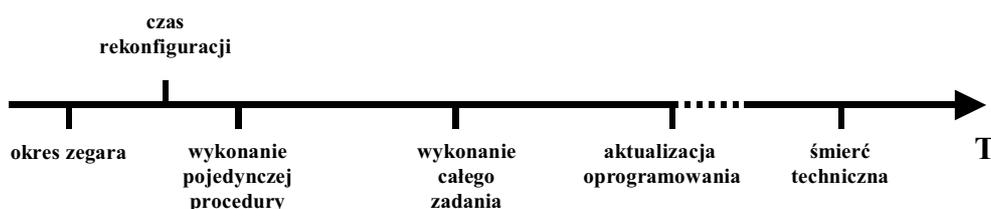
* Katedra Elektroniki, Akademia Górniczo-Hutnicza, ACK „Cyfronet”, Kraków;
russek@agh.edu.pl; wiatr@agh.edu.pl

ciągłe konkurencyjnej cenie, to pojęcie RC wymaga i zakłada pełniejsze wykorzystanie unikalnej właściwości logiki programowalnej: rekonfigurowalności.

2. Systemy rekonfigurowalne

Pojęcie RC prowadzi do powstawania systemów rekonfigurowalnych. Praca takiego systemu opiera się na założeniu, że rekonfigurowalne zasoby cyfrowe wchodzące w jego skład zmieniają swoją konfigurację w zależności od środowiska pracy systemu, aktualnie wykonywanego zadania, wybranego trybu pracy czy etapu realizowanych obliczeń. Jednym z pierwszych rekonfigurowalnych systemów obliczeniowych przedstawionych w literaturze był układ koderów wideo opisany w [9].

Systemy rekonfigurowalne można klasyfikować według względnej częstości rekonfiguracji zasobów programowalnych (rys. 1). Rysunek przedstawia możliwe odstępstwa czasu pomiędzy kolejnymi zmianami struktury połączeń układu FPGA w odniesieniu do czasu trwania zjawisk związanych z pracą układu.



Rys. 1. Skala czasu zdarzeń systemu cyfrowego

Częstość reprogramowania świadczy o efektywności wykorzystania zdolności rekonfigurowalności logiki. Oczywiście rekonfigurowanie sieci połączeń FPGA w celu uaktualnienia systemu czy usunięcia błędów pokazuje możliwości logiki programowalnej podobne do właściwości software'u, jednak nie mają struktury sprzętowej zbudowanej w oparciu o układy ASIC, to jednak celem RC jest zmierzanie do częstego rekonfigurowania sprzętu [7]. Najbardziej wymiernym rezultatem takiego postępowania jest oszczędność zasobów obliczeniowych, bowiem te same bramki logiczne są wielokrotnie wykorzystywane do różnych zadań realizowanych przez sprzęt i nie ma potrzeby budowania wielu bloków urządzenia które przez większość czasu pracy systemu nie są wykorzystane. Można na przykład podmieniać konfiguracje adekwatnie do aktualnie realizowanej procedury będącej składową wykonywanego algorytmu [11, 12].

2.1. Rekonfiguracja w locie, częściowa i relokacja

Rekonfiguracja w locie RTR (*Run-Time Reconfiguration*) polega na zmianie konfiguracji układu FPGA w czasie jego pracy. Założenie jest takie, że przeprowadzanie procesu

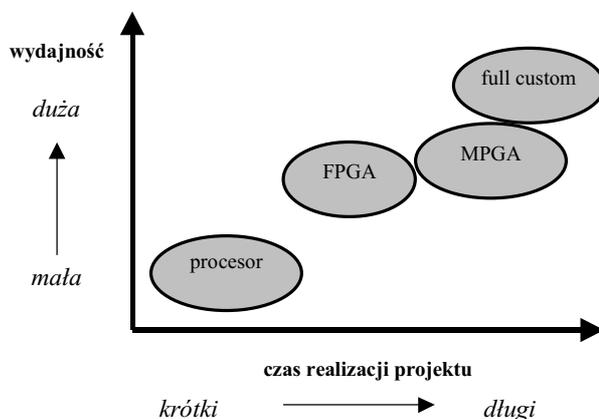
rekonfiguracji, które z natury rzeczy jest czasochłonne (duża ilość koniecznych do zaprogramowania bitów konfiguracji), nie powinno zakłócać normalnej pracy układu. W praktyce oznacza to istnienie dla każdej komórki konfiguracji tzw. rejestrów cieni (*shadow registers*), czyli komórki pamięci, która bezpośrednio decyduje o konfiguracji układu, ale sama jest zapisywana w właściwej pamięci konfiguracji, dopiero podczas włączenia zasilania albo na komendę. Programowanie polega na umieszczeniu nowej konfiguracji w pamięci konfiguracji, a następnie – aby zmiany przyniosły efekt – przepisaniu danych do rejestrów cieni.

Rekonfiguracja częściowa polega na możliwości zmiany połączeń tylko w części struktury. Funkcja ta nie jest wcale oczywista dla większości układów FPGA, ponieważ w przeważającej części komórki pamięci konfiguracji są połączone szeregowo i dostęp do nich jest sekwencyjny [2, 3].

Możliwością, która jak do tej pory sprawia najwięcej kłopotów, jest relokacja. Chodzi w niej o to, aby dany moduł sprzętowy mógł zajmować w strukturze FPGA dowolną pozycję. Funkcja ta sprowadza się do możliwości manipulowania strumieniem konfiguracyjnym tak, aby było możliwe załadowanie modułu sprzętowego pod dowolny adres wewnątrz tablicy FPGA, bez konieczności rekompilacji kodu. Próby budowy systemów tego typu były już z powodzeniem podejmowane [5]. Jednak kłopot z tą funkcją polega na tym, że jej realizacja jest silnie związana z platformą sprzętową, co sugerowałoby producenta technologii FPGA jako dostawcę stosownych rozwiązań. Jednak, jak do tej pory, autorzy nie spotkali się z komercyjną ofertą tego typu. Z drugiej strony relokacja modułów ma sens tylko w wypadku całkowicie homogenicznej struktury zasobów rekonfigurowalnych, co jest oczywiście z założenia cechą układów FPGA. Niestety, z tego punktu widzenia, ostatnio w poszukiwaniu polepszenia parametrów czasowych swoich układów programowalnych i konkurencyjności z technologią *semi-custom* i *full-custom* ASIC ich producenci coraz bardziej odchodzą od tradycyjnej jednorodnej struktury dokładając specjalizowane bloki o ustalonej strukturze.

3. Sprzętowe wspomaganie obliczeń

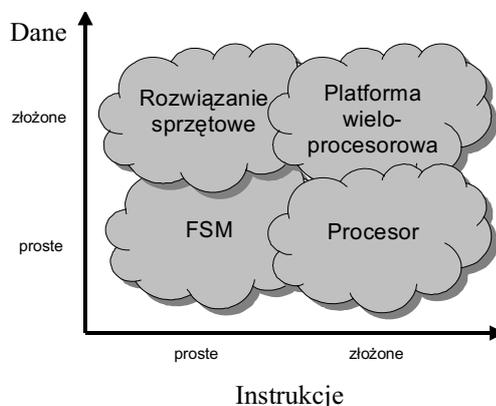
Pomysł na systemy rekonfigurowalne mocno wiąże się z pojęciem obliczeń realizowanych sprzętowo. W praktyce projektowej bardzo często zadania realizowane przez procesor ogólnego przeznaczenia lub nawet procesor DSP są, w przypadku krytycznych wymagań czasowych, przenoszone do realizacji sprzętowej. Realizacja za pomocą dedykowanego hardware, czerpiąc z możliwości potokowości i równoległości wykonywania zadania daje możliwość przyspieszenia wykonania algorytmu. Do czasu pojawienia się układów FPGA układy dedykowane (procesory dedykowane) powstawały na platformach ASIC i z tego powodu, ze względu na całociowy koszt powstania i czas realizacji projektu, ograniczały się do zastosowań typowych, np.: mnożenia macierzy, estymacji ruchu, dekompozycji częstotliwościowych itd.



Rys. 2. Zależność wydajności obliczeniowej od czasu/kosztu realizacji zadania dla różnych platform obliczeniowych

Rysunek 2 obrazuje, w jaki sposób konieczność podniesienia wydajności systemu wpływa na koszt/czas rozwiązania problemu. Układy FPGA oferują możliwość realizacji rozwiązania hardwarowego i znaczącego polepszenia parametru wydajnościowego rozwiązania w porównaniu z rozwiązaniem opartym na procesorze (rozwiązanie softwarowe), przy jednoczesnym braku konieczności ponoszenia kosztów fabrykacji specjalizowanego układu. Dodatkowy koszt wynika z większego wysiłku projektowego polegającym na opracowywaniu szczegółowego rozwiązania niskiego poziomu uogólnienia. W ostatnim okresie koszt opracowania rozwiązania hardwarowego znacząco się obniżył, dzięki zastosowaniu nowoczesnych narzędzi projektowania i syntezy układów cyfrowych, użyciu do projektowania systemów języków opisu sprzętu HDL (VHDL, Verilog), języków HLL (*High Level Languages*) takich jak: *System-C*, *Handel-C* oraz możliwości bezpośredniego implementowania sprzętu ze środowisk naukowo-inżynierskich typu MATHLAB i Simulink.

Charakterystyczną cechą każdego algorytmu jest jego podatność na realizację sprzętową oraz złożoność obliczeniowa. Pewne operacje obliczeniowe łatwo dają się realizować w sprzęcie przy niewielkiej jego złożoności, a z kolei inne nie pochłaniają aż tak dużo czasu procesora ogólnego stosowania, aby warto było się zajmować ich implementacją sprzętową. Wyważenie pomiędzy tymi dwoma czynnikami decyduje o wybranej do realizacji algorytmu platformie obliczeniowej. Tradycyjny podział metodologii rozwiązań problemu przedstawia rysunek 3. Mówimy o algorytmach zorientowanych na dane i algorytmach zorientowanych na instrukcje. Najefektywniej sprzętowo są realizowane algorytmy, w których istnieje przewaga danych nad instrukcjami, to znaczy wykonywane są te same proste operacje arytmetyczne na ogromnej liczbie danych. Typowymi przykładami operacji tego typu są na przykład operacje typu DSP, np.: filtracje, konwolucje.

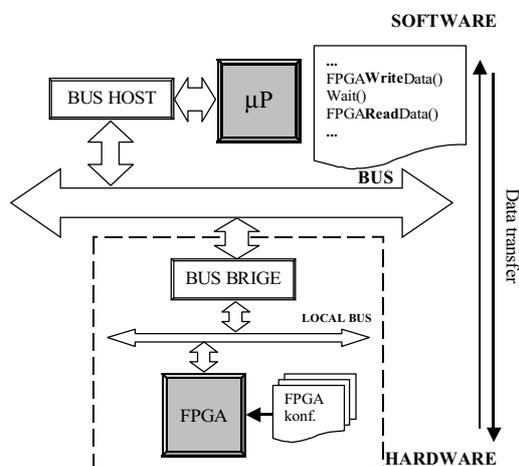


Rys. 3. Algorytmy zorientowane na dane/instrukcje i ich platformy realizacji

3.1 Systemy hybrydowe

We praktyce inżynierskiej bardzo rzadko można spotkać użyteczny praktycznie algorytm, który w całości przystaje do schematu algorytmu zdominowanego przez dane. Mamy raczej do czynienia z systemami rozwiązywania problemów, składających się z wielu modułów, których tylko część odpowiada implementacji sprzętowej. Przykładem mogą być tu na przykład współczesne algorytmy kompresji dźwięku czy obrazu. Wraz ze wzrostem wymagań stały się one skomplikowanymi procedurami z rozgałęzionym systemem decyzyjnym, przy jednoczesnym zachowaniu tradycyjnych dobrze realizowanych fragmentów algorytmów kompresji takich, jak dekompozycja częstotliwościowa czy estymacja ruchu. Cechą wielu algorytmów jest równocześnie zapotrzebowanie na duże bloki pamięci w których przechowywane są wyniki pośrednie operacji.

Powyższe spostrzeżenia prowadzą do metody rozwiązywania problemów opartej o tandem software-hardware (rys. 4). W rozwiązaniu tym system obliczeniowy wyposażony jest w koprocessor rekonfigurowalny, który wspomaga pracę procesora ogólnego stosowania. Rozwiązanie z koprocessorem jest od dawna znane i sprawdzone w technice komputerowej. Innowacja polega na uniwersalności zaproponowanego koprocessora. Projektowanie systemu obliczeniowego opartego o takie rozwiązanie może polegać na przekształcaniu istniejących kodów programu w taki sposób, aby najbardziej wymagające fragmenty kodu zastępować odwołaniami (transferem danych) do sprzętu. Jest to technika również znana jako Hardware-Software Co-Design (HW-SW Co-Design). Różnica w zaproponowanym systemie polega na tym, że zanim odpowiednie wywołanie sprzętu z programu głównego zostanie wykonane, najpierw układ koprocessora należy odpowiednio skonfigurować. Oczywiście tym sposobem za pomocą jednego układu FPGA można przyspieszać różne fragmenty kodu. Oprogramowanie umożliwiające stosowne przerobienie rozwiązań SW na rozwiązanie HW-SW jest dostępne komercyjnie [6].



Rys. 4. System hybrydowy z koprocesorem rekonfigurowalnym

W ostatnim okresie daje się zauważyć rosnące zainteresowanie firm produkujących systemy obliczeniowe alternatywnymi do prostego skalowania procesorów metodami prowadzenia obliczeń, mającymi na celu przyspieszenie pracy komputerów. Jest to niewątpliwie wynik załamania się prawa Moore'a i bliskiego kresu możliwości zmniejszania tranzystora półprzewodnikowego. Wśród rozwiązań dających szansę na jakościową rewolucję w metodach prowadzenia obliczeń są niewątpliwie układy rekonfigurowalne. Już teraz pojawiają się opracowania wychodzące z laboratoriów badawczo-rozwojowych firm oferujących kompleksowe rozwiązania przetwarzania danych rozważające perspektywy zastosowania układów FPGA w oferowanym sprzęcie [8]

Oczywiście możliwości rozwoju i zastosowania logiki rekonfigurowalnej w praktycznie wykorzystywanych do obliczeń systemach jest ściśle powiązane z rozwojem odpowiedniego oprogramowania, które będzie korzystało z dostępnych akceleratorów rekonfigurowalnych. Trzeba zaznaczyć, że niewątpliwie bardziej kłopotliwe i pracochłonne jest opracowanie rozwiązań typu HW-SW, w przeciwieństwie do zautomatyzowanego i ułatwionego dzięki językom wysokiego poziomu projektowanie software'u. Propozycją na przyszłość są niewątpliwie języki HLL, które umożliwiają opisywanie zarówno oprogramowania, jak i sprzętu za pomocą tego samego języka. W chwili obecnej jest to język C, dla którego istnieją sprzętowe odpowiedniki takie, jak System-C i Handel-C. Jest to niewątpliwie krok w dobrym kierunku, choć obecnie nie wszystkie konstrukcje programowe są możliwe do zaimplementowania w sprzęcie. Z natury rzeczy brak jest sprzętowego odpowiednika rekurencji i alokacji pamięci. Aktualnie brak także obsługi typów zmiennych typu *float*.

4. Obiekty sprzętowe w środowisku wielozadaniowym

Ciekawym przypadkiem pracy systemu hybrydowego RHW-SW (*Reconfigurable Hardware-Software*) jest przypadek, w którym oprogramowanie wykonywane jest pod nad-

zorem systemu operacyjnego wielozadaniowego lub nawet wieloprocesorowego. W systemie wieloprocesorowym wygodne byłoby przyjęcie architektury, w której każdy procesor wyposażony jest w niezależny rekonfigurowalny koprocesor. W taki sposób znika problem współdzielenia zasobów rekonfigurowalnych, a tym samym zarządzanie i przydzielanie tych zasobów.

Ciekawym problemem do rozwiązania jest współdzielenie logiki rekonfigurowalnej w ramach jednego procesora przez wątki wykonywane na tym procesorze. Do rozwiązania tego przypadku wprowadzono pojęcie obiektu sprzętowego (*hardware object*) [4].

4.1. Obiekty sprzętowe

Pojęcie obiektu sprzętowego umożliwia zrealizowanie współdzielenia zasobów rekonfigurowalnych przez procesy programowe pracujące w systemie wielowątkowym. Przez obiekt sprzętowy rozumiemy wydzielony geometrycznie obszar struktury reprogramowalnej zaprogramowany tak, że jest integralny pod względem funkcjonalności. Obiekt sprzętowy jest własnością wątku, który wykorzystuje jego strukturę hardware'ową do realizacji swojego zadania. Pojęcie bloków sprzętowych w swej istocie podobne jest do bloków pamięci alokowanych przez wątki dla swoich celów. Podobieństwo polega na tym, że zasoby reprogramowalne stają się w systemie wielowątkowym dobrem dzielonym, z którego korzystać można dopiero po dokonaniu odpowiedniej alokacji zasobów. Z tego powodu w systemie wspierającym RC oprócz tzw. menedżera pamięci MMU (*Memory Management Unit*) znajdować się powinien menedżer zasobów rekonfigurowanych LMU (*reconfigurable Logic Management Unit*).

W przeciwieństwie do zarządzania pamięcią, praca LMU rodzi pewne problemy.

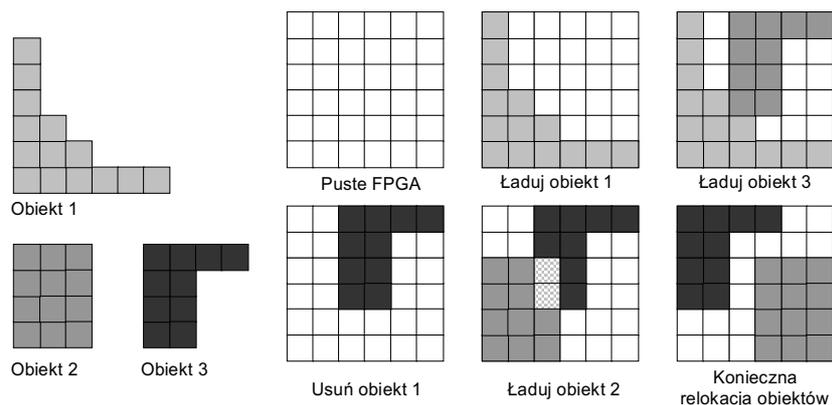
Zakładamy, że logika użyta do realizacji wspomaganego sprzętowego systemu cechuje się trzema cechami:

- 1) rekonfiguracją w locie,
- 2) częściową rekonfiguracją,
- 2) relokacją modułów.

Jest to z oczywistych powodów niezbędne założenie. Problemy, jakie może napotkać LMU, przedstawia rysunek 5. Trzeba tu zauważyć, że ze względu na dwuwymiarową strukturę FPGA problem relokacji zasobów logicznych dodatkowo się komplikuje. Najprostszym rozwiązaniem tego problemu jest założenie, że obiekty sprzętowe zajmują zawsze prostokątny obszar, najlepiej o stałej długości jednego z boków.

Kolejnym zagadnieniem utrudniającym współdzielenie struktury reprogramowalnej w ramach systemów wielowątkowych jest problem dostępu do obiektu z zewnątrz struktury FPGA w celu wymiany danych. Dodatkowo, oprócz konieczności przesłania argumentów dla operacji sprzętowej bardzo często zachodzi potrzeba użycia wewnętrznego magazynu danych: pamięci wyników pośrednich, przez obiekt w czasie realizacji operacji obliczeniowych. Co prawda nowoczesne układy FPGA oferują pamięci RAM wewnątrz swojej struktury, to jednak bloki te są umieszczone w niektórych obszarach FPGA i mogą nie być do-

stępne z dowolnego miejsca struktury. Dodatkowo możliwość wymiany informacji lub danych pomiędzy poszczególnymi obiektami może powodować problemy przy rekonfiguracji. Trzeba bowiem przewidzieć zachowanie dwóch modułów współpracujących ze sobą, z których jeden jest w danej chwili rekonfigurowany.



Rys. 5. Zarządzanie obiektami sprzętowymi w tablicowej strukturze reprogramowalnej

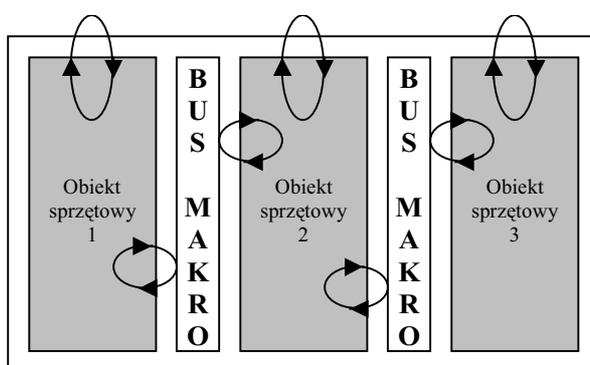
Należy również zauważyć, że oferowane współcześnie przez firmy (Altera Xilinx) układy FPGA o największych pojemnościach coraz mniej przypominają klasyczną strukturę FPGA bazującą na założeniu w pełni symetrycznej i równomiernej siatki identycznych zasobów logicznych. Taka siatka najlepiej pasuje do koncepcji obiektów sprzętowych, które mogą być lokowane gdziekolwiek wewnątrz struktury. W zamian za lepsze parametry czasowe przy rozwiązywaniu pewnych typowych problemów współczesne FPGA oprócz struktury reprogramowalnej zawierają w swoich strukturach bloki pamięci, mnożarki czy procesory. Psuje to pierwotną formę FPGA i utrudnia opracowanie rekonfigurowalnego, współdzielonego koprocatora sprzętowego.

5. Xilinx Virtex FPGA

Oferowane przez firmę Xilinx układy FPGA oraz narzędzia projektowe dostępne komercyjnie są w sposób ciągły rozwijane zarówno w postaci coraz to większych pojemności układów, coraz lepszego dopasowania architektury do najczęstszych zadań projektowych stosowanych w praktyce, jak i w kierunku sprostania wyzwaniom, które w przyszłości niewątpliwie staną się udziałem FPGA, to znaczy opisywanej tu sprzętowej akceleracji algorytmów i systemów HW-SW [10].

Układy Virtex oferują możliwość zarówno RTR, jak i rekonfiguracji częściowej. Strumień konfiguracyjny jest podzielony na ramki, z których każda odpowiada za konfigurację pojedynczej, całej kolumny w ramach tablicy struktur reprogramowalnej. Minimalnym obszarem rekonfiguracji jest właśnie taka jedna ramka. Rekonfiguracja częściowa może odby-

wać się zatem jedynie całymi kolumnami. Narzuca to zatem kształt obiektów sprzętowych możliwych do umieszczania w tych układach. Wysokość obiektu ma pełną wysokość struktury reprogramowalnej, co jeszcze bardziej ułatwia gospodarowanie strukturą w celu jej współdzielenia. Taka organizacja obiektów znakomicie ułatwia również dostęp z zewnątrz do obiektów w celu wymiany danych. Problem łączności, wymiany między obiektami wewnątrz struktury oraz niespójności systemu przy rekonfigurowaniu jego części, Xilinx rozwiązał wprowadzając element Bus-Makro (rys. 6). Bus-Makro jest w istocie magistralą trójstanową, która oddziela od siebie poszczególne moduły – obiekty sprzętowe.



Rys. 6. Wymiana danych obiektów sprzętowych z otoczeniem i pomiędzy sobą

6. Podsumowanie

Przedstawione argumenty pokazują, że zagadnienie RC dojrzeźwa technologicznie do praktycznego zastosowania w systemach o wielkich mocach obliczeniowych. W zagadnieniach obliczeniowych, z którymi zetknęli się autorzy, prowadzone są obliczenia, których realizacja na superkomputerach zainstalowanych w ACK „Cyfronet” liczona jest w dniach. Mowa tu o obliczeniach kwantowo-chemicznych [13]. Pośród prowadzonych obliczeń jako najbardziej wymagające czasowo wyróżnić można obliczenia znajdujące praktyczne zastosowania w farmacji i katalizie. Wiele potencjalnie istotnych zagadnień z tych dziedzin nie jest w ogóle podejmowanych ze względu na niewystarczającą dostępną moc obliczeniową. W wyniku realizacji przyspieszenia sprzętowego obliczenia w ACK „Cyfronet” AGH będą mogły odbywać się szybciej.

Natura obliczeń kwantowych pozwala na liniowe skalowanie czasu obliczeń wraz z liczbą jednostek obliczeniowych. Każda jednostka będzie prowadziła obliczenia w środowisku hybrydowym HW-SW. Wyposażona one będzie w odpowiednią kartę rozszerzeń z koprocesorem sprzętowym. Celem realizowanego projektu będzie opracowanie takiego koprocesora z możliwością jego powielania w wielu egzemplarzach w celu przyspieszenia obliczeń w wielu węzłach. Realizacja wspomnianego koprocesora za pomocą technologii FPGA pozwoli na praktyczną realizację przy użyciu konkurencyjnych w stosunku do superkomputerów i kosztów.

W operacjach realizowanych przez oprogramowanie kwantowe ważną klasę operacji stanowią operacje wektorowe na macierzach. Ten typ operacji znany z problemów przetwarzania i obróbki sygnałów z powodzeniem udało się przenieść do realizacji sprzętowych. Wszelkie operacje multimedialne typu cyfrowa filtracja sygnałów czy też różne dekompozycje częstotliwościowe są w swej istocie operacjami mnożenia macierzy. Praktyka pokazała, że to pole zastosowań aplikacyjnych FPGA jest obecnie podstawowym obszarem zastosowań układów reprogramowalnych. Nie ma zatem przeszkód aby zastosować te układy również do obliczeń chemicznych.

Alternatywne podejście polega na budowaniu instalacji sprzętowych pozwalających na wykorzystanie masywnej równoległości. Wymaga ono poniesienia ogromnych kosztów, i w praktyce stosowane jest jedynie przez bardzo nieliczne ośrodki. Należy dodać, że ośrodki te, dysponując odpowiednimi zasobami, wywierają duży wpływ na rozwój algorytmów powszechnie stosowanych w obliczeniach kwantowo-chemicznych. Zagadnienia akceleracji sprzętowej specyficznej dla obliczeń kwantowo-chemicznych było badane jedynie w stosunkowo wąskim zakresie. O tyle problem nie jest wyczerpany i jego badanie może doprowadzić do sformułowania nowych, lepiej dopasowanych do takiego rozwiązania algorytmów.

Literatura

- [1] Atmel: *AT6000 Configuration Guide. FPGA Configuration Guide*. www.atmel.com
- [2] Atmel: *AT40K Series. Configuration AT40K Application Notes*. www.atmel.com
- [3] Xilinx: *XC6200 Field Programmable Gate Array. Users Manual*. www.xilinx.com
- [4] Barr M.: *Reconfigurable Computing Primer*. Multimedia Systems Design, September 1998, 44–47.
- [5] Carvalho E., Briao E., Moeller L., Moeller F., Moraes F., Calazans N.: *Controlling Configurations on Dynamic Reconfigurable Systems*. XIX SIM-South Symposium on Microelectronics
- [6] Celoxica: *DK Design Suit Datasheet*. www.celoxica.com
- [7] Russek P., Wiatr K.: *FPGA Based Vector Quantizer for Bitrate Control in Image Compression*. ERSA'02, Las Vegas, Nevada 2002
- [8] Silicon Graphics: *Extraordinary Acceleration of Workflows with Reconfigurable Application-specific Computing from SGI*. White Paper, November 2004
- [9] Villasenor J., Jones Ch., Schoner B.: *Video Communications Using Rapidly Reconfigurable Hardware*. IEEE Trans. on Circuit and Systems for Video Technology, vol. 5, Dec. 1995, 565–567
- [10] Xilinx: *Virtex II Platform FPGA, Complete Data Sheet*.
- [11] Wiatr K.: *Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego*. Kraków, UWND AGH 2002
- [12] Wiatr K.: *Akceleracja obliczeń w systemach wizyjnych*. Warszawa, WNT 2003
- [13] Gaussian 03, Revision C.02, Gaussian, Inc., Wallingford CT, 2004