

Metody częściowej rekonfiguracji układów FPGA

Mgr inż. Michał DOLIGALSKI

Absolwent Uniwersytetu Zielonogórskiego (2006). Ukończył studia o specjalności Inżynieria Komputerowa. Od roku 2006 pracuje jako asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: M.Doligalski@iie.uz.zgora.pl

**Dr inż. Marek WĘGRZYN**

Adiunkt w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Pełni funkcję kierownika Zakładu Inżynierii Komputerowej. Zainteresowania badawcze obejmują zagadnienia projektowania systemów cyfrowych, ze szczególnym uwzględnieniem logiki programowalnej i języków opisu sprzętu (VHDL i Verilog).

e-mail: M.Wegrzyn@iie.uz.zgora.pl



Streszczenie

Układy reprogramowalne mają szereg różnych zastosowań. Mogą być wykorzystane w systemach sterowania, akwizycji danych, czy też kompleksowych rozwiązaniach typu *smart home*. Jedną z najważniejszych cech układów FPGA jest możliwość ich swobodnej rekonfiguracji. Przygotowanie plików konfiguracyjnych odbywa się w środowisku CAD/CAE (np. Xilinx ISE). Powszechnie stosowana jest pełna, statyczna rekonfiguracja układów. Wymaga ona zatrzymania pracy układu na czas rekonfiguracji. W związku z tym, że zmianie ulega cała konfiguracja układu, w niektórych przypadkach, proces ten może być czasochłonny. Alternatywą jest przeprowadzenie częściowej rekonfiguracji układów FPGA. Uzyskuje się w ten sposób zmniejszenie czasu rekonfiguracji oraz dodatkową funkcjonalność. Artykuł omawia rodzaje rekonfiguracji układów FPGA oraz zalety takiego podejścia. Porównuje również metody częściowej różnicowej i częściowej modułowej rekonfiguracji.

Abstract

There are a lot of applications of FPGA devices. It can be used in control systems, data acquisition, or in complex solutions like *smart home*. One of the most important features of FPGA devices is possibility of random reconfiguration. Configuration file's preparing is made using CAD/CAE tools (e.g. Xilinx ISE). Full static reconfiguration is the most widespread method. There is necessary to stop the device during its reconfiguration. Partial reconfiguration of FPGA devices is the alternative method. This paper presents methods and advantages of partial reconfiguration.

Słowa kluczowe: FPGA, częściowa rekonfiguracja, statechart.

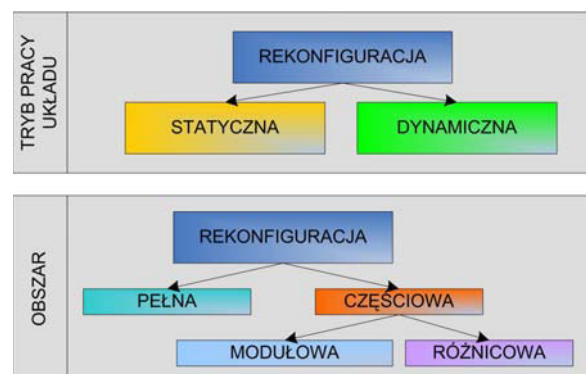
Keywords: FPGA, partial reconfiguration, statechart.

1. Wstęp

Układy FPGA znajdują zastosowanie jako układy prototypowe w procesie projektowania układów typu ASIC; są również z powodzeniem wykorzystywane w systemach sterowania [2], akwizycji danych, filtrów cyfrowych [13], sprzętowej akceleracji obliczeń i in. [1]. Układy te mogą być wykorzystywane autonomicznie, bądź jako elementy większych systemów. Dla krótkich serii koszt wyprodukowania układów FPGA jest niższy w porównaniu do układów typu ASIC. Główną zaletą układów FPGA jest możliwość ich rekonfiguracji [7,11].

2. Rodzaje rekonfiguracji

Można wyróżnić kilka rodzajów rekonfiguracji (Rys.1). W zależności od trybu pracy układu rekonfiguracja może być statyczna, bądź dynamiczna. Ze względu na obszar podlegający rekonfiguracji można wyróżnić rekonfigurację pełną oraz częściową. Tryby rekonfiguracji układów FPGA mogą różnić się w zależności od producenta układu. Co więcej, układy jednego producenta mogą wykorzystywać tylko niektóre rodzaje rekonfiguracji. W przypadku układów FPGA firmy Xilinx istnieją dwa sposoby przeprowadzenia częściowej rekonfiguracji: różnicowa bądź modułowa [11]. Częściowa rekonfiguracja jest możliwa w wybranych układach firmy Xilinx, np. Spartan-3, Spartan-3E, Virtex, Virtex-E, Virtex-II Pro i nowszych.



Rys. 1. Rodzaje rekonfiguracji

Fig. 1. Types of reconfigurations

Jak już wcześniej wspomniano, istnieje kilka sposobów rekonfiguracji układów FPGA. Powszechnie stosowana jest pełna statyczna rekonfiguracja - jest to najprostsza i najczęściej wykorzystywana metoda rekonfiguracji układów. Zmianie ulega cały obszar matrycy programowalnej. W związku z tym rozmiar plików konfiguracyjnych (ang. *bitstream*) jest proporcjonalny do rozmiaru układu, który będzie podlegał rekonfiguracji. Konieczne jest również zatrzymanie układu na czas rekonfiguracji. Metoda ta posiada ograniczenia związane z rozmiarem plików konfiguracyjnych - jest to szczególnie istotne w rozproszonych systemach sterowania [10]. W tym przypadku krytycznym parametrem jest rozmiar plików konfiguracyjnych. Związane jest to z ograniczonym pasmem do transmisji danych pomiędzy poszczególnymi węzłami. Rozmiar plików konfiguracyjnych wpływa bezpośrednio na czas rekonfiguracji.

W procesie pełnej statycznej rekonfiguracji, można wyróżnić niezależne ścieżki projektowe dla poszczególnych konfiguracji układu. Jedyną częścią wspólną, dla poszczególnych konfiguracji, może być etap specyfikacji systemu cyfrowego, najczęściej z wykorzystaniem języków opisu sprzętu VHDL oraz Verilog. Kolejne etapy (synteza i implementacja) przeprowadzane są niezależnie.

Niedogodności związane z rozmiarem kolejnych plików konfiguracyjnych eliminuje częściowa rekonfiguracja (modułowa bądź różnicowa). Zastosowanie częściowej rekonfiguracji pozwala na zmniejszenie rozmiarów plików konfiguracyjnych, uzyskanie dodatkowej funkcjonalności. Jest ona szerzej omówiona w rozdz. 3.

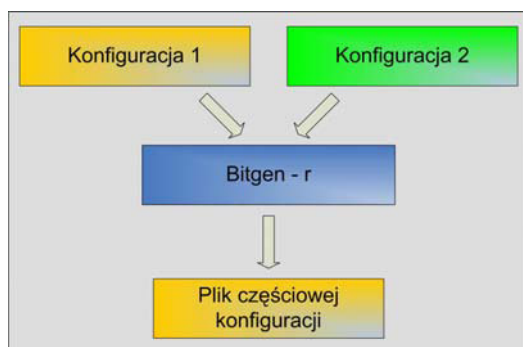
Omówione powyżej metody oparte były na rekonfiguracji jednokontekstowej: układ FPGA posiada jeden kontekst (konfigurację), która w części bądź w całości ulega wymianie. Inną metodą jest rekonfiguracja wielokontekstowa. Układ może przechowywać informacje o kilku kontekstach (konfiguracjach), jednocześnie tylko jeden kontekst jest aktywny, pozostałe mogą ulegać wymianie (modyfikacji). Rekonfiguracja polega na wymianie kontekstu nieaktywnego a następnie przełączeniu (uaktywnieniu) danego kontekstu.

Wybór właściwej metody rekonfiguracji uzależniony jest w dużej mierze od konkretnego zadania projektowego oraz zamierzo-

nej funkcjonalności. Artykuł przedstawia sposoby rekonfiguracji na przykładzie układów Xilinx, w testach wykorzystywano układy Spartan 3E oraz Virtex II PRO.

3. Rekonfiguracja różnicowa i modułowa

W procesie syntezy i implementacji otrzymywane są pliki konfiguracyjne. Przykładowo, układy Spartan 3E firmy Xilinx umożliwiają przeprowadzenie częściowej różnicowej rekonfiguracji. Rekonfiguracja różnicowa polega na porównaniu ze sobą dwóch plików: pliku konfiguracji podstawowej (np. Konfiguracja 1) oraz pliku konfiguracji po modyfikacji projektu (np. Konfiguracja 2). W wyniku otrzymuje się plik różnicowy. Generowanie pliku różnicowego odbywa się z użyciem programu *bitgen* (z przełącznikiem *-r*) (Rys.2). Narzędzie to wchodzi w skład pakietu CAD/CAE Xilinx ISE.



Rys. 2. Generowanie pliku różnicowego
Fig. 2. Generation of difference file

W przypadku rekonfiguracji różnicowej, zmiany w projekcie mogą być przeprowadzone na dwóch etapach: na etapie specyfikacji (HDL), bądź pliku NCD (ang. *Native Circuit Description*).

Pliki NCD są plikami binarnymi zawierającymi informacje na temat konfiguracji układu FPGA. Wprowadzenie zmian na poziomie specyfikacji układu pociąga za sobą konieczność powtórzenia procesu syntezy i implementacji dla zmodyfikowanej konfiguracji. Najczęściej zmiany w pliku NCD polegają na modyfikacji zawartości pamięci *Block RAM*, korekcie równań w tablicy odniesień LUT, czy też modyfikacji portów wejścia/wyjścia. Rekonfiguracja różnicowa może być zarówno statyczna, jak i dynamiczna. Tryb rekonfiguracji należy określić podczas tworzenia pliku różnicowego (*-g ActiveReconfig: Yes|No*).

Przykładem praktycznego zastosowania częściowej różnicowej rekonfiguracji układów FPGA jest mikroprogramowany układ sterujący [8]. Rozkazy (operacje) przechowywane są w pamięci BRAM. Podczas rekonfiguracji struktura układu mikroprogramowanego nie zmienia się, wymianie ulega jedynie zawartość pamięci.

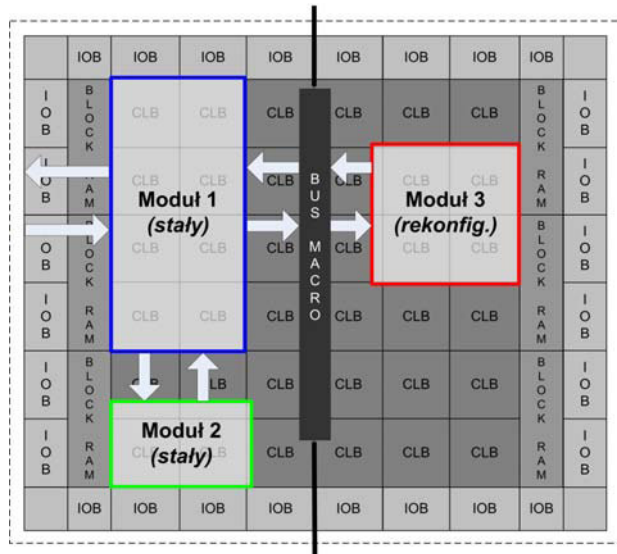
Proces projektowania układu z wykorzystaniem częściowej rekonfiguracji odbywa się w standardowy sposób – nie jest konieczne dostosowanie, czy modyfikowanie struktury układu.

Zastosowanie rekonfiguracji różnicowej jest uzasadnione, gdy wprowadzane są jedynie drobne zmiany w projekcie. W przypadku dużych zmian należy zastosować modułową rekonfigurację. Różni się ona znacznie od zaprezentowanej wcześniej. Główną ideą rekonfiguracji modułowej jest podział projektu na bloki (moduły) na etapie specyfikacji. Rekonfiguracja polega na wymianie zawartości (konfiguracji) poszczególnego modułu.

Proces projektowania oparty o częściową modułową rekonfigurację jest dość skomplikowany. Firma Xilinx opracowała przewodnik [11] opisujący ścieżkę projektową oraz proponowaną strukturę katalogów, dla projektowanego układu. Brakuje jednak na rynku narzędzi wspierających ten proces. Określone zostały również pewne ograniczenia, co do struktury układu.

Rysunek 3 przedstawia strukturę układu FPGA z zaznaczonymi trzema obszarami (modułami). Struktura układu przedstawiona jest w dość dużym uproszczeniu. Dwa moduły (*Moduł 1*, *Moduł 2*) są modułami stałymi. *Moduł 3* jest modułem podlegającym re-

konfiguracji. Moduły podlegające rekonfiguracji muszą komunikować się pomiędzy sobą, jak i pozostałymi modułami za pomocą odpowiednio przygotowanych magistral (ang. *bus macro*). W omawianym przypadku jest to *Moduł 3*, połączony jest on za pomocą magistrali z *Modułem 1*.



Rys. 3. Rozmieszczenie modułów w strukturze układu FPGA
Fig. 3. Modules placement in a FPGA device

Moduły te mają narzucone pewne ograniczenia. Wysokość każdego modułu jest równa wysokości zastosowanego układu FPGA. Każdy moduł musi mieć szerokość równą czterem blokom logicznym lub wielokrotności czterech bloków, lewy brzeg modułu musi być położony odpowiednio w 0, 4, 8 .. itd. kolumnie.

Bloki wejścia/wyjścia leżące bezpośrednio powyżej i poniżej modułu rekonfigurowanego należą do jego zasobów. Jeśli moduł znajduje się przy lewym lub prawym brzegu układu FPGA, przylegająca pamięć BRAM oraz moduły wejścia/wyjścia należą do jego zasobów.

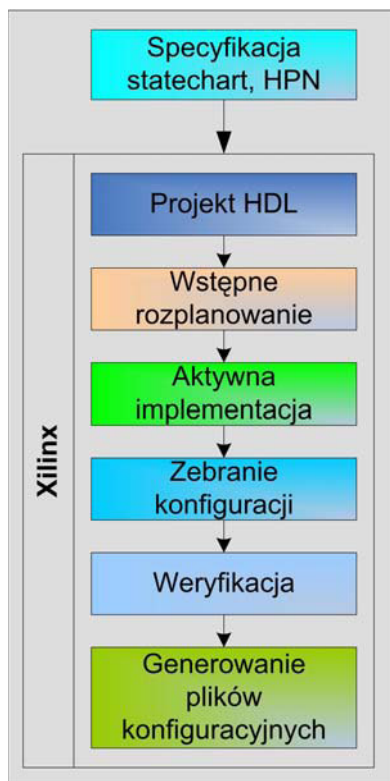
W porównaniu do rekonfiguracji różnicowej, częściowa modułowa rekonfiguracja pozwala na modyfikowanie struktury układu w znacznie szerszym zakresie. Liczba modułów, które mogą podlegać rekonfiguracji, ograniczona jest jedynie wielkością (szerokością) układu. Maksymalną liczbę modułów można obliczyć dzieląc liczbę kolumn bloków logicznych przez 4. Producent sugeruje, aby w projektowanej strukturze znajdował się jedynie jeden moduł podlegający rekonfiguracji. Sugestia ta wynika z faktu, iż zwiększanie liczby modułów pociąga za sobą wzrost liczby plików projektu, a w końcowej fazie plików konfiguracyjnych.

3. Projektowanie pod kątem częściowej modułowej rekonfiguracji

Proces projektowania układu pod kątem częściowej modułowej rekonfiguracji jest dość skomplikowany. Firma Xilinx zaproponowała ścieżkę projektową dla projektów modułowych [12]. Proces projektowania można znacznie uprościć stosując odpowiednie formy specyfikacji. Specyfikacja zorientowana na modułową rekonfigurację powinna umożliwiać modelowanie na różnych poziomach abstrakcji. Dwie proponowane formy specyfikacji to hierarchiczne sieci Petriego (HPN) [1] oraz diagramy maszyny stanowej UML (ang. *statechart*) [6]. W przypadku sieci Petriego modułem jest makromiejsce. Diagramy maszyny stanowej posiadają stany złożone – stan taki może być również traktowany jako pojedynczy moduł. Diagramy utworzone na etapie specyfikacji mogą być również wykorzystane jako dokumentacja projektu. Ścieżka projektowa, rozszerzona o hierarchiczną formę specyfikacji, przedstawia rysunek 4.

Na początkowym etapie projektowania opracowany jest syntezowany model w języku HDL, z uwzględnieniem założeń dla modułowej rekonfiguracji. Na tym etapie są tworzone i weryfikowane poszczególne moduły systemu. Kolejnym etapem jest wstępne

rozmieszczenie zasobów. Na tym etapie opracowywany jest plik HDL, który znajduje się na szczycie hierarchii. Określa się, które moduły będą podlegały rekonfiguracji, następuje połączenie poszczególnych modułów. W projekcie umieszcza się również moduły magistral. Ponadto, określa się zależności czasowe.



Rys. 4. Ścieżka projektowa
Fig. 4. Design flow

Sytuacja, w której moduły stale komunikują się z modulem rekonfigurowanym w trakcie procesu jego rekonfiguracji jest niedopuszczalna. Na tym etapie projektowania należy szczegółowo przeanalizować działanie układu i wprowadzić mechanizmy zabezpieczające. Mechanizmy te mogą być oparte o generowanie statusów dla poszczególnych modułów – sygnalizowane tryby: normalna praca, stan oczekiwania, rekonfiguracja.

W fazie aktywnej implementacji następuje synteza i implementacja poszczególnych modułów. Tworzone są pliki NGD określające fizyczną strukturę układów oraz pliki konfiguracyjne dla poszczególnych modułów.

Faza zebrania konfiguracji odpowiada za połączenie poszczególnych modułów (bloków) w jedną całość. Zostaje utworzona konfiguracja startowa oraz konfiguracje dodatkowe. Na tym etapie widać wyraźnie złożoność projektów wykorzystujących modułową rekonfigurację. Maksymalna liczba kombinacji konfiguracji może być dość znaczna. Można ją jednak w prosty sposób obliczyć.

Przyjąć należy, że każda z możliwych wersji modułu podlegającego rekonfiguracji jest wierzchołkiem grafu skierowanego K_n , gdzie n to liczba wierzchołków grafu. Liczba kombinacji konfiguracji m może być obliczona na podstawie liczby krawędzi w danym grafie (równanie 1). Jest to zmodyfikowany wzór na liczbę krawędzi grafu pełnego.

$$m = 2 * \left(\frac{n * (n - 1)}{2} \right) \quad (1)$$

Wskazane jest, aby generowanie plików konfiguracyjnych było dodatkowo poprzedzone wizualną weryfikacją układu za pomocą odpowiedniego narzędzia (np. *FPGA Editor* z pakietu Xilinx ISE). Należy sprawdzić, czy nie występują przypadkowe przekroczenia granic pomiędzy modułami oraz czy połączenia pomiędzy nimi przeprowadzone są w sposób prawidłowy.

4. Podsumowanie

Częściowa rekonfiguracja układów FPGA pozwala na zwiększenie funkcjonalności projektowanych systemów oraz zmniejszenie wykorzystywanych zasobów sprzętowych [4,11]. Pozwala również na szybsze reprogramowanie układu. Ma to szczególne znaczenie w systemach rozproszonych oraz systemach czasu rzeczywistego, w których czas odpowiedzi układu jest parametrem krytycznym [10].

Dynamiczna (aktywna) rekonfiguracja pozwala na przeprowadzenie uaktualnień sprzętu w istniejących systemach, bez konieczności zatrzymywania jego pracy. Możliwe jest również tworzenie systemów adaptacyjnych [3], których adaptacja polegałaby na modyfikowaniu konfiguracji struktury matrycy FPGA.

Obecnie w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego prowadzone są prace nad zastosowaniem częściowej modułowej rekonfiguracji, w tym rekonfiguracji różnicowej w sterownikach cyfrowych [5]. Zastosowanie odpowiedniej formy specyfikacji pozwala na uproszczenie procesu projektowania. Hierarchiczny diagram maszyny stanowej jest doskonałym narzędziem do modelowania i graficznego projektowania sterownika. Inną, rozpatrywaną formą specyfikacji są hierarchiczne sieci Petriego. Niewątpliwą ich zaletą jest możliwość przeprowadzanie formalnej analizy (np. pod kątem żywotności czy ograniczoności) [9].

Prace są ukierunkowane na opracowanie metodologii projektowania sterowników cyfrowych wykorzystujących częściową rekonfigurację oraz opracowaniu odpowiednich narzędzi CAD/CAE wspierających, a w niektórych obszarach w pełni automatyzujących ten proces. Wsparcie projektowania ma odbywać się zarówno na etapie specyfikacji, jak i na poszczególnych etapach przygotowania plików konfiguracyjnych.

5. Literatura

- [1] M.Adamski, A.Karatkevich and M.Węgrzyn (Eds.), *Design of Embedded Control Systems*, Springer Science, Boston, 2005.
- [2] M.Adamski, M.Węgrzyn and P.Wolański, "Simulating and synthesising of reconfigurable logic controllers using VHDL", *Proc. of the 42nd International Scientific Workshop, IWK'97*, Ilmenau (Germany), 1997, Band 1, ss.522-527.
- [3] P.Bubacz, "Wykorzystanie układów rekonfigurowalnych w regulacji adaptacyjnej", *Materiały Konferencji Naukowej „Informatyka - sztuka czy rzemiosło”*, KNWS 20004, 2004.
- [4] C.Kao, *Benefits of Partial Reconfiguration*, Xilinx, Inc. 2005
- [5] M.Doligalski, M.Węgrzyn, "Partial reconfiguration-oriented design of logic controllers", *Proceedings of SPIE: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2007*, Vol. 6937, 2007.
- [6] M.Doligalski, "Specyfikacja programów sterowania oparta na hierarchicznym modelu maszyny stanów UML", *Archiwum Konferencji PTETIS*, Vol. 23, 2007, ss.299-304.
- [7] R.Hartenstein, "Reconfigurable Computing: a New Business Model - and its Impact on SoC Design", *Proceedings of the EUROMICRO Symposium on Digital Systems Design, DSD 2001*, ss.103-110.
- [8] R.Wisniewski, "Częściowa rekonfiguracja mikroprogramowanych układów sterujących implementowanych z wykorzystaniem struktur FPGA", *Archiwum Konferencji PTETIS*, Vol.21, 2005, ss.239-242.
- [9] A.Węgrzyn, „Symbolic Analysis of Logical Control Devices using Selected Methods of Petri Net Analysis”, Ph.D. Thesis, Politechnika Warszawska, 2003.
- [10] M.Węgrzyn, „Rozproszony system sterowania bezpiecznego z rekonfigurowalnymi układami lokalnymi”, *Materiały II Konferencji Naukowej „Informatyka - sztuka czy rzemiosło”*, KNWS 2005, Złotniki Lubąńskie, 2005, ss.127-132.
- [11] Xilinx, Inc., *XAPP290, Two Flows for Partial Reconfiguration: Module-Based or Difference-Based*, www.xilinx.com, 2004.
- [12] Xilinx, Inc., *Development System Reference Guide*, www.xilinx.com.
- [13] Y.-J.Oh., Ch.-S.Choi, H.Lee, C.-H.Lee, "A Reconfigurable CSD FIR Filter Design using Dynamic Partial Reconfiguration", *International SoC Design Conference, ISOC*, Seoul, Oct. 2005, ss.381-384.

Title: Methods of FPGA partial reconfiguration.

Artykuł recenzowany