

Program konwertujący tekstową sieć działań do struktur mikroprogramowanych układów sterujących ze współdzieleniem kodów

dr inż. Małgorzata KOŁOPIENCZYK

Dr inż. Małgorzata Kołopieńczyk w roku 2007 obroniła rozprawę doktorską i uzyskała tytuł doktora w dziedzinie informatyka. Od 1999 roku jest zatrudniona na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.



e-mail: m.kolopienczyk@iie.uz.zgora.pl

Streszczenie

Do opisu działania mikroprogramowanych układów sterujących wykorzystuje się sieć działań (ang. Flow-Chart of Algorithm, FCA). W artykule zaprezentowano autorskie oprogramowanie *fca2cmcu* umożliwiające przekonwertowanie, zapisanej w formie tekstowej, sieci działań do struktur mikroprogramowanych układów sterujących (ang. Compositional Microprogram Control Unit, CMCU). Wynikiem działania programu *fca2cmcu* jest plik z opisem w języku VHDL struktur mikroprogramowanych układów sterujących. Ponadto zaprezentowano typową ścieżkę projektową wykorzystującą omawiane oprogramowanie.

Abstract

To describe a compositional microprogram control unit there will be used a flow-chart of algorithm. In the paper a *fca2cmcu* software is presented, which permits to translate the text flow-chart into the structures of compositional microprogram control unit. As a result, the *fca2cmcu* program generates a valid VHDL file describing the structure of the compositional microprogram control unit. Additionally, a control unit design path with use of *fca2cmcu* is given.

Słowa kluczowe: sieć działań, mikroprogramowany układ sterujący, *fca2cmcu*.

Keywords: flow-chart of algorithm, compositional microprogram control unit, *fca2cmcu*.

1. Wstęp

Teorie o układach mikroprogramowanych (ang. Compositional Microprogram Control Unit, CMCU) powstały już na początku lat 80. Ich głównym założeniem było wykorzystanie prostych układów cyfrowych, oraz pamięci ROM do tworzenia złożonych układów sterujących. Rozwój w dziedzinie techniki cyfrowej spowodował pojawienie się zintegrowanych układów takich jak System-on-a-Chip (SoC) czy System-on-Programmable-Chip (SoPC), w których bloki funkcjonalne projektowanego układu implementowane są z wykorzystaniem matryc programowalnych FPGA [3, 7].

Do opisu działania mikroprogramowanych układów sterujących wykorzystuje się sieć działań (ang. flow-chart of algorithm).

Układy mikroprogramowane są specyficzną techniką realizacji układów sterowania, polegająca na bezpośredniej transformacji sieci działań na mikroinstrukcje tzw. mikroprogramu sterowania [5].

W artykule zaprezentowano autorskie oprogramowanie *fca2cmcu* umożliwiające przekonwertowanie, zapisanej w postaci tekstowej, sieci działań do następujących struktur mikroprogramowanych układów sterujących [4]:

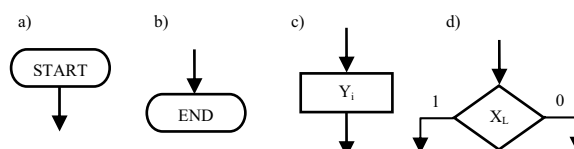
- CMCU ze współdzieleniem kodów
- CMCU z konwerterem adresów,
- CMCU z rozszerzonymi mikroinstrukcjami,
- CMCU z kodowaniem mikrooperacji,
- CMCU z elementarnymi łańcuchami,
- CMCU z elementarnymi łańcuchami i konwerterem adresów,
- CMCU z elementarnymi łańcuchami i rozszerzonymi mikroinstrukcjami,

- CMCU z elementarnymi łańcuchami i kodowaniem mikrooperacji.

2. Sieć działań

Opis układów sterujących przy pomocy sieci działań jest pierwszym krokiem w projektowaniu układów mikroprogramowanych. Specyficzną właściwością sieci działań jest wyeksponowanie stanów wejść i wyjść układów sterujących, co znacznie ułatwia realizację układu sterującego jako mikroprogramowanego [2, 4].

Składnikami sieci działań są bloki: początkowy, końcowy, operacyjny i warunkowy (rys. 1).



Rys. 1. Elementy sieci działań - bloki: a) początkowy; b) końcowy; c) operacyjny; d) warunkowy

Fig. 1. Components of flow-chart – block: a) initial; b) final; c) operational; d) conditional

W sieci działań przepływ sterowania odbywa się po drodze od bloku początkowego przez bloki operacyjne i warunkowe. Bloki operacyjne opisują stan wyjść układu sterującego, zaś bloki warunkowe analizują stan jego wejść. Symbole sygnałów wyjściowych Y_i wpisane w blokach operacyjnych nazywane są mikroinstrukcjami.

Przy syntezie mikroprogramowanego układu sterującego na podstawie danej sieci działań istotne jest właściwe przyporządkowanie stanów wewnętrznych układu sterowania poszczególnym segmentom sieci działań. Przyporządkowanie to implikuje jednocześnie podział sieci działań na mikroinstrukcje, tj. takie jej segmenty, które są wykonywane w jednym, elementarnym takcie pracy układu sterującego [5].

3. Format danych wejściowych

Do opisu danych wejściowych wykorzystano formę tekstową sieci działań [1]. Reprezentacja tekstowa sieci działań musi zawierać wszystkie informacje o strukturze sieci, jak również o wszystkich sygnałach wejściowych i wyjściowych. Zastosowany w pracy format spełnia te wymagania.

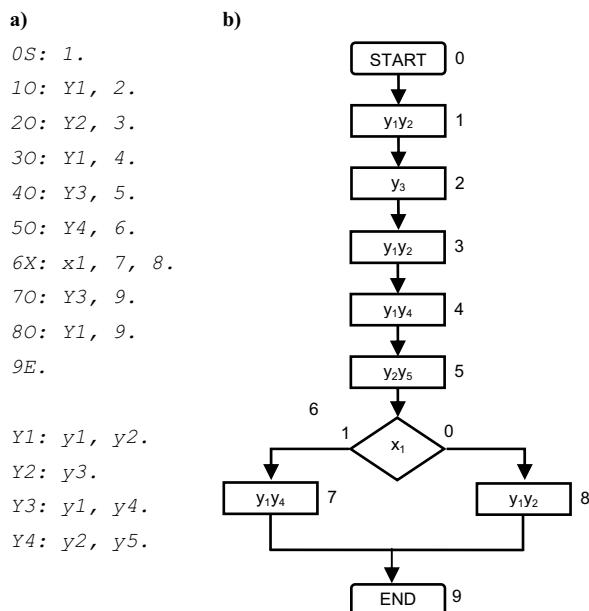
W pliku można wyróżnić dwie części: pierwszą opisującą strukturę sieci, w której elementy zaczynają się od numeru bloku oraz drugą opisującą sygnały wyjściowe poszczególnych bloków, w której elementy zaczynają się od znaku Y. Opis poszczególnych symboli używanych przy tworzeniu formy tekstowej sieci działań zamieszczono w tabeli 1.

Tab. 1. Symbole wykorzystywane do opisu sieci działań (# - numer)

Tab. 1. Symbols used to describe flow-chart (# - number)

Symbol	Znaczenie
#S: #.	Blok początkowy
#E.	Blok końcowy
#O: Y#, #.	Blok operacyjny, Y - zbiór wyjść bloku
#X: x#, #, #.	Blok warunkowy, x - sygnał wejściowy
Y#: y#, ..., y#.	Kolekcja mikrooperacji (sygnałów wyjściowych)

W części opisującej strukturę sieci każda cyfra na początku linii oznacza numer bloku, następnie podawany jest symbol oznaczający rodzaj bloku. Ostatnia cyfra w linii oznacza numer bloku następnego. W przypadku bloku warunkowego X przedostatnia cyfra oznacza numer bloku następnego dla ścieżki w kierunku prawdy, a ostatnia numer bloku następnego dla ścieżki w kierunku fałszu. Plik ma strukturę listy połączeń. Na rysunku 2 przedstawiono przykładową sieć działań z odpowiadającym jej opisem tekstowym [1, 4].

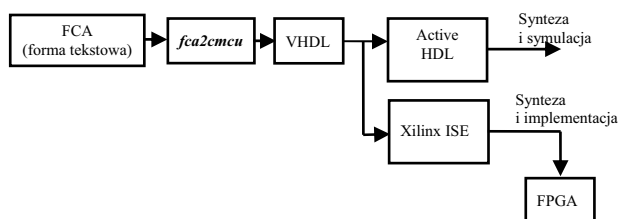


Rys. 2. Sieć działań: a) reprezentacja tekstowa; b) reprezentacja graficzna
Fig. 2. Flow-chart: a) textual representation; b) graphical representation

4. Budowa programu *fca2cmcu*

Program *fca2cmcu* został napisany w Języku ANSI C i może być uruchamiany na dowolnej platformie, dla której istnieje kompilator Języka ANSI C.

Na rysunku 3 przedstawiono typową ścieżkę projektową z wykorzystaniem oprogramowania *fca2cmcu* [4].



Rys. 3. Ścieżka projektowa z wykorzystaniem oprogramowania *fca2cmcu*
Fig. 3. Design path using the *fca2cmcu* software

Wejściem dla programu jest plik z opisem tekstowym sieci działań. Podstawowym zadaniem programu jest translacja tekstowej postaci sieci działań do struktur mikroprogramowanych układów sterujących opisanych w języku VHDL.

Wynikiem działania programu *fca2cmcu* jest plik z opisem w języku VHDL struktur mikroprogramowanych układów sterujących realizujących algorytm sterowania zapisany w postaci sieci działań.

Dla każdej sieci wejściowej generowane jest osiem struktur w języku VHDL [4, 6]:

- CMCU_CS ze współdzieleniem kodów,
- CMCU_AT z konwerterem adresów,
- CMCU_XM z rozszerzonymi mikroinstrukcjami,
- CMCU_EM z kodowaniem mikrooperacji,
- CMCU_EOLC_CS z elementarnymi łańcuchami,
- CMCU_EOLC_AT z elementarnymi łańcuchami i konwerterem adresów,

- CMCU_EOLC_XM z elementarnymi łańcuchami i rozszerzonymi mikroinstrukcjami,
- CMCU_EOLC_EM z elementarnymi łańcuchami i kodowaniem mikrooperacji.

Ponadto w wyniku działania programu użytkownik otrzymuje raport, zawierający rozmiary pamięci mikroprogramowanych układów sterujących wygenerowanych dla każdej testowanej sieci działań.

Każdy plik w języku VHDL składa się z części opisującej poszczególne moduły oraz z części głównej opisującej strukturę danego układu mikroprogramowanego. Na rysunku 4 zaprezentowano fragment opisu układu w języku VHDL.

```
CMCU_OLC_Ux_CC -----
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity TF is
  port(
    set : in  STD_LOGIC;
    rst : in  STD_LOGIC;
    q   : out STD_LOGIC
  );
end TF;

entity CM is
  port(
    clk  : in  STD_LOGIC;
    fetch : in  STD_LOGIC;
    addr  : in  STD_LOGIC_VECTOR (1 to 7);
    data  : out STD_LOGIC_VECTOR (0 to 8)
  );
  attribute rom_extract : string;
  attribute rom_extract of CM: entity is "yes";
  attribute rom_style : string;
  attribute rom_style of CM: entity is "block";
end CM;
```

Rys. 4. Fragment pliku z opisem CMCU w VHDL

Fig. 4. A fragment of the output file with CMCU description using VHDL

5. Podsumowanie

W artykule zaprezentowano program *fca2cmcu* umożliwiający translację tekstowego opisu sieci działań do struktur mikroprogramowanych układów sterujących zapisanych w języku VHDL. Dla każdej przetwarzanej sieci działań program generuje plik zawierający rozmiary pamięci mikroprogramowanych układów sterujących. Kolejnym krokiem w ścieżce projektowej wykorzystującej zaprezentowane oprogramowanie może być synteza i symulacja działania struktur mikroprogramowanych w środowisku *Active-HDL* oraz implementacja na przykład z wykorzystaniem pakietu *Xilinx ISE*.

6. Literatura

- [1] Barkalov A.A., Salomatin V.A., Starodubov K.E: System of CAD for design of control units with PALs and ROMs. Control systems and machines, No. 5, 1991, s.22-26.
- [2] Barkalov A., Węgrzyn M.: Design of control units with programmable logic. University of Zielona Góra Press 2006.
- [3] Grushnitsky R., Mursaev A., Ugrjumov E.: Development of systems on chips with programmable logic. – Spb: BHV – Petersburg, 2002 (in Russian)- 626 p.
- [4] Kołopieńczyk M.: Zastosowanie konwertera adresów do zmniejszenia rozmiaru pamięci mikroprogramowanego układu sterującego ze współdzieleniem kodów, Rozprawa doktorska, Uniwersytet Zielonogórski, Zielona Góra 2007.
- [5] Łuba T.: *Synteza układów cyfrowych*, Praca zbiorowa pod redakcją prof. Tadeusza Łuby Warszawa: WKŁ 2003, s. 296.
- [6] Titarenko L., Kołopieńczyk M.: Optymalizacja mikroprogramowanego układu sterującego poprzez zastosowanie współdzielenia kodów, Pmiary Automatyka Kontrola 2006, nr 6, wyd. spec. s. 53-55.
- [7] Maxfield C.: *The Design Warriors Guide to FPGAs*. Amsterdam: Elsevier 2004.

Title: Software converting text flow-chart into microprogram control unit with codes sharing

Artykuł recenzowany