

Małgorzata KOŁOPIEŃCZYK
UNIwersytet Zielonogórski

Koncepcja systemu wspomagającego projektowanie mikroprogramowanych układów sterujących

Dr inż. Małgorzata KOŁOPIEŃCZYK

Absolwentka Wydziału Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego (1999 r.). Obecnie adiunkt w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zajmuje się następującymi zagadnieniami: projektowaniem i syntezą sterowników logicznych oraz inżynierią oprogramowania.

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



Streszczenie

W artykule zaprezentowano aplikację wspomagającą proces projektowania mikroprogramowanych układów sterujących (ang. Compositional Microprogram Control Unit, CMCU). Do opisu algorytmu sterowania układów mikroprogramowanych wykorzystano sieć działań (ang. flow-chart). Głównym zadaniem aplikacji jest usprawnienie procesu opisu układów sterujących za pomocą sieci działań jak również translacja graficznej formy sieci działań do specyfikacji tekstowej. W artykule zaproponowano i omówiono struktury danych umożliwiające tworzenie, przechowywanie oraz prezentację sieci działań.

Słowa kluczowe: sieć działań, mikroprogramowany układ sterujący, reprezentacja tekstowa i graficzna sieci działań

Concept of system aiding design of compositional microprogram control unit

Abstract

This paper presents a concept of a system which supports the design process of compositional microprogram control units. The control algorithm of the compositional control unit is very often described by flow-charts. The proposed approach improves the design process of microprogram control unit algorithm defined by flow-charts. It allows to translate the graphical flow-charts representation into a textual flow-chart specification form as well. In this paper the data structures for creation, storage and presentation of flow-charts is also proposed. This paper has been divided into four parts. First chapter is a brief introduction to the issues of compositional microprogram control unit design. In the second chapter graphical and textual flow-chart specification is presented. In chapter third the structure of the data formats and programmable environment are described. The last – fourth – chapter contains a summary.

Keywords: flow-chart, compositional microprogram control unit, textual and graphical flow-charts representation

1. Wstęp

Koncepcja mikroprogramowanego układu zaprezentowana została po raz pierwszy w 1951 r. przez M. V. Wilkensa [8]. Powstała ona w wyniku poszukiwań systematycznych metod projektowania układów sterujących komputerów. Podstawową jej zaletą jest przejrzystość i elastyczność. Ciągowi operacji elementarnych, z których składa się cykl wykonania rozkazu, odpowiada ciąg mikrorozkazów wykonywanych przez stosunkowo prostą sieć logiczną o regularnej strukturze.

W początkowym okresie rozwoju techniki komputerowej, sterowanie wykonaniem operacji odbywało się za pomocą wbudowanych na stałe układów cyfrowych (bramki, liczniki, przerzutniki, sumatory, rejestry, itp.), a wszystkie wykonywane przez komputer funkcje były „rozrzucone” po całej wewnętrznej strukturze logicznej komputera. Takie podejście do projektowania urządzeń uniemożliwiało użytkownikowi zmianę tzw. listy rozkazów, po-

nieważ rodzaj sterowania był zawsze wkomponowany w strukturę logiczną komputera [3, 4].

W związku z tym zaczęto poszukiwać nowych rozwiązań projektowania układów sterujących komputerów i tak oto w ich wyniku zamiast ustalonych układów cyfrowych o nieregularnej strukturze wprowadzono pamięć stałą ROM, do której można na stałe zapisywać sekwencje funkcji sterujących, czyli tzw. mikroprogram. W oryginalnej koncepcji Wilkensa mikroprogram jest realizowany przez wybieranie różnych linii z jego matrycy. Sygnały sterujące sterują odpowiednim przesyłaniem informacji w komputerze i powodują określone skutki. Wybór określonego mikrorozkazu powoduje przygotowanie w matrycy stanu następnego, którego to kod po odpowiednim opóźnieniu (niezbędnym do realizacji poprzedniego mikrorozkazu) jest przesyłany do deszyfrowania i deszyfrowany. Rozgałęzienia w algorytmie sterowania są możliwe dzięki doprowadzeniu sygnałów warunkowych do matrycy stanu. Początek obszaru realizowanego programu jest zależny od kodu operacji rozkazu, który ma być wykonywany i jest przechowywany w rejestrze [8].

Upowszechnienie metody mikroprogramowania w technice cyfrowej nastąpiło dopiero wraz z rozwojem technologii układów scalonych i opracowania szybkich pamięci półprzewodnikowych ROM [8].

W chwili obecnej mikroprogramowane układy sterujące znajdują zastosowanie w wielu dziedzinach naszego życia. Coraz większe zapotrzebowanie na tego typu układy pociąga za sobą zapotrzebowanie na narzędzia wspomagające proces projektowania tych układów.

W artykule zaproponowano koncepcję systemu wspomagającego proces projektowania mikroprogramowanych układów sterujących, dla których algorytm sterowania został opisany z wykorzystaniem sieci działań. Omówiono również struktury danych umożliwiające tworzenie, przechowywanie oraz prezentację sieci działań.

Aplikacja wspomagająca projektowanie mikroprogramowanych układów sterujących została zaimplementowana w języku Java, dzięki czemu jest łatwo przenaszalna na inne systemy operacyjne wspierające Javę, takie jak Windows, SUN, Linux czy BSD. Powstały kod jest niezależny od systemu operacyjnego i procesora.

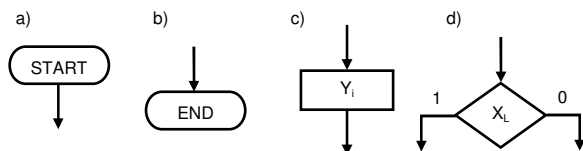
System wspomagający projektowanie mikroprogramowanych układów sterujących umożliwia zapisanie sieci działań do jednego z trzech dostępnych w systemie formatów.

Omawiany system jest ukierunkowany w stronę implementacji z wykorzystaniem języków opisu sprzętu takich jak VHDL i Verilog. Umożliwi to użytkownikom aplikacji syntezę i symulację działania mikroprogramowanych układów sterujących w środowisku Active HDL oraz implementację na przykład z wykorzystaniem pakietu Xilinx ISE. Planowana jest również rozbudowa systemu o moduł umożliwiający zapisanie sieci działań w formacie KISS2.

2. Sieć działań

Opis układów sterujących przy pomocy sieci działań (ang. flow-chart) jest pierwszym krokiem w projektowaniu układów mikroprogramowanych. Główną właściwością sieci działań jest wyekspozowanie stanów wejść i wyjść układów sterujących, co znacznie ułatwia realizację układu sterującego jako mikroprogramowanego [2].

Sieć działań składa się z następujących elementów: bloku początkowego, bloku końcowego oraz bloku operacyjnego i warunkowego (Rys. 1).



Rys. 1. Elementy sieci działań
Fig. 1. Components of flow-chart

Sieć działań można zdefiniować jako poprawne połączenie bloków operacyjnych, warunkowych, początkowego i końcowego.

Połączenie bloków jest poprawne, jeżeli zostaną spełnione dwie podstawowe reguły [1, 5]:

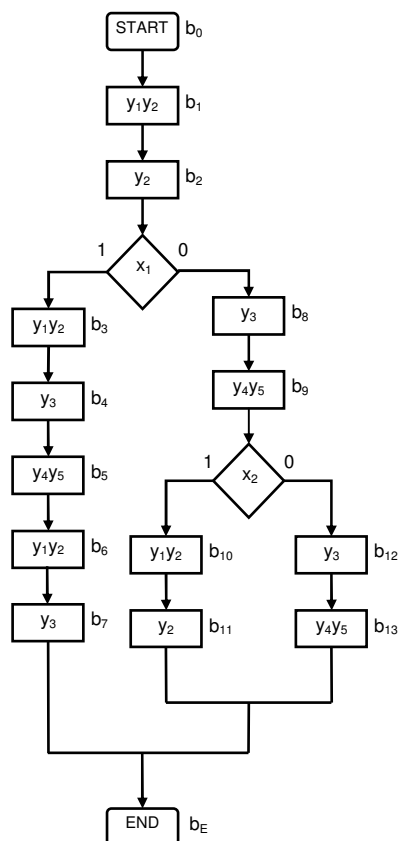
- do każdego bloku, z wyjątkiem początkowego, może prowadzić wiele połączeń,
- z każdego bloku, z wyjątkiem końcowego, może wychodzić tylko jedno połączenie.

Przepływ sterowania w sieci działań odbywa się po drodze od bloku początkowego przez bloki operacyjne i warunkowe. Bloki operacyjne opisują stan wyjść układu sterującego, zaś bloki warunkowe stan jego wejść. Symbole sygnałów wyjściowych wpisane w blokach operacyjnych nazywane są mikroinstrukcjami [6].

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 [7].

2.1. Graficzna reprezentacja sieci działań

Reprezentacja graficzna sieci działań jest schematem blokowym zawierającym wymienione powyżej cztery rodzaje elementów. Schemat blokowy reprezentuje, przez figury geometryczne połączone strzałkami, kolejne czynności w projektowanym algorytmie. Przykładową sieć działań przedstawia rysunek 2.



Rys. 2. Sieć działań
Fig. 2. Flow-chart

2.2. Tekstowa reprezentacja sieci działań

Sieć działań można także przedstawić w postaci tekstowej. Taka reprezentacja musi jednak zawierać wszystkie informacje o strukturze sieci, jak również o wszystkich sygnałach wejściowych i wyjściowych.

W omawianej aplikacji wykorzystano format tekstowy sieci działań opisany w pracy [1], który spełnia wymienione wyżej wymagania.

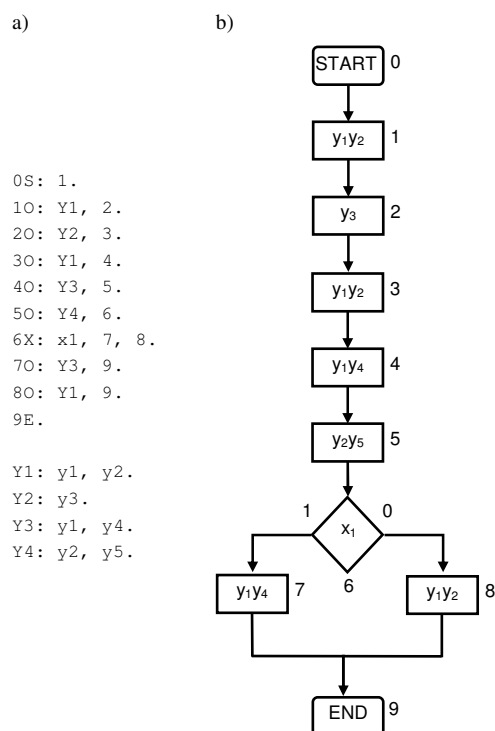
W pliku opisującym sieć działań można wyróżnić dwie części: pierwszą opisującą strukturę sieci, w której elementy zaczynają się od numeru bloku oraz drugą opisującą sygnały wyjściowe poszczególnych bloków, w której elementy zaczynają się od znaku Y. Opis poszczególnych symboli używanych przy tworzeniu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

Tab. 1. Znaczenie symboli wykorzystywanych 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 prawdy, a ostatnia numer bloku następnego dla ścieżki w kierunku fałszu.

Plik ma strukturę listy połączeń. Na rysunku 3 przedstawiono przykładową sieć działań z odpowiadającym jej opisem tekstowym.



Rys. 3. Sieć działań: a) reprezentacja tekstowa, b) reprezentacja graficzna
Fig. 3. Flow-chart: a) textual representation, b) graphical representation

Definicje dotyczące sieci działań jak również szczegółowe zasady obowiązujące przy jej tworzeniu zostały opisane w pracy [5].

3. Format danych i środowisko programistyczne

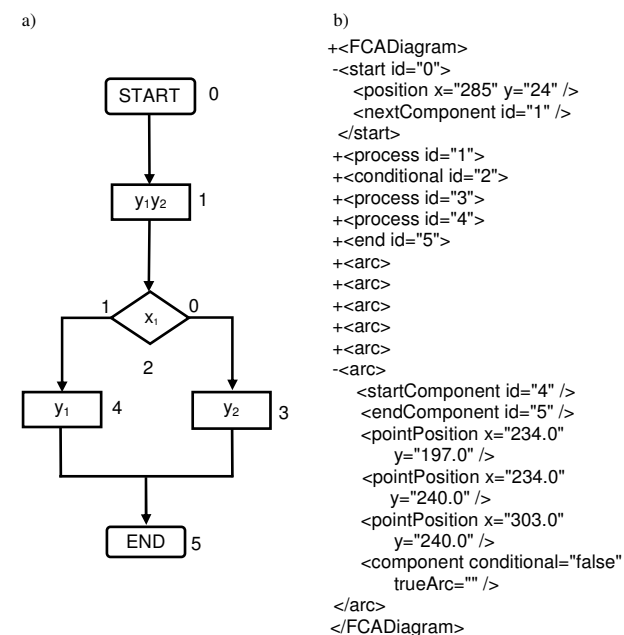
Aplikacja wspomagająca projektowanie mikroprogramowanych układów sterujących została wykonana w języku Java. Język ten jest jednym z nowszych języków programowania, który dzięki swoim zaletom ciągle zyskuje nowych użytkowników. Język Java ma wbudowany system ochrony i bezpieczeństwa danych, który zapewnia komfort zarówno programistom, jak i użytkownikom programów Javy. Dzięki Javie można w prosty sposób realizować zaawansowane zagadnienia programistyczne, takie jak: programowanie sieciowe, łączność pomiędzy bazami danych i wielowątkowość. Programy napisane przy użyciu tego języka są również łatwo przenaszalne na inne systemy operacyjne wspierające Jave. Wszystkie wyżej wymienione zalety przyczyniły się do wyboru Javy jako języka wykorzystanego do stworzenia omawianej aplikacji.

Głównym zadaniem systemu jest możliwość łatwego tworzenia sieci działań opisującej algorytm sterowania mikroprogramowanego układu sterującego, a następnie zapisanie utworzonego modelu do jednego z trzech dostępnych w aplikacji formatów: pliku tekstowego, dokumentu XML oraz pliku graficznego JPEG.

W aplikacji przewidziano również możliwość importu sieci działań zapisanej tylko w formacie tekstowym do formatu graficznego.

W związku z tym system wspomagający projektowanie mikroprogramowanych układów sterujących obsługuje dwa formaty danych wejściowych. Jednym z nich jest format XML, który umożliwił odczyt i zapis plików zawierających sieci działań. Skorzystano tu z dostępnego w języku Java parsera DOM, który po sprawdzeniu poprawności formatu pliku, umieszcza elementy dokumentu w strukturze drzewiastej. Następnie przeszukując kolejne węzły dokumentu, parser odpowiednio je interpretuje, a zawarte w nich informacje zamienia na odpowiednie obiekty.

Na rysunku 4 pokazano przykładową graficzną sieć działań i jej odpowiednik w postaci dokumentu XML.



Rys. 4. Sieć działań: a) reprezentacja graficzna, b) reprezentacja w postaci dokumentu XML

Fig. 4. Flow-chart: a) graphical representation, b) XML representation

Drugim formatem danych obsługiwanym przez aplikację jest plik tekstowy. Umożliwiło to zaimportowanie oraz wyeksportowanie sieci działań do pliku tekstowego. Wczytanie pliku odbywa się linia po linii.

Działanie i budowa aplikacji została oparta na strukturze pliku tekstowego. Rozwiązanie takie nie oferuje jednak możliwości zapisu oraz odczytu pozycji poszczególnych elementów sieci działań.

W celu rozwiązania tego problemu zdecydowano się na adaptację, na potrzeby tworzonej aplikacji, algorytmu automatycznego rozmieszczania elementów sieci działań. Algorytm ten stara się rozmieścić powiązane ze sobą elementy blisko siebie w danym obszarze roboczym. Wyznaczenie współrzędnych następuje w każdej iteracji. Umożliwiło to wizualizację modeli sieci działań zapisanych w pliku tekstowym.

4. Wnioski

W artykule zaprezentowano system wspomagający projektowanie mikroprogramowanych układów sterujących, dla których algorytm sterowania opisywany jest z wykorzystaniem sieci działań. Aplikacja umożliwia tworzenie sieci działań a następnie zapisanie utworzonego modelu do:

- pliku tekstowego,
- dokumentu XML,
- pliku graficznego JPEG.

Prace nad implementacją aplikacji objęły również adaptację algorytmu do automatycznego rozmieszczenia elementów sieci działań w danym obszarze roboczym. Umożliwiło to wizualizację modeli zapisanych w pliku tekstowym, które nie zawierają informacji o współrzędnych poszczególnych elementów. Ze względu na szereg obliczeń algorytm automatycznego rozmieszczania elementów nie jest zbyt wydajny. Przeprowadzone testy wykazały, że im bardziej złożona sieć działań, a co za tym idzie bardziej złożona reprezentacja tekstowa, tym dłuższy czas oczekiwania na narysowanie graficznej reprezentacji.

W aplikacji zaimplementowano moduł obsługi dokumentów XML, który zapewnił możliwość bezpośredniego odczytu i zapisu na dysku lokalnym plików reprezentujących sieć działań. Ponieważ struktura dokumentu XML zawiera informacje o współrzędnych poszczególnych elementów, pozwoliło to na uniknięcie szeregu skomplikowanych obliczeń potrzebnych do prawidłowego rozmieszczenia elementów sieci, a co za tym idzie znacznie przyspieszy wczytanie danej sieci działań.

5. Literatura

- [1] A.A. Barkalov, V. A. Salomatin, K.E. Starodubow: System of CAD for design of control units with PALs and ROMs. Control systems and machines, No. 5, 1991, s.22-26.
- [2] A. Barkalov, M. Węgrzyn: Design of control units with programmable logic. University of Zielona Góra Press 2006.
- [3] R. Grushnitsky, A. Mursaev, E. Ugrjumov: Development of systems on chips with programmable logic. – SPb: BHV – Petersburg, 2002 (in Russian)- 626 p.
- [4] M. B. Gorzałczany: Układy cyfrowe metody syntezy. Tom II Układy sekwencyjne układy mikroprogramowane. Wydawnictwo Politechniki Świętokrzyskiej, Kielce 2000
- [5] M. Kołopieńczyk: Zastosowanie konwertera adresów do zmniejszenia rozmiaru pamięci mikroprogramowanego układu sterującego ze współdzieleniem kodów, Rozprawa doktorska, Uniwersytet Zielonogórski, Zielona Góra 2007.
- [6] M. Kołopieńczyk: Program konwertujący tekstową sieć działań do struktur mikroprogramowanych układów sterujących ze współdzieleniem kodów. Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne 2008, nr 6, s. 809-810.
- [7] T. Łuba: Synteza układów cyfrowych, Praca zbiorowa pod redakcją prof. Tadeusza Łuby Warszawa: WKŁ 2003, s. 296.
- [8] M. Molski: Modułowe i mikroprogramowalne układy cyfrowe. WKŁ, Warszawa 1986.