

## WYKORZYSTANIE LOGIKI SEKWENTÓW GENTZENA DO PROJEKTOWANIA KOMBINACYJNYCH UKŁADÓW CYFROWYCH

**Jacek Tkacz**

**Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski  
65-246 Zielona Góra, ul. Podgórna 50**

*e-mail: j.tkacz@iie.uz.zgora.pl*

### STRESZCZENIE

W niniejszym artykule przedstawiono metodę projektowania układu cyfrowego przeprowadzoną w rozszerzonym rachunku sekwentów. Oryginalnym, niespotykanym w literaturze pomysłem zaproponowanym przez prof. Mariana Adamskiego [1] jest sposób wykorzystania w procesie projektowania układu cyfrowego metody wnioskowania naturalnego Gentzena. Gentzen sformalizował naturalny tok myślenia człowieka podczas wnioskowania prowadzącego od twierdzeń – aksjomatów do nowych twierdzeń. Reguły wnioskowania Gentzena w naturalny sposób pozwalają przekształcić specyfikację działania układu, aż do momentu uzyskania opisu implementacji.

### 1. WPROWADZENIE

Jednym ze sposobów projektowania układów cyfrowych jest zastosowanie logiki matematycznej i zastąpienie przekształceń tablicowych przekształceniami symbolicznymi oraz formalnym wnioskowaniem. Projektowanie układów cyfrowych w sposób naturalny wiąże się z metodami logiki matematycznej [1]. Prace Gentzena, matematyka niemieckiego, zmarłego tuż po zakończeniu II wojny światowej zostały zauważone dopiero w latach 1970-1980 w związku z rozwojem metod sztucznej inteligencji oraz automatycznej dedukcji.

Jednym z pierwszych popularnych autorów prac Gentzena w zastosowaniach informatycznych był Zdzisław Pawlak [5]. Prace dotyczące wykorzystania logiki Gentzena w projektowaniu układów cyfrowych zostały zapoczątkowane przez Mariana Adamskiego [1].

Wykorzystanie logiki formalnej jako języka projektowania układów cyfrowych, a w szczególności jako narzędzia syntezy jest ciągle na poziomie badawczym. Niewielka ilość informacji w literaturze nie pozwala ocenić zaawansowania prac w niniejszym kierunku, jak i uniemożliwia wykorzystanie opracowanych już metod.

## 2. SYSTEM GENTZENA

W systemie Gentzena zastosowano dziesięć reguł wnioskowania [1, 6] dla następujących spójników logicznych: negacja „/”, alternatywa „+”, koniunkcja „\*”, implikacja „->” i równoważność „,<->”. Dla każdego spójnika zostały podane dwie reguły jego eliminowania w zależności jego usytuowania w wyrażeniu. Jedna reguła jest stosowana, gdy rugowany spójnik występuje przed znakiem wynikania logicznego, druga zaś, gdy rugowany spójnik znajduje po znaku wynikania logicznego. Reguły stosowane są zawsze do spójnika głównego formuły nieelementarnej w redukowanym sekwencie. Sekwent jest zdaniem warunkowym, w którym warunek i skutek są połączone znakiem wynikania logicznego „|-” [7]. Cały proces eliminowania powtarzany jest tak długo, aż otrzymane zostaną same sekwenty znormalizowane, czyli sekwenty niezawierające żadnych spójników logicznych. Dodatkowo w systemie Gentzena występują cztery reguły minimalizacji wyrażeń normalizowanych: tautologia, sklejanie, pochłanianie oraz consensus.

Ze względu na małą złożoność obliczeniową i ogólną znajomość, jako przykład przedstawiony zostanie dowód jednego z praw De Morgana:

a) reprezentacja symboliczna

$(a*b) |- a+/b;$

-----

2,1 ' / |- ' |- a+/b, (a\*b);

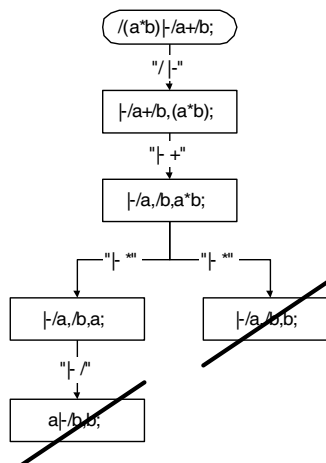
3,2 ' |- + ' |- a,/b, a\*b;

4,3 ' |- \* ' |- a,/b, a; //tautologia

5,3 ' |- \* ' |- a,/b, b;

6,5 ' |- / ' a |- b, b; //tautologia

b) reprezentacja graficzna



Rys. 1. Przebieg normalizacji przykładowego sekwentu

Z normalizacji powyższego wyrażenia nie pozostaje żaden sekwent (rys. 1). Tautologie zatrzymują dalszy proces normalizacji. Końcowe wyrażenia w poszczególnych gałęziach są zawsze prawdziwe, więc badane wyrażenie także jest zawsze prawdziwe.

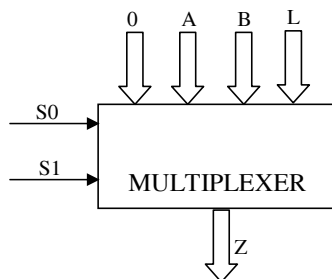
### 3. PRAKTYCZNE ZASTOSOWANIE PRZEKSZTAŁCEŃ SYMBOLICZNYCH

Wnioskowanie gentzenowskie znajduje zastosowania w różnych dziedzinach nauki. Praktycznie każdą formułę opisującą zjawisko o charakterze dyskretnym, które da się opisać w postaci wyrażeń boolowskich, można poddać naturalnemu wnioskowaniu formalnemu, w celu jego dalszego przekształcenia, a w szczególności uproszczenia [1]. Metoda symbolicznego wnioskowania doskonale nadaje się do analizy symbolicznej grafów, na przykład do wyszukiwania wszystkich maksymalnych niezależnych zbiorów wierzchołków grafu nieorientowanego metodą dokładną [1]. Wykorzystanie rachunku sekwentów w logice matematycznej umożliwia w łatwy sposób dowodzenie twierdzeń i praw, a w technice cyfrowej może wspomagać projektowanie układów cyfrowych [1, 6].

Prace dotyczące wykorzystania logiki Gentzena w projektowaniu układów cyfrowych zostały zapoczątkowane przez Mariana Adamskiego i podsumowane monografią [1]. Jednym z proponowanych tam sposobów projektowania układów cyfrowych jest zastosowanie języka logiki matematycznej do specyfikacji układu i zastąpienie przekształceń tablicowych przekształceniami symbolicznymi związanymi z formalnym wnioskowaniem. Praktyczne wykorzystanie przekształceń Gentzena może być ilustrowane na przykładzie projektu nietypowego multipleksera o dwóch wejściach adresowych i trzech czterobitowych wejściach danych. Jeśli na wejściach adresowych występują sygnały  $S0=0$ ,  $S1=0$  to na wyjściach multipleksera występują sygnały  $z3=0$ ,  $z2=0$ ,  $z1=0$  i  $z0=0$ , niezależnie od stanu sygnałów wejściowych. Przykład projektu został zaczerpnięty z książki [2]. Deklaracje zmiennych boolowskich, wejściowych i wyjściowych zawiera tab.1. Schemat blokowy multipleksera przedstawiono na (rys. 2).

Tab. 1. Deklaracje zmiennych

| W e j ś c i a      | W y j ś c i a      |
|--------------------|--------------------|
| A=[a3, a2, a1, a0] | Z=[z3, z2, z1, z0] |
| B=[b3, b2, b1, b0] |                    |
| L=[l3, l2, l1, l0] |                    |



Rys. 2. Schemat blokowy multipleksera

Mikrooperacje zapisane w postaci ogólnej zawarto w tab.2.

Tab. 2. Postać ogólna mikrooperacji

|                        |                            |
|------------------------|----------------------------|
| $/s0*/s1 \mid - [Z=0]$ | { Wyjście Z jest równe 0 } |
| $/s0*/s1 \mid - [Z=A]$ | { Wyjście Z jest równe A } |
| $s0*/s1 \mid - [Z=B]$  | { Wyjście Z jest równe B } |
| $s0*/s1 \mid - [Z=L]$  | { Wyjście Z jest równe L } |

Specyfikacja multiplexera w zapisie regułowym w postaci sekwentów przedstawiona została w tab. 3.

Tab. 3. Specyfikacja multiplexera

|                                                           |
|-----------------------------------------------------------|
| $/s0*/s1 \mid - /z0*/z1*/z2*/z3;$                         |
| $/s0*/s1 \mid - (z0<->a0)*(z1<->a1)*(z2<->a2)*(z3<->a3);$ |
| $s0*/s1 \mid - (z0<->b0)*(z1<->b1)*(z2<->b2)*(z3<->b3);$  |
| $s0*/s1 \mid - (z0<->l0)*(z1<->l1)*(z2<->l2)*(z3<->l3);$  |

Normalizacja sekwentów polega na usunięciu z nich wszystkich spójników logicznych i ich zapisaniu w postaci zbliżonej do klauzul, zawierających wyłącznie formuły atomowe. Przykładowo drugi z sekwentów, po normalizacji ma postać zamieszczoną w tab. 4.

Tab. 4. Przykłady sekwentów znormalizowanych

|                       |                       |
|-----------------------|-----------------------|
| $s1,a0 \mid - z0,s0;$ | $s1,z0 \mid - a0,s0;$ |
| $s1,a1 \mid - z1,s0;$ | $s1,z1 \mid - a1,s0;$ |
| $s1,a2 \mid - z2,s0;$ | $s1,z2 \mid - a2,s0;$ |
| $s1,a3 \mid - z3,s0;$ | $s1,z3 \mid - a3,s0;$ |

Po normalizacji sekwentów i określeniu rodzaju każdej zmiennej (czy jest ona typu wejściowego czy wyjściowego) można podać wyrażenia, opisujące wyjścia multiplexera, zarówno dla stanu wyjścia 0, jak i dla stanu 1, w postaci uporządkowanej sumy iloczynów:

Tab. 5. Wyrażenia znormalizowane w postaci sumy iloczynów

|                                                             |
|-------------------------------------------------------------|
| $/s0*/s1 + s1*/a0*/s0 + s0*/b0*/s1 + s0*s1*/l0 \mid - /z0;$ |
| $/s0*/s1 + s1*/a1*/s0 + s0*/b1*/s1 + s0*s1*/l1 \mid - /z1;$ |
| $/s0*/s1 + s1*/a2*/s0 + s0*/b2*/s1 + s0*s1*/l2 \mid - /z2;$ |
| $/s0*/s1 + s1*/a3*/s0 + s0*/b3*/s1 + s0*s1*/l3 \mid - /z3;$ |
| $s1*a0*/s0 + s0*b0*/s1 + s0*s1*/l0 \mid - z0;$              |
| $s1*a1*/s0 + s0*b1*/s1 + s0*s1*/l1 \mid - z1;$              |
| $s1*a2*/s0 + s0*b2*/s1 + s0*s1*/l2 \mid - z2;$              |
| $s1*a3*/s0 + s0*b3*/s1 + s0*s1*/l3 \mid - z3;$              |

Problem dalszej syntezy i ewentualnej dekompozycji zaimplementowanego układu z wykorzystaniem uniwersyteckiego systemu DEMAİN [3] został rozwiązany poprzez transformację specyfikacji do postaci akceptowalnej przez system berkeleyowski „Espresso”.

System ten jest rozbudowanym, heurystycznym programem minimalizacji funkcji boolowskich, opracowanym w początkach lat osiemdziesiątych na Uniwersytecie Kalifornijskim. Swoją ogromną popularność zawdzięcza wolnej dystrybucji oraz dużej skuteczności procesów minimalizacji funkcji boolowskich. System ten, pierwotnie planowany do optymalizacji struktur PLA w uniwersyteckich systemach projektowania układów Full Custom, obecnie jest także stosowany w komercyjnych kompilatorach układów PLD np. ABEL, LOGIC, CUPL [4].

Innym ciekawym przykładem zastosowania logiki sekwentów Gentzena jest analiza porównawcza funkcji dwóch różnych układów kombinacyjnych. Przy pomocy logiki sekwentów w łatwy sposób można stwierdzić, czy badane układy realizują dokładnie tę samą funkcję. W przypadku, gdy badane funkcje realizowanych układów są różne, proces normalizacji precyzyjnie wyznacza wszystkie różnice. Jako przykład analizie zostaną poddane dwa proste układy realizujące bardzo zbliżone funkcje (1,2).

$$Y1 = X0/X2 + /X0*X2 + X0*X1 \quad (1)$$

$$Y2 = X0*X2 + /X0*X2 + X0*X1 \quad (2)$$

Z dwóch funkcji możemy utworzyć dwa sekwenty (3,4) stawiając pomiędzy nimi znak wynikania logicznego.

$$X0/X2 + /X0*X2 + X0*X1 \vdash X0*X2 + /X0*X2 + X0*X1 \quad (3)$$

$$X0*X2 + /X0*X2 + X0*X1 \vdash X0/X2 + /X0*X2 + X0*X1; \quad (4)$$

Jeżeli z procesu normalizacji dwóch badanych sekwentów, odpowiednio zbudowanych z dwóch funkcji ( $Y1 \vdash Y2$  i  $Y2 \vdash Y1$ ), nie powstanie żaden sekwent znormalizowany, to oba układy realizują tę samą funkcję.

W rozpatrywanym przykładzie wynikiem normalizacji są dwa sekwenty, przedstawione w postaci sumy iloczynów(5,6), które jednoznacznie identyfikują różnicę układów.

$$X0/X2*/X1 \vdash; \quad (5)$$

$$X0*X2*/X1 \vdash; \quad (6)$$

#### 4. ZAKOŃCZENIE

Notacja logiki sekwentów odzwierciedla naturalny tok rozumowania człowieka. Z drugiej strony notacja symboliczna w języku logiki, w odróżnieniu od powszechnie stosowanej notacji algebraicznej, grafowej i tablicowej zachowuje jednorodność na wszystkich etapach projektowania, ułatwiając syntezę hierarchiczną metodą kolejnych uściśleń. Przyjęcie formalnej i precyzyjnej notacji uwalnia umysł człowieka od pamiętania zbędnych szczegółów. Względna niezależność poszczególnych sekwentów odzwierciedla naturalny

sposób podziału skomplikowanego zagadnienia na możliwe proste i zamknięte podzagadnienia.

W ramach badań o charakterze eksperymentalnym wykonano oprogramowanie pozwalające na efektywną minimalizację behawioralnego opisu układu kombinacyjnego z wykorzystaniem metody wnioskowania rachunku sekwentów Gentzena. Oprogramowanie częściowo spełnia podobną rolę, jak analogiczny, znany na świecie program minimalizacji wyrażeń logicznych, bazujący na metodzie wymnażania logicznego Mathonego [8]. Program może być przydatny w różnorodnych algorytmach optymalizacji lub weryfikacji układów cyfrowych. Oprogramowanie o charakterze akademickim, zrealizowane w języku obiektowym (Borland C++) dodatkowo przygotowuje dane wyjściowe w postaci akceptowanej przez Espresso i inne znane na świecie systemy uniwersyteckie.

#### LITERATURA

- [1] M. Adamski: *Projektowanie układów cyfrowych systematyczną metodą strukturalną*, Wydawnictwo Wyższej Szkoły Inżynierskiej w Zielonej Górze, Zielona Góra 1990
- [2] *Logika Formalna. Zarys encyklopedyczny z zastosowaniem do informatyki i lingwistyki*, Praca zbiorowa pod red. W. Marciszewskiego. PWN, Warszawa 1987
- [3] T. Łuba, M.A. Markowski, B. Zbierchowski: *Komputerowe projektowanie układów cyfrowych w strukturach PLD*, Wydawnictwa Komunikacji i Łączności, Warszawa 1993
- [4] T. Łuba, K. Jasiński, B. Zbierchowski: *Specjalizowane układy cyfrowe w strukturach PLD i FPGA*, Wydawnictwa Komunikacji i Łączności, Warszawa 1997
- [5] Z. Pawlak: *Automatyczne dowodzenie twierdzeń*, Państwowe Zakłady Wydawnictw Szkolnych, Warszawa 1965
- [6] J. Szajna: *Projektowanie układów cyfrowych z wykorzystaniem programowania logicznego*, Wydawnictwo Politechniki Zielonogórskiej, Zielona Góra 1996
- [7] *Układy przełączające w automatyce przemysłowej. Zadania.*, Praca zbiorowa pod kierunkiem H. Małysza, WNT, Warszawa 1981
- [8] H.J. Mathony: *Universal logic design algorithm and its application the synthesis of two-level switching circuits*, IEE Proceedings Letters, Elsevier Science Publishers (North Holland), Vol.29,1990,pp. 195-210