

Kompakcja wejściowego strumienia bitowego z użyciem metod zbiorów przybliżonych do symulacji mocy w układach sekwencyjnych

*Tomasz GRZEŚ**

1. Wprowadzenie

Minimalizacja mocy oraz projektowanie układów o obniżonym poborze energii jest jednym z wiodących trendów we współczesnej technice cyfrowej. Minimalizacja zużycia energii prowadzi m.in. do wydłużenia czasu między cyklami ładowania baterii w systemach mobilnych. Służy również do projektowania urządzeń przyjaznych środowisku. Projektowanie układów o obniżonym poborze mocy i projektowanie zorientowane na moc w jednym z etapów wymagają obliczenia mocy pobieranej przez system. Jedną z możliwości jest symulacja układu w celu wyznaczenia poboru mocy, ale proces ten może być bardzo czasochłonny, ponieważ wymaga długiego wejściowego strumienia bitowego, który jest przekazywany do wejść symulowanego obwodu. Aby skrócić czas symulacji strumień można poddać procesowi kompaktacji, tzn. zmniejszenia długości.

Teoria zbiorów przybliżonych została opracowana przez prof. Zdzisława Pawłaka w latach 80. XX wieku. Zbiory przybliżone są używane do analizy i przetwarzania danych, a jednym z pól ich wykorzystania jest redukcja danych. Usunięcia nadmiarowości w wejściowym strumieniu bitowym można dokonać wykorzystując metody zbiorów przybliżonych; z pomocą mogą przyjść także redukcja (redukt) i dyskretyzacja [15].

W przypadku opisanym w niniejszym opracowaniu do kompaktacji wejściowego strumienia bitowego wykorzystywany jest redukt. Proponowany algorytm potrzebuje dwóch informacji: wejściowego strumienia bitowego i listy przejść układu sekwencyjnego. Strumień danych jest traktowany jako tabela decyzyjna rozszerzona o dane wyodrębnione z listy przejść w celu utworzenia atrybutu decyzyjnego. Tak przygotowana tablica jest redukowana, a następnie poddawana dyskretyzacji, co pozwala na zmniejszenie zarówno ilości bitów, jak i ilości słów w strumieniu.

Wcześniejsze prace związane z kompaktacją wejściowego strumienia bitowego wykorzystywały metody wymagające skomplikowanych obliczeń arytmetycznych (np. transformata Fouriera), przez co czas ich wykoania znacznie się wydłużał. Zastosowanie

* Politechnika Białostocka

metod zbiorów przybliżonych pozwala na realizację algorytmu, wykorzystując prawie wyłącznie operacje boolowskie, które można np. zrealizować w układzie FPGA [5]. Prowadzi to do znaczącego przyspieszenia operacji.

Niniejsze opracowanie jest zorganizowane następująco: w punkcie drugim opisane zostały zagadnienia związane z mocą w układach cyfrowych, trzeci prezentuje techniki związane z wyznaczaniem mocy w układach sekwencyjnych oraz opisuje metody kompaktacji strumienia wejściowego, czwarty przedstawia zagadnienia związane ze zbiorami przybliżonymi, w piątym został opisany proponowany algorytm oraz wyniki jego badań, natomiast w szóstym następuje podsumowanie.

2. Moc w układach cyfrowych

Pobór mocy przez układy CMOS jest spowodowany następującymi czynnikami:

- prądem upływności zależnym od fizycznych parametrów układu, takich jak technologia i sposób wykonania, wymiary elementów, jakość materiałów itp.;
- przepływem prądu związanego z przełączaniem (ang. *crowbar current*), gdy prąd płynie podczas przełączania z VDD do masy;
- przepływem prądu, który powoduje ładowanie lub rozładowanie pojemności wyjściowych każdego z elementów [11].

Należy jednak zaznaczyć, że większość mocy w układach CMOS jest wydzielana głównie na pojemnościach wyjściowych.

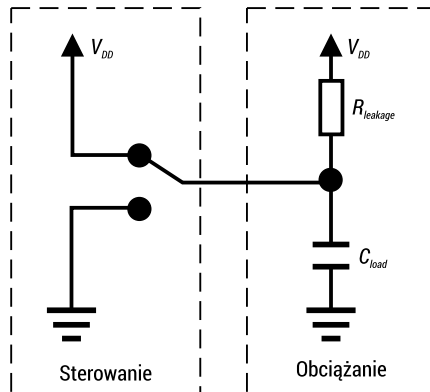
Przed rozpoczęciem obliczania mocy układów logicznych CMOS należy założyć, że [17]:

- moc jest tracona wyłącznie w pojemnościach znajdujących się na wyjściach elementów układu;
- prąd płynie albo od V_{DD} do pojemności wyjściowej albo od pojemności wyjściowej do masy;
- napięcie na wyjściu elementu logicznego (bramki, przerzutnika) może się zmieniać jedynie z V_{DD} na 0 lub z 0 na V_{DD} .

W związku z tym moc P_a pobieraną przez element układu cyfrowego (logicznego) a można wyliczyć korzystając z równania [17]:

$$P_a = \frac{1}{2} \cdot V_{DD}^2 \cdot f_{max} \cdot N_a \cdot C_a \quad (1)$$

gdzie: V_{DD} – napięcie zasilające układ; f_{max} – maksymalna częstotliwość pracy układu (maksymalna częstotliwość, względem której wyliczane są parametry sygnałów wejściowych i wyjściowych); C_a – pojemność wyjściowa elementu a ; N_a – aktywność przełączania (średnia liczba zmian stanu wyjścia w czasie trwania jednego cyklu zegara) elementu a .



RYS. 1. Uproszczony schemat inwertera CMOS (na podstawie [12])
 FIG. 1. Simplified schematic of the CMOS inverter (based on [12]).

ŹRÓDŁO: [12]
 SOURCE: [12].

CMOS jest technologią najczęściej wykorzystywaną w produkcji układów cyfrowych. W układach CMOS moc jest wydzielana głównie na pojemnościach wyjściowych (C_{load}) oraz na rezystancjach odpowiadających za upływność ($R_{leakage}$). Najczęściej korzysta się z modelu inwertera przedstawionego na rysunku 1. W modelu układ sterowania dołącza do obciążenia (wyjścia) bramki napięcie zasilania V_{DD} lub masę – w zależności od wartości, która ma zostać osiągnięta na wyjściu. Każdorazowa zmiana stanu wyjścia powoduje przepływ prądu związany z przeładowaniem (ładowaniem lub rozładowaniem) pojemności obciążenia C_{load} , a co za tym idzie – wydzielanie mocy.

Równanie (1) pozwala określić moc pobieraną przez element układu. Wszystkie parametry występujące po jego prawej stronie, za wyjątkiem N_a , są wartościami stałymi, zależnymi od technologii wykonania układu scalonego. W związku z tym problem obliczania mocy można zredukować do problemu obliczania aktywności przełączania układu.

3. Metody obliczania mocy w układach cyfrowych

Obliczanie mocy w układach cyfrowych jest jednym z ważniejszych procesów w projektowaniu systemów cyfrowych. Jest niezbędne do przeprowadzenia minimalizacji poboru mocy, jak również do oceny finalnego rozwiązania pod kątem poboru energii. Technicznie wyróżnia się dwie grupy metod obliczania mocy: metody statyczne oraz metody dynamiczne [3].

Styczne metody obliczania mocy opierają się na własnościach statystycznych (probabilistycznych) badanego układu. Do obliczeń wymagana jest znajomość takich parametrów sygnału wejściowego, jak prawdopodobieństwo pojawienia się jedynki na linii wejściowej czy współczynniki korelacji pomiędzy sygnałami dla poszczególnych linii wejściowych [4].

W przypadku układów kombinacyjnych szacowanie mocy polega na wyliczeniu prawdopodobieństwa zmiany stanu dla wyjść wszystkich bramek w układzie. Aby wyliczyć prawdopodobieństwo zmiany stanu należy wyliczyć prawdopodobieństwo, że przedstawiona poniżej funkcja f przyjmie wartość 1 [2]:

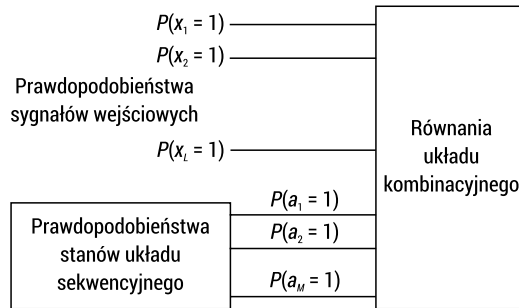
$$f = y_{t-1} \cdot \overline{y_t} + \overline{y_{t-1}} \cdot y_t \quad (2)$$

gdzie: y_t – stan wyjścia w czasie t ; y_{t-1} – stan wyjścia w czasie $t - 1$. Gdy funkcja f przyjmie wartość 1, wtedy nastąpi zmiana wartości na wyjściu y .

W przypadku układów sekwencyjnych stosuje się modelowanie za pomocą dyskretnych łańcuchów Markowa $\{X^t \mid t \in T\}$ ze skończoną liczbą stanów $A = \{a_1, \dots, a_M\}$ przy założeniu, że przestrzeń czasu T jest dyskretna [16]. Do obliczenia prawdopodobieństw statycznych wykorzystuje się natomiast równania Chapmana-Kołmogorowa.

Wyliczone prawdopodobieństwa statyczne pozwalają na obliczenie mocy wydzielonej w pamięci układu sekwencyjnego (czyli w przerzutnikach). Obliczenie mocy całkowitej wymaga dodatkowo obliczenia mocy wydzielonej w części kombinacyjnej układu.

Na rysunku 2 przedstawiono graficznie sposób obliczania mocy w części kombinacyjnej układu sekwencyjnego. Jako prawdopodobieństwa sygnałów wejściowych układu kombinacyjnego są podawane zarówno wartości dla wejść układu sekwencyjnego, jak i prawdopodobieństwa poszczególnych stanów układu sekwencyjnego. Na tej podstawie wylicza się aktywność przełączania, a zarazem moc wydzieloną w części kombinacyjnej układu.



RYS. 2. Statyczna metoda obliczania mocy w układach sekwencyjnych

FIG. 2. Static method of power calculation in sequential circuits

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Dynamiczne metody obliczania mocy oparte są na symulacji. Układ pobudzany jest stworzonym wcześniej ciągiem wejściowym. Ciąg ten może być budowany na podstawie parametrów statystycznych, bądź wynikać z przeznaczenia układu, np. w przypadku układu kodującego można podać ciąg bajtów, który ma zostać zakodowany.

Symulacja może być przeprowadzona na dwa sposoby. W pierwszym przypadku na wejście układu podawany jest cały wygenerowany ciąg wejściowy. Wymaga to przeprowadzenia pełnego cyklu obliczeń. Można również przeprowadzić symulację dla niepełnego ciągu – wówczas obliczenia są przerywane, gdy średnia moc zmienia się tylko w zadanym zakresie (następuje konwergencja).

Do przeprowadzenia symulacji układu potrzebne jest wygenerowanie ciągu wejściowego. Ciąg może być tworzony na podstawie danych statystycznych opisujących prawdopodobieństwa pojawienia się sygnałów wejściowych i korelacje pomiędzy nimi. Można również zastosować rzeczywisty ciąg. W celu zminimalizowania czasu symulacji przygotowywany ciąg powinien być jak najkrótszy, a jednocześnie nie powinien znacząco ujemnie wpływać na wynik symulacji (nie powinien powodować powstawania błędów).

W pracy [10] przedstawiono jedną z metod generowania ciągu wejściowego. Zastosowano tu analizę spektralną opartą o dyskretną transformację Fouriera, której poddawany jest wygenerowany ciąg. Wyluczane są także współczynniki widma $X(k)$ ze wzoru:

$$X(k) = \frac{1}{N} \sum_{n=1}^{N-1} x(n) e^{j \frac{2\pi kn}{N}} \quad (3)$$

gdzie: $x(n)$ – ciąg wejściowy; $X(k)$ – współczynniki widma sygnału; N – liczba próbek (długość ciągu), a zarazem liczba współczynników widma.

Z uzyskanego ciągu współczynników wybiera się podzbiór, na podstawie którego generuje się ciąg wynikowy, korzystając z odwrotnej dyskretnej transformaty Fouriera. Poszczególne wartości ciągu $x(n)$ wylucza się ze wzoru:

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi kn}{N}} \quad (4)$$

Powyższa metoda została zbadana zarówno dla układów kombinacyjnych, jak i sekwencyjnych. Uzyskany przez autorów błąd w porównaniu z ciągiem niepoddanym kompaktacji nie przekraczał kilku procent.

Metodę generowania ciągu wejściowego opartą o łańcuchy Markowa opisano w [9]. Ciąg generowany jest na podstawie takich parametrów statystycznych, jak średnie prawdopodobieństwo wejściowe, średnia gęstość przejść oraz przestrzenną korelację wejść. Zaprezentowany algorytm pozwala na bardzo szybkie wygenerowanieżądanego ciągu wejściowego.

Niejednokrotnie wygenerowany lub rzeczywisty ciąg jest bardzo długi, przez co proces symulacji staje się czasochłonny. Istnieją metody pozwalające zmniejszyć długość ciągu nie zmieniając jego własności, a przez to skracające proces symulacji.

W [7] opisano technikę kompaktacji opartą o grupowanie i kolejne próbkowanie. Grupowanie polega na wydzieleniu z ciągu grup par wektorów wejściowych posiadających zbliżone wartości odległości, np. z użyciem charakterystyki ładowania-rozładowania pojemności CDC (ang. *charging discharging capacitance*). Następnie odbywa się próbkowanie, które ma na celu wyeliminowanie powtarzających się fragmentów ciągu.

Technika opisana w [6] bazuje na wartości czułości na moc wejść podstawowych. Pobór mocy zależy nie tylko od aktywności przełączania (częstotliwości zmian sygnałów wejściowych z jedynki na zero lub odwrotnie), ale również od zmienności aktywności przełączania wejść układu. W [1] wprowadzono pojęcie czułości na moc wejść podstawowych $S_{a(x)}$, będącej miarą wpływu zmienności aktywności przełączania wejść na moc, opisaną jako:

$$S_{a(x_i)} = \lim_{\Delta a(x_i) \rightarrow 0} \frac{\Delta Power(x_i)}{\Delta a(x_i)} = \frac{\partial Power(x_i)}{\partial a(x_i)} \quad (5)$$

gdzie: $a(x_i)$ jest aktywnością przełączania wejścia x_i , $\Delta a(x_i)$ to zmienność aktywności przełączania, a $\Delta Power(x_i) = Power[a(x_i) - \Delta a(x_i)] - Power[a(x_i)]$ jest zmiennością pobieranej mocy dla zadanej zmienności aktywności przełączania $\Delta a(x_i)$.

Techniki kompaktacji oparte o fraktale przedstawiono w pracach [13] oraz [14]. Fraktalami nazywa się kształty zbudowane z fragmentów podobnych do jego całości. Ciągi wejściowe również mogą składać się z podobnych do siebie fragmentów, w związku z czym można je porównywać do fraktali.

Wszystkie wymienione metody wymagają przeprowadzania obliczeń arytmetycznych, przez co czas niezbędny do ich realizacji może być dość długi. Alternatywą, która jeszcze nigdy nie była brana pod uwagę przy kompaktacji ciągu wejściowego, jest zastosowanie zbiorów przybliżonych, których metody w znacznej mierze opierają się na operacjach na funkcjach boolowskich. Pozwala to na zastosowanie np. układów FPGA do przeprowadzenia kompaktacji, co znacząco przyspiesza otrzymywanie wyniku.

4. Zbiory przybliżone i redukt

Zbiory przybliżone stanowią uzupełnienie rachunku prawdopodobieństwa i zbiorów rozmytych i służą wraz z nimi do pozyskiwania wiedzy z niepełnych i nieprecyzyjnych zbiorów danych. Teoria ta powstała w latach 80. XX wieku, a jej twórcą był prof. Zdzisław Pawlak [15].

Pojęcie zbioru przybliżonego rozszerza pojęcie klasycznego zbioru o niepewność, przez co doskonale nadaje się m.in. do klasyfikacji w przypadku nieprecyzyjnych i niepełnych danych.

Podstawowymi pojęciami określającymi granice zbioru przybliżonego są dolna ($\underline{B}X$) i górna aproksymacja ($\overline{B}X$) [16]. Wszystkie obiekty x , które na pewno należą do zbioru przybliżonego X , stanowią dolną aproksymację. Dolną aproksymację systemu informacyjnego $SI = \{U, A\}$, gdzie U – zbiór obiektów $\{x_1, \dots, x_N\}$, natomiast A – zbiór atrybutów $\{a_1, \dots, a_M, d\}$ wraz z atrybutem decyzyjnym d , można wyznaczyć, korzystając z klas abstrakcji $IND_{SI,B}(x)$. Klasy abstrakcji określane są za pomocą poniższego równania:

$$IND_{SI,B}(x) = \{y \in U : (x, y) \in IND_{SI}(B)\} \quad (6)$$

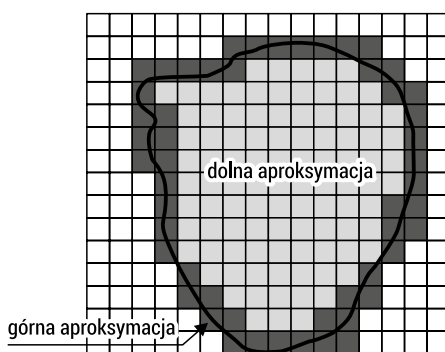
gdzie: $IND_{SI,B}(x)$ – klasa nierozróżnialności względem atrybutów ze zbioru B ;
 $IND_{SI}(B) = \{(x, y) : \forall_{a \in B} d(x) = d(y)\}$ – relacja nierozróżnialności względem atrybutów ze zbioru B [16].

W takim przypadku dolna aproksymacja jest określona wzorem:

$$\underline{B}X = \{x \in U : IND_{SI,B}(x) \subseteq X\} \quad (7)$$

Wszystkie obiekty x_i , które prawdopodobnie należą do zbioru przybliżonego X , stanowią górną aproksymację [15]. Można ją również wyliczyć, korzystając z klas nierozróżnialności za pomocą zależności:

$$\overline{B}X = \{x \in U : IND_{SI,B}(x) \cap X \neq \emptyset\} \quad (8)$$



RYS. 3. Górna i dolna aproksymacja zbioru przybliżonego

FIG. 4. Lower and upper approximation of a rough set

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Na rysunku 3 przedstawiono schematycznie dolną i górną aproksymację zbioru przybliżonego. Dolną aproksymację stanowi jasnoszary obszar wewnątrz obramowania oznaczającego zbiór przybliżony. Górną aproksymacją są jasno- i ciemnoszare elementy, przy czym ciemnoszare elementy są nazywane obszarem granicznym.

Nie wszystkie atrybuty ze zbioru A są niezbędne do zachowania pełnej rozróżnialności obiektów ze zbioru X . Dla danego systemu informacyjnego IS może istnieć podzbiór $R \subseteq A$, który pozwala na taki sam podział obiektów na klasy decyzyjne jak pełny zbiór atrybutów A . Podzbiór R nazywany jest reduktem [5].

Dla danego IS może istnieć więcej niż jeden redukt, jak również może nie być żadnego reduktu. Jednocześnie przy więcej niż jednym redukcje może istnieć pewna grupa atrybutów C , która jest fundamentalna i występuje w każdym z nich. Grupa atrybutów C nazywana jest rdzeniem. Żaden redukt nie może nie zawierać wszystkich atrybutów znajdujących się w rdzeniu [5].

Czasami atrybut może przyjmować wartości ciągłe lub dyskretne, ale z bardzo licznego zbioru. Może to powodować pewne utrudnienia przy klasyfikacji oraz niepotrzebnie zwiększać rozmiar tablicy decyzyjnej. W związku z tym często stosuje się dyskretyzację, czyli proces zmniejszania liczności zbioru wartości danego atrybutu [5]. Należy jednak dochować wszelkich starań, aby proces ten miał jak najmniejszy wpływ na podejmowane decyzje.

5. Proponowany algorytm oraz badania

Istnienie dwóch przekształceń tablicy decyzyjnej, tzn. redukcji (wyznaczanie reduktu i usuwanie atrybutów nieznajdujących się w redukcje) oraz dyskretyzacji (zmniejszanie liczności zbioru wartości danego atrybutu w taki sposób, żeby w jak najmniejszym stopniu wpłynąć na decyzję końcową) pozwala wysnuć wniosek, iż można je zastosować do kompaktacji wejściowego strumienia bitowego. Do tego celu należy przygotować tablicę decyzyjną składającą się z atrybutów warunkowych, atrybutu decyzyjnego oraz obiektów (tab. 1).

TAB. 1. Tablica decyzyjna

TAB. 1. Decision table

		Atrybuty						
		a_1	a_2	a_3	a_4	...	a_N	d
Obiekty	x_1							
	x_2							
	...							
	x_M							

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Tablica decyzyjna z tabeli 1 składa się N atrybutów warunkowych, jednego decyzyjnego oraz M obiektów. W przypadku strumienia bitowego można przyjąć, że atrybutem jest każdy z bitów poszczególnych słów wejściowych. Dodatkowo obiektami będą słowa strumienia bitowego. Do przeprowadzenia redukcji oraz dyskretyzacji trzeba jeszcze wprowadzić do tablicy decyzyjnej atrybut decyzyjny. W normalnych warunkach atrybut decyzyjny określa decyzję podjętą na podstawie wartości atrybutów warunkowych. W przypadku ciągu bitowego decyzja będzie polegała

na określeniu, do którego stanu nastąpi przejście w momencie pojawienia się na wejściu układu słowa ze strumienia wejściowego. Do realizacji tego zadania konieczna jest znajomość architektury układu sekwencyjnego, dla którego przeprowadzana jest symulacja. Jednym z najczęściej spotykanych opisów układu sekwencyjnego jest lista przejść. W liście przejść (tab. 2) każdy element listy składa się z czterech wartości:

- stanu bieżącego a_i – stanu układu przed przejściem;
- słowa wejściowego $x(a_i \rightarrow a_i')$, które powoduje przejście ze stanu a_i do stanu następnego;
- słowa wyjściowego $y(a_i')$, które pojawi się na wyjściu przy przejściu do stanu następnego;
- stanu następnego a_i' , do którego przejdzie układ po pojawieniu się na wejściu słowa $x(a_i \rightarrow a_i')$.

TAB. 2. Lista przejść układu sekwencyjnego

TAB. 2. Transition list of a sequential circuit

Stan bieżący	Słowo wejściowe	Słowo wyjściowe	Stan następny
a_1	$x(a_1 \rightarrow a_1')$	$y(a_1')$	a_1'
a_2	$x(a_1 \rightarrow a_1')$	$y(a_1')$	a_1'
...
al.	$x(a_l \rightarrow a_l')$	$y(a_l')$	a_l'

ŹRÓDŁO: opracowanie własne.

SOURCE: own elaboration.

Biorąc pod uwagę kolumnę drugą i czwartą z listy przejść z tabeli 2, można łatwo uzupełnić tablicę decyzyjną o atrybut decyzyjny. Gdy tablica decyzyjna zostanie przygotowana można przystąpić do jej redukcji. Do generowania reduktu można zastosować algorytm opisany w [5], gdzie bazuje się na częstości wystąpień danego atrybutu w tzw. macierzy rozróżnialności. Natomiast dyskretyzację można zrealizować za pomocą algorytmu opisanego w [8], gdzie stosuje się tzw. dyskretyzację z maksymalną rozróżnialnością. Dyskretyzację należy wykonać, zakładając, że atrybutem warunkowym jest całe słowo, a nie jak w przypadku generowania reduktu poszczególne bity. Jest to spowodowane tym, że nie da się dokonać dyskretyzacji atrybutu, który może przyjmować tylko dwie wartości.

Poniżej przedstawiono opis proponowanego algorytmu kompaktacji strumienia wejściowego.

Algorytm kompaktacji strumienia wejściowego

dane: wejściowy strumień bitowy $S = \{s_i\}$, lista przejść $T = \{< a_i, x(a_i \rightarrow a_i'), y(a_i'), a_i' >\}$

wyjście: skompaktowany strumień S'

1. Przygotuj tablicę decyzyjną D składającą się z $N+1$ kolumn, gdzie N jest liczbą wejść układu sekwencyjnego, oraz M wierszy, gdzie M jest liczbą słów w strumieniu wejściowym S .
2. Dla każdego wiersza k tablicy decyzyjnej D :
 - a) w kolumnie $N+1$ umieść wartość a_i' z listy przejść T , dla której słowo s_k strumienia S ma wartość identyczną z $x(a_i \rightarrow a_i')$.
3. Wykonaj procedurę generowania reduktu R dla tablicy decyzyjnej D zgodnie z poniższą procedurą (na bazie [5]):
 - a) stwórz kopię tablicy decyzyjnej D , na której wykonasz poniższe operacje;
 - b) dla każdej pary obiektów należących do różnych klas decyzyjnych dla każdego atrybutu a_i oblicz liczbę wystąpień różnicy wartości atrybutu między elementami pary;
 - c) wybierz atrybut a_{max} z największą liczbą wystąpień i dodaj go do reduktu;
 - d) usuń wszystkie obiekty, które należały do par wskazujących na konieczność wybrania atrybutu a_{max} ;
 - e) jeżeli w tablicy pozostają jakieś obiekty, to wróć do punktu b).
4. Scal wszystkie bity (atrybuty warunkowe) w tablicy decyzyjnej w jeden atrybut warunkowy.
5. Wykonaj procedurę dyskretyzacji zgodnie z poniższym opisem (na bazie umieszczonego w [8]):
 - a) posortuj wartości atrybutów w porządku rosnącym;
 - b) podziel atrybuty na dwie części w taki sposób, aby liczba obiektów należących do różnych klas decyzyjnych w każdej z części była maskymalna;
 - c) dla każdej z części wykonaj ponowny podział zgodnie z punktem b);
 - d) przerwanie pęti następuje w momencie, gdy liczba obiektów należących do różnych klas decyzyjnych w każdej z części osiągnie założoną wartość minimalną.
6. W każdej z części pozostaw po jednym (pierwszym) obiekcie.

Badania algorytmu zostały przeprowadzone dla losowego strumienia wygenerowanego przez funkcję `rand()` z biblioteki `glibc` (środowisko MinGW dołączone do IDE Code::Blocks w wersji 17.12). Strumień składał się z 1024 słów o rozmiarze 8 bitów. Strumień wejściowy został poddany kompaktacji dla układów sekwencyjnych ze zbioru [18]. Wyniki badań przedstawiono w tabeli 1. W kolumnie „Benchmark” umieszczono nazwę układu, kolumna „Długość” zawiera długość strumienia po procesie kompaktacji, natomiast kolumna „% redukcji” opisuje o ile procent zmniejszył się strumień po procesie kompaktacji.

TAB. 3. Lista przejść układu sekwencyjnego
 TAB. 3. Transition list of a sequential circuit

Benchmark	Długość	% redukcji
bbara	936	8,6
bbsse	818	20,1
bbtas	832	18,8
beecount	835	18,5
keyb	880	14,1
s27	910	11,1
tav	881	14,0

ŹRÓDŁO: opracowanie własne.
 SOURCE: own elaboration.

Wyniki przedstawione w tabeli 3 potwierdzają, iż proponowany algorytm można wykorzystać do kompaktacji ciągu. Największą redukcję uzyskano dla układu „bbsse”, natomiast najgorsze wyniki – dla układu „bbara”. Średnia wartość redukcji wyniosła 15%.

6. Podsumowanie i wnioski

Przedstawione rozwiązanie stanowi przykład nowego podejścia do procesu kompaktacji oraz wskazuje możliwość zastosowania metod zbiorów przybliżonych do zmniejszania długości wejściowych strumieni bitowych.

Badania eksperymentalne potwierdziły, że możliwa jest redukcja rozmiaru strumienia bitowego z wykorzystaniem algorytmu bazującego na teorii zbiorów przybliżonych, a stopień redukcji, wynoszący 15%, pozwala mieć nadzieję, iż po dokonaniu usprawnień algorytm może mieć praktyczne zastosowanie w symulacji poboru mocy układów sekwencyjnych.

Przyszłe prace powinny skupić się na sprawdzeniu możliwości innego podziału bitów ciągu na atrybuty (w tym wielobitowe), innego sposobu przypisania atrybutów decyzyjnych bądź zastosowania deyskretyzacji wielowymiarowej (dla wielu atrybutów).

Literatura

1. Chen Z, Roy K, Chou TL. *Power Sensitivity – a New Method to Estimate Power Dissipation Considering Uncertain Specifications of Primary Inputs*. Proceedings of the 1997 IEEE/ACM International Conference on Computer Aided Design, San Jose, CA, 1997; 40-44.
2. Ghosh A, Devedas S, Kreutzer K, White J. *Estimation of Average Switching Activity in Combinational and Sequential Circuits*. Proc. of 29th ACM/IEEE Design and Automation Conference, 8-12 June 1992; 253-259.
3. Grześ T, Salauyou V. Metody obliczania mocy w układach cyfrowych. *Pomiary, Automatyka, Kontrola*. 2006; 7bis: 101-102.
4. Grześ T. *Sequential Circuits Power Modeling for Low Power Design*. Proceedings of XVI Ukrainian-Polish Conference „CAD in Machinery Design. Implementation and Educational Problems” (CADMD’2006), Polyana, Ukraine, May 22-23, 2006; 54-56.
5. Grześ T, Kopczyński M, Stepaniuk J. FPGA in rough set based core and reduct computation. *Lecture Notes in Artificial Intelligence*. 8171: 263-270.
6. Hsu C-Y, Wei C-W, Shen W-Z. *A Pattern Compaction Technique for Power Estimation Based On Power Sensitivity Information*. Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Sydney, Australia, May 2001, vol. 5; 467-470.
7. Hsu C-Y, Shen W-Z. *Vector Compaction for Power Estimation with Grouping and Consecutive Sampling Techniques*. IEEE International Symposium on Circuits and Systems, Scottsdale, Arizona (USA), May 2002; II-472 – II-475.
8. Kopczyński M, Grześ T, Stepaniuk J. Maximal Discernibility Discretization of Attributes – a FPGA Approach. *Studies in Big Data*. 2016; 19: 171180.
9. Liu X, Papaefthymiou MC. A Markov Chain Sequence Generator for Power Macromodeling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2004; vol. 23, no. 7: 1048-1062.
10. Macii A, Macii E, Poncino M, Scarsi R. Stream Synthesis for Efficient Power Simulation Based on Spectral Transforms. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2001; vol. 9, no. 3: 417-426.
11. Pedram M. Design technologies for Low Power VLSI. *Encyclopedia of Computer Science and Technology*. 1995; 36.
12. Pedram M. Power simulation and estimation in VLSI circuits. In: Chen W-K, ed. *The VLSI Handbook*. The CRC Press and the IEEE Press; 1999.
13. Radjassamy R, Carothers JD. *A Fractal Compaction Algorithm for Efficient Power Estimation*. Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors ICCD ‘98, 5-7 Oct 1998, Austin, TX, USA; 542-547.

14. Radjassamy R, Carothers D. Faster Power Estimation of CMOS Designs Using Vector Compaction – A Fractal Approach. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*. 2003; vol. 33, no. 3: 476-488.
15. Stepaniuk J. Knowledge discovery by application of rough set models. In: Polkowski L, Tsumoto S, Lin TY, eds *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*. Heidelberg: Physica-Verlag; 2000;137-233.
16. Stepaniuk J. *Rough-Granular Computing in Knowledge Discovery and Data Mining*. Springer; 2008.
17. Tsui C-Y, Monteiro J, Pedram M, Devadas S, Despain AM, Lin B. Power Estimation Methods for Sequential Logic Circuits. *IEEE Transactions on VLSI Systems*. 1995; vol. 3, no. 3: 404-416.
18. Yang S. Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0. „Technical Report”, Microelectronics Center of North Carolina, 1991; 43.

Badania zostały zrealizowane w ramach pracy nr S/WI/1/2018 i sfinansowane ze środków na naukę Ministerstwa Nauki i Szkolnictwa Wyższego.

Streszczenie

Minimalizacja mocy i projektowanie układów o obniżonym poborze mocy to bardzo ważne trendy w projektowaniu współczesnych układów cyfrowych. Projektowanie takie wymaga obliczenia mocy pobieranej przez system, co można uzyskać za pomocą symulacji. Proces ten może być bardzo długi, ponieważ wymaga wejściowego strumienia bitowego.

Kompresja bitowego strumienia danych jest jedną z technik wykorzystywanych do usprawniania symulacji poboru mocy układów sekwencyjnych. Do tego celu nigdy nie były stosowane metody zbiorów przybliżonych. Niniejsze opracowanie jest pierwszym podejściem do kompaktacji danych przy użyciu zbiorów przybliżonych.

Słowa kluczowe: układy sekwencyjne, symulacja mocy, kompaktacja, zbiory przybliżone

Summary

Bitstream compaction using rough set theory for power simulation in sequential circuits

Power minimization and low power design are very important trends in digital circuits design. Minimizing power consumption, for example, leads to increasing the time between recharging the battery in mobile systems. It is also needed to design environmentally friendly devices. Low-power designing and power-aware designing, in one of the stages, need to calculate the power consumed by the system. This can be obtained by power simulation, but this process can be very long, as it needs the stream of data that is given to inputs of the simulated circuit. The stream may be compacted to cut the simulation time. The rough sets theory was developed by Prof. Z. Pawlak

in eighties of the 20th century. Rough sets are used for data analysis and processing, and one of the fields of its utilization is a data reduction. Removing the redundancy of the data stream in the rough sets can be achieved with the reduct and the discretization.

In this study, the reduct is used for data stream compaction. Proposed algorithm need two informations: the data stream and the transition list of the sequential circuit. Data stream is treated as the decision table, extended by the data extracted from the transition list to form the decision attribute.

In proposed solution data stream is treated as the decision table – any bit of the data stream represent one conditional attribute. To complete the decision table there is decision attribute needed. Decision attribute is generated from the transition list of the sequential circuit. Full decision table is then processed to obtain the reduct and to perform the discretization. These two processes provide data reduction and the resulting data stream is significantly smaller, that lead to decrease the power simulation time.

Data stream compaction is one of the techniques used in improving the power simulation of the sequential circuits, but never used rough sets methods. This work is first approach to data compaction/compression using rough sets methods.

Keywords: sequential circuits, power simulation, compaction, rough sets