

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I ELEKTRONIKI

ROZPRAWA DOKTORSKA

MGR INŻ. IRENEUSZ BRZOZOWSKI

**PROBLEM MINIMALIZACJI STRAT ENERGII
W CYFROWYCH UKŁADACH VLSI**

Promotor:
Prof. dr hab. inż. Andrzej Kos

Kraków, 2006

Chciałbym w tym miejscu wyrazić słowa wdzięczności dla Prof. Andrzeja Kosa za inspirację, nieocenioną pomoc oraz krytyczne uwagi w trakcie pracy nad rozprawą.

Dzięki Jego wsparciu i wysiłkom organizacyjnym miałem zapewnione bardzo dobre warunki pracy i prowadzenia badań. Miałem także możliwość uczestniczenia i prezentacji pośrednich wyników swoich prac na kilkunastu konferencjach krajowych i zagranicznych.

Spis treści

SPIS TREŚCI	1
WYKAZ WAŻNIEJSZYCH SKRÓTÓW I OZNACZEŃ	3
WPROWADZENIE	4
<i>Geneza pracy</i>	<i>4</i>
<i>Cel i zakres pracy.....</i>	<i>7</i>
<i>Teza</i>	<i>7</i>
<i>Układ pracy</i>	<i>7</i>
 <i>Rozdział 1</i>	
MODELOWANIE STRAT ENERGII W CYFROWYCH UKŁADACH CMOS.....	9
1.1. Modelowanie tradycyjne	10
1.1.1. Aktywność układu logicznego	11
1.1.2. Pojemność węzłowa	16
1.2. Nowy model strat energii w cyfrowych układach CMOS.....	18
1.2.1. Motywacja.....	19
1.2.2. Definicja pojemności ekwiwalentnej	22
1.2.3. Definicja sposobu sterowania bramką.....	25
1.2.4. Model statycznej bramki CMOS.....	27
1.2.5. Model sieci logicznej.....	27
1.3. Ocena własności energetycznych statycznych bramek CMOS.....	29
1.3.1. Parametry zależne od sposobu sterowania bramki – nowy model.....	30
1.3.2. Wyniki oceny parametrów energetycznych bramek dla nowego modelu	34
1.3.3. Parametry niezależne od sposobu sterowania bramki – model tradycyjny	42
 <i>Rozdział 2</i>	
ESTYMACJA STRAT ENERGII W UKŁADACH KOMBINACYJNYCH.....	45
2.1. Zastosowanie nowego modelu w estymacji mocy na przykładzie układu C17.....	47
2.2. Wyniki estymacji mocy z zastosowaniem modelu tradycyjnego i nowego	50

*Rozdział 3***OPTIMALIZACJA UKŁADÓW CYFROWYCH Z ZASTOSOWANIEM****NOWEGO MODELU STRAT ENERGII..... 53****3.1. Procedura optymalizacji55****3.2. Wyniki optymalizacji układów kombinacyjnych.....60***Rozdział 4***REDUKCJA STRAT ENERGII NA ETAPIE SYNTEZY LOGICZNEJ..... 65****4.1. Metody redukcji poboru mocy w dwupoziomowych układach logicznych66****4.2. Synteza logiczna dwupoziomowych układów o obniżonym poborze mocy
z wykorzystaniem nowego modelu69**

4.2.1. Praktyczny algorytm syntezy71

4.2.2. Obliczanie prawdopodobieństwa sposobów sterowania bramek w układzie80

4.2.3. Realizowalność rozkładu prawdopodobieństwa zmian wektorów wejściowych87

4.2.4. Wyniki syntezy układów dwupoziomowych87

**4.3. Redukcja poboru mocy w układach dwupoziomowych przez przekształcenie
do układów trójpoziomowych92**

4.3.1. Redukcja liczby przełączeń – zmniejszanie aktywności układu92

4.3.2. Usuwanie inwerterów wejściowych w kontekście nowego modelu strat energii103

*Rozdział 5***PODSUMOWANIE 110****ZAŁĄCZNIKI..... 113****Załącznik A – Kody źródłowe programów113****Załącznik B – Parametry energetyczne bramek CMOS124****Załącznik C – Skrypty programu SIS126****BIBLIOGRAFIA 127**

Wykaz ważniejszych skrótów i oznaczeń

Skrót:

AND	Iloczyn logiczny
CMOS	Complementary Metal-Oxide-Semiconductor
MOS	Metal-Oxide-Semiconductor
NAND	Negacja iloczynu logicznego
NMOS	MOS z przewodnictwem typu n
NOR	Negacja sumy logicznej
NOT	Negacja logiczna
OR	Suma logiczna
PMOS	MOS z przewodnictwem typu p
VLSI	Very Large Scale of Integration
BSIM	Berkeley Short-Channel IGFET Model for MOS Transistors [She87]
PLA	Programmable Logic Array

Symbol:

Jednostka:

A_i	–	liczba przełączeń węzła i	-
a_i	–	aktywność przełączeniowa węzła i	-
C_{equ}	–	pojemność ekwiwalentna	[F]
C_{In}	–	pojemność wejściowa	[F]
C_L	–	pojemność obciążająca	[F]
C_{Lall}	–	całkowita pojemność ekwiwalentna bramki	[F]
dw	–	sposób sterowania bramki	-
f	–	częstotliwość	[Hz]
I_{dd}	–	prąd zasilania	[A]
p	–	prawdopodobieństwo	-
$p(dw)$	–	prawdopodobieństwo sposobu sterowania	-
P_d	–	dynamiczna moc strat	[W]
V_{dd}	–	napięcie zasilania	[V]
V_{ov}	–	napięcie przerzutu	[V]
V_{TN}	–	napięcie progowe tranzystora NMOS	[V]
V_{TP}	–	napięcie progowe tranzystora PMOS	[V]

Wprowadzenie

Geneza pracy

Jeszcze nie tak dawno, niespełna trzy dekady temu, pobór mocy przez układy elektroniczne był dla ówczesnych projektantów systemów cyfrowych problemem drugoplanowym. Rozwiązanie sprowadzało się do zastosowania odpowiedniego zasilacza, aby dostarczyć wymaganej ilości energii. Stosowano wtedy powszechnie układy scalone o małej i średniej skali integracji, co powodowało, że ilość wydzielanego w nich ciepła była mała w stosunku do objętości. Wtedy też pojawiły się pierwsze mikroprocesory i wraz z ich rozwojem następował postęp w zakresie układów scalonych. Coraz bardziej rozbudowane układy cyfrowe wymagały większej miniaturyzacji i gęstości upakowania elementów z jednej strony, a z drugiej, rozwój technologii mikroelektronicznych dawał coraz większe możliwości scalania układów elektronicznych i budowy bardziej zaawansowanych systemów. Sytuacja ta właściwie powtarza się do dzisiaj, działając jak dodatnie sprzężenie zwrotne.

Ciągłe zwiększanie stopnia scalenia i miniaturyzacja układów elektronicznych doprowadziła do tego, że w dzisiejszych układach scalonych występują jedne z największych, o ile nie największe, gęstości mocy na świecie. Profesor G. de Mey w [DeM94] oszacował gęstość mocy na poziomie 10^{15}W/m^3 dla hipotetycznego tranzystora MOS o wymiarach $1\mu\text{m} \times 1\mu\text{m}$, w którym wydzielana się moc $1\mu\text{W}$ ($5\text{V} \cdot 0,2\mu\text{A}$). Trzeba wspomnieć, że dzisiaj na skalę komercyjną produkuje się układy scalone w technologii 65nm, co oznacza, że wymiary tranzystorów są piętnastokrotnie mniejsze niż ten hipotetyczny z [DeM94]. Nie oznacza to jednak ponad dwustukrotnego (15×15) wzrostu gęstości mocy, gdyż wraz ze zmniejszeniem wymiarów charakterystycznych tranzystorów maleją również inne wymiary i parametry układów scalonych. Nazywamy to procesem skalowania układów scalonych. Mimo tego, gęstość jak również pobór mocy, ciągle wzrasta we współczesnych układach scalonych. Główną przyczyną jest wzrost liczby elementów aktywnych – tranzystorów – spowodowany scalaniem na jednej płytce krzemowej coraz bardziej zaawansowanych systemów. W ostatnich latach zaczęto nawet mówić o systemie na krzemie (ang. *System-On-Chip*) scalającym w jednej obudowie elektroniczne układy analogowe, cyfrowe a nawet mikromaszyny, co może świadczyć o obecnym poziomie zaawansowania technologii produkcji układów scalonych.

Tak duże możliwości technologiczne pociągają nieustanny wzrost funkcjonalności i możliwości dzisiejszych urządzeń. Co więcej, oczekiwania użytkowników również ciągle rosną.

W tym aspekcie silny rozwój technologii bezprzewodowych w ostatnich latach można uznać za naturalny. Należy oczekiwać w przyszłości integracji funkcji wielu urządzeń w jednym systemie, dającym dużą uniwersalność, ale i niezależność. Bez wątpienia budowa coraz bardziej skomplikowanych systemów i urządzeń bezprzewodowych wymaga ciągłego rozwoju i opracowywania nowych metod projektowania układów cyfrowych oszczędzających energię. Profesor T. Sakurai twierdzi nawet, że w nadchodzącym, właściwie to już obecnym, wieku wszechobecnej elektroniki układy VLSI „świadome” konsumpcji mocy są bardzo ważne. Jeśli współczesna elektronika nie będzie dostarczać rozwiązań oszczędnych energetycznie, to moc konsumowana przez sprzęt elektroniczny może się stać jednym z największych problemów dla współczesnych społeczeństw [Sak04a], [Sak04b].

Od ponad dwóch dekad rozwijają się metody redukcji poboru mocy układów cyfrowych. Wraz z rozwojem technologii mikroelektronicznych obserwujemy powstawanie coraz bardziej zaawansowanych metod projektowania układów oszczędnych energetycznie. Od kilkunastu lat, kiedy to układy cyfrowe wykonane w technologii CMOS dorównały szybkością innym technologiom (TTL, ECL, itd.), a przewyższały je pod względem oszczędności energii, następuje silny rozwój metod redukcji poboru mocy dedykowanych właśnie dla układów CMOS. W dniu dzisiejszym ta technologia stała się dominującą. Również jej rozwój i budowa zaawansowanych systemów doprowadziły do tego, że już nie wystarcza optymalizacja układu na poziomie bramek, ale konieczna jest na wszystkich etapach projektu. Począwszy od najwyższego poziomu opisu projekt jest tworzony pod kątem redukcji pobieranej mocy. Co więcej, zagadnienie optymalizacji mocy pojawia się również w trakcie tworzenia oprogramowania. Spotykamy zatem metody optymalizacji systemu i architektury. Optymalizacji energetycznej może również podlegać układ opisany na poziomie przesłań rejestrowych, następnie na poziomie bramek i w końcu tranzystorów [Mac98]. Ponadto, dla rozbudowanych systemów obserwuje się rozwój metod zarządzania mocą [Ben00]. Są to metody polegające na takim zarządzaniu zasobami systemu, by wyłączając bądź ograniczając parametry podzespołów, do minimum ograniczyć pobór energii bez widocznej straty wydajności całego systemu. Ma to duże znaczenie w przypadku np. przenośnych komputerów lub innych urządzeń zasilanych bateryjnie.

Redukcja strat energii w systemie wymaga uwzględnienia różnych informacji zależnie od poziomu opisu, którego dotyczy. Obok estymacji, dla której dokładność i prostota modelu strat mocy jest bardzo ważna, modelowanie poboru energii jest niemniej istotne dla procesu optymalizacji. Właściwe określenie funkcji celu optymalizacji decyduje o jej powodzeniu i efektywności. Z drugiej strony uproszczenia i niedokładne modelowanie często pozwalają na

szybka estymację poboru mocy, ale mogą spowodować, że otrzymane wyniki różnią się od rzeczywistości. Podobnie w procesie optymalizacji układu pod kątem poboru mocy niewłaściwy model może spowodować otrzymanie nieoptymalnego rozwiązania. Ze względu na ograniczenia modelowania konsumpcji energii właściwa weryfikacja układu jest możliwa dopiero po jego fizycznym wykonaniu. Jednakże w przypadku układów scalonych koszt fabrykacji jest znacznie większy niż nawet symulacje z bardzo zaawansowanym i kosztownym obliczeniowo modelem. Dlatego może warto ponieść nawet większe nakłady na etapie projektowania układu.

Obserwując na przestrzeni kilkunastu ostatnich lat metody redukcji strat mocy w układach cyfrowych CMOS na poziomie bramek, można odnieść wrażenie, że stosowany jest ten sam model. W przypadku strat dynamicznych jest on zbudowany w oparciu o pojemność węzłową i informację o aktywności sygnałów w układzie. Owszem, w wielu pracach rozważano skomplikowane zależności czasowe oraz przestrzenne między sygnałami, wykorzystywano zaawansowany aparat matematyczny do obliczeń, ale właściwie stosowano taki sam model strat mocy dla bramki logicznej. Rozwinięto metody probabilistyczne obliczania aktywności układu, co znacznie przyspieszyło procesy estymacji i optymalizacji.

W tym punkcie można by postawić tezę, że tradycyjny model poboru mocy wyczerpał już swój potencjał i prawdopodobnie trudno już będzie opracować nowe metody redukcji strat energii w oparciu o niego. Z drugiej strony, dokładna analiza pracy statycznej bramki CMOS oraz pewne przyjęte uproszczenia w modelu tradycyjnym sugerują, że możliwa jest dalsza redukcja pobieranej mocy. Przede wszystkim należy uwzględnić pojemności pasożytnicze występujące w bramce. W związku z ciągłym zmniejszaniem rozmiarów tranzystorów maleją także grubości poszczególnych warstw w układzie scalonym a pasożytnicze pojemności międzywarstwowe nabierają większego znaczenia. Ponadto znaczenie ma fakt, że podczas przełączania bramki lub tylko zmiany stanów na jej wejściach sieć kondensatorów, jaką można zastąpić bramkę, zmienia swoją konfigurację z powodu włączania lub wyłączania się tranzystorów. Bramkę można zatem potraktować jako dynamiczny układ, którego konfiguracja zależy od sygnałów sterujących jej wejścia. Ma to oczywiście wpływ na pobór mocy i wymaga nieco bardziej skomplikowanego modelowania. Potrzebna jest nowa miara aktywności bramki określająca sposób jej sterowania, czyli dająca informację o zmianie wszystkich sygnałów wejściowych. Należy więc mówić o zmianie wektora wejściowego bramki.

Powyżej zasygnalizowane problemy, a zwłaszcza dokładna analiza zachowania się bramki CMOS podczas przełączania, skłoniły autora do podjęcia próby opracowania nowego modelu poboru mocy przez bramkę, pozbawionego dotychczasowych słabości. Bazując na tym modelu

opracowanie nowych lub modyfikacja istniejących metod optymalizacji i syntezy logicznej wydawały się atrakcyjne i celowe. Co więcej, uwzględnienie w modelu pomijanych dotychczas zjawisk pozwalało mieć nadzieję, że istnieje możliwość uzyskania lepszych wyników estymacji i redukcji mocy strat niż metody oparte na modelu tradycyjnym.

Cel i zakres pracy

Za cel pracy autor przyjął sprawdzenie możliwości dalszej redukcji dynamicznych strat energii w układach cyfrowych CMOS wykorzystując zaproponowany nowy model poboru energii uwzględniający pomijane dotychczas zjawiska. Ponadto, adaptacja istniejących i opracowanie nowych metod redukcji strat mocy w układach cyfrowych, będzie potwierdzeniem celowości wprowadzenia nowego modelu. Przez straty mocy będzie rozumiana średnia moc pobrana przez układ cyfrowy w pewnym przedziale czasu.

W związku z powyższym należy opracować metody określania wartości stosownych parametrów energetycznych bramek dla nowego modelu poboru mocy. W celu obliczania nowej miary aktywności układu trzeba opracować odpowiednie algorytmy. Kolejnym punktem jest zastosowanie nowego modelu do estymacji mocy strat układu oraz porównanie otrzymanych wyników z metodami tradycyjnymi. Bazując na zaproponowanym modelu możliwa jest optymalizacja układu przez dobór odpowiedniej kolejności wejść do bramek w układzie. Należy sprawdzić efektywność tej metody. Ponadto, trzeba zbadać przydatność nowego modelu podczas syntezy logicznej i na podstawie wniosków opracować stosowne algorytmy syntezy układów o zredukowanym poborze mocy.

Teza

Istnieje możliwość minimalizacji strat energii w układach logicznych CMOS, jeśli zostanie wykorzystana informacja o prawdopodobieństwie zmian wszystkich sygnałów, traktowanych jako wektor, na wejściach układu i poszczególnych bramek.

Układ pracy

Niniejsza praca została podzielona na pięć rozdziałów, w których opisano modelowanie układów cyfrowych pod kątem strat mocy, estymację i optymalizację mocy oraz możliwości redukcji mocy strat na etapie syntezy logicznej z wykorzystaniem nowego modelu.

W *Rozdziale 1* przedstawiono tradycyjne modelowanie poboru energii przez układy cyfrowe CMOS i zaproponowano, nowy oryginalny model. Wyjaśniono motywację, główne założenia

modelowania oraz metodę wyznaczania parametrów energetycznych bramek dla nowego modelu.

Rozdział 2 zawiera opis zastosowania nowego modelu do estymacji poboru mocy przez układy cyfrowe oraz wyniki estymacji dla zbioru układów testowych i porównanie z wynikami metod tradycyjnych.

W *Rozdziale 3* przedstawiono algorytm optymalizacji układu polegającej na doborze właściwej kolejności wejść do bramek z wykorzystaniem nowego modelu. Ponadto, zamieszczono wyniki optymalizacji zbioru układów testowych i na końcu przedyskutowano przydatność metody optymalizacji.

Rozdział 4 dotyczy redukcji poboru mocy na etapie syntezy logicznej. Dokonano przeglądu istniejących metod syntezy dwupoziomowych układów logicznych o zredukowanym poborze mocy. Następnie przedyskutowano zachowanie się bramek w układzie w kontekście nowego modelu. Efektem jest opracowanie oryginalnego algorytmu syntezy układów dwupoziomowych o obniżonym poborze mocy. Zaproponowano też autorskie metody obliczania nowej miary aktywności układu. Przedyskutowano wpływ inwerterów wejściowych na energię pobieraną przez układ oraz możliwości redukcji strat mocy przez przekształcenie układów dwupoziomowych do trójpoziomowych. Porównano wyniki pracy oryginalnego algorytmu i zaadaptowanego z literatury.

W *Rozdziale 5* podsumowano niniejszą rozprawę i przedstawiono wypływające z niej wnioski.

W *Załączniku* zamieszczono kody źródłowe opracowanych i wykorzystywanych do badań programów oraz parametry energetyczne bramek zaprojektowanych do celów testowych.

ROZDZIAŁ 1.

Modelowanie strat energii w cyfrowych układach CMOS

W ostatnich kilkunastu latach dynamicznie rozwijają się metody projektowania układów elektronicznych prowadzące do redukcji poboru mocy. Jest to z jednej strony efekt drastycznego wzrostu gęstości upakowania układów scalonych, co umożliwia wzrost możliwości funkcjonalnych projektowanych urządzeń, ale też powoduje, że ilość wydzielanego ciepła prowadzi do ekstremalnych gęstości mocy w układzie elektronicznym [DeM94]. Z drugiej strony szybki rozwój urządzeń przenośnych zasilanych bateryjnie również wymaga ciągłego rozwoju metod redukcji poboru mocy.

Optymalizacja jakiejś cechy pewnego obiektu wymaga zdefiniowania funkcji celu. Również w przypadku redukcji strat energii potrzebny jest odpowiedni model opisujący dokładnie i zwięźle konsumpcję mocy przez układ. Dysponując stosownym modelem, czyli analitycznym opisem zjawiska, można na drodze modyfikacji poszczególnych zmiennych wejściowych poszukiwać optimum.

Każdy proces projektowania wymaga weryfikacji osiągniętych rezultatów. Jest to potrzebne w celu kontroli i oceny projektu na jego poszczególnych etapach. Również w procesie projektowania układów elektronicznych o obniżonym poborze mocy w pewnym momencie trzeba oszacować wielkość konsumpcji mocy. Tu właśnie potrzebny jest stosowny model. Powszechnie znane i używane są w elektronice komputerowe symulatory układów elektronicznych (np. SPICE). Zatem wydaje się, że projektant cyfrowych układów dysponuje już narzędziem potrzebnym do weryfikacji swojego projektu. Jednakże czas potrzebny na symulację układu pomimo dużej dokładności jest zbyt duży w wielu przypadkach. Ponadto czasami trzeba tylko pobieżnie, ale bardzo szybko oszacować pobór mocy, aby podjąć właściwe decyzje projektowe. Dlatego w ostatnich kilkunastu latach powstało wiele modeli poboru mocy o różnej dokładności.

Model poboru mocy oprócz zastosowania do estymacji mocy może pozwolić na opracowanie nowych metod redukcji strat energii, jeśli zostaną wzięte pod uwagę zjawiska wcześniej zaniedbywane. Co więcej, nowy opis zjawisk fizycznych w układzie może być taki, że daje nowe możliwości optymalizacji wcześniej nie znane. W tym rozdziale będzie rozważane modelowanie dynamicznych strat energii powstałych podczas przełączania bramki a związanych z ładowaniem, rozładowaniem bądź przeładowaniem pojemności występujących w cyfrowych

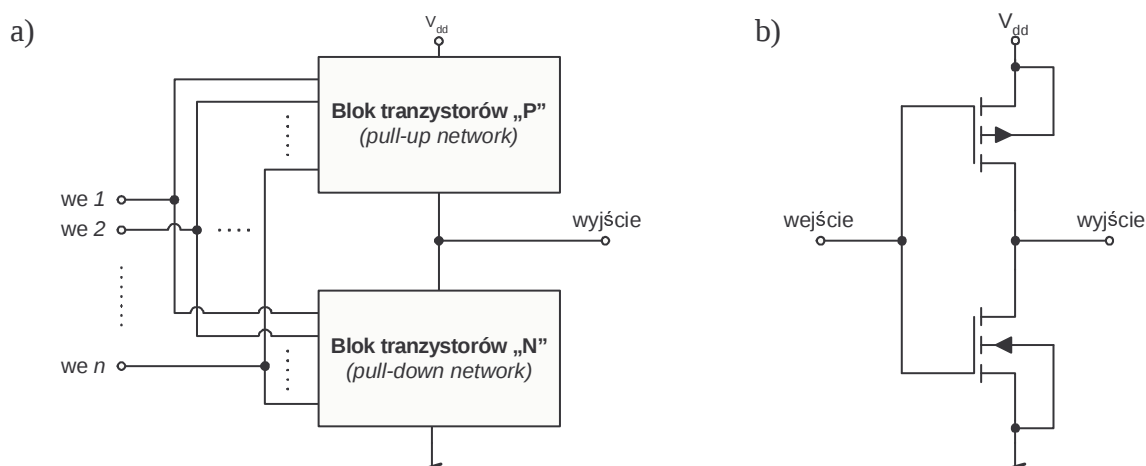
układach CMOS. W punkcie pierwszym przedstawiono modele spotykane w literaturze. Punkt drugi opisuje nowy model strat energii, uwzględniający pewne zaniechane dotychczas zależności w statycznych bramkach CMOS. Natomiast w punkcie trzecim podano metody i wyniki wyznaczania parametrów energetycznych bramek dla potrzeb opisywanych modeli.

1.1. Modelowanie tradycyjne

Stacyjne bramki CMOS są zbudowane z wykorzystaniem bloków komplementarnych tranzystorów MOS z przewodnictwem typu n i p . Taka konstrukcja bramki (rys. 1.1) zapewnia, że przez bramkę nie przepływa żaden prąd w czasie, gdy nie ma przełączenia – statyczny stan logiczny na wyjściu. Zatem nie ma wtedy poboru mocy z zasilania. Jedynie podczas zmiany stanu na wyjściu jest pobierana energia z zasilania związana z ładowaniem pojemności pasożytniczych występujących zarówno w tranzystorach MOS jak i wynikających z layout'u bramki oraz połączeń w układzie scalonym. Nasuwa się najprostszy model reprezentujący konsumpcję mocy zbudowany z idealnej bramki oraz kondensatora reprezentującego wyjściową pojemność obciążającą bramkę. Często uwzględnia on również pojemności pasożytnicze linii połączeniowych w układzie scalonym oraz pojemności wejściowe następnych bramek. Mówi się wtedy o pojemności węzła. Ponadto w najprostszym modelu zakłada się, że pobór energii z zasilania następuje tylko podczas narastającego zbocza na wyjściu bramki, czyli wtedy, gdy pojemność węzłowa jest ładowana. W czasie zbocza opadającego następuje rozładowanie przez załączone tranzystory NMOS. Dla inwertera z rysunku 1.1.b, który jest przełączany z częstotliwością f można opisać średni pobór mocy dynamicznej P_d jako:

$$P_d = C_L V_{dd}^2 f \quad (1.1)$$

gdzie: V_{dd} – napięcie zasilania, C_L – pojemność obciążająca bramkę. Powyższe równanie odzwierciedla energię, która wypłynęła ze źródła zasilania w związku z ładowaniem obciążenia.



Rys. 1.1. Budowa statycznych bramek CMOS: (a) zasada ogólna, (b) inwerter

W rzeczywistości jednak połowa energii jest rozpraszana w czasie narastającego zbocza sygnału wyjściowego bramki, a połowa magazynowana w pojemności obciążającej. Dopiero podczas następnego, opadającego zbocza, ta energia jest rozpraszana.

Na tym etapie modelowania pomija się straty związane z quasi-zwarcie oraz straty statyczne. Quasi-zwarcie polega na chwilowym utworzeniu się ścieżki prądowej pomiędzy zasilaniem a masą przez jednocześnie przewodzące tranzystory obydwu bloków podczas przełączania. Jest efektem niedoskonałości tranzystorów jako kluczy oraz nieskokowej zmiany napięcia wejściowego bramki.

1.1.1. Aktywność układu logicznego

W przypadku bramek wielowejsiowych równanie (1.1) nie może być bezpośrednio zastosowane, ponieważ nie każda zmiana sygnału wejściowego powoduje zmianę sygnału wyjściowego [Bel95]. Ponadto, gdyby nawet bramka wielowejsiowa generowała na swym wyjściu sygnał o częstotliwości f i wówczas było możliwe zastosowanie równania (1.1) do estymacji średniej mocy strat, to jednak w ogólnym przypadku sygnał logiczny w jakimś węźle układu nie musi być okresowy. Wtedy notacja częstotliwościowa nie może być stosowana [Naj93]. Dlatego wprowadza się parametr mówiący o aktywności bramki lub węzła. Aktywność przełączeniowa węzła (ang. *switching activity*) określa, jak często zmienia się stan logiczny w danym węźle układu. Inaczej mówiąc, aktywność przełączeniowa reprezentuje prawdopodobieństwo, z jakim możliwa jest zmiana stanu na wyjściu bramki w czasie jednego okresu zegara $T = 1/f$ [Bel95]. Jeśli potraktujemy sygnał logiczny jako stacjonarny proces stochastyczny to aktywność przełączeniowa jest nazywana prawdopodobieństwem przełączenia p_t (ang. *transition probability*) i jest definiowana jako wartość oczekiwana liczby przełączeń węzła w okresie zegara [Naj93]. Innymi słowy jest to spodziewana liczba przełączeń węzła [Ima98]. Takie podejście traktuje obydwie zbocza ($1 \rightarrow 0$) i ($0 \rightarrow 1$) jako równoważne, chociaż rzeczywista konsumpcja mocy z zasilania występuje tylko w przypadku zbocza narastającego – ładowanie pojemności obciążającej. Dlatego równanie (1.1) należy zmodyfikować o stałą $\frac{1}{2}$, a aktywność obliczać bez rozróżniania rodzaju zbocza. Zatem, ogólnie w statycznej bramce CMOS dynamiczne straty mocy są reprezentowane przez:

$$P_d = \frac{1}{2} C_L V_{dd}^2 f p_t \quad (1.2)$$

Zakładając nierozróżnialność zboczy współczynniki aktywności można obliczyć, wykorzystując pojęcie prawdopodobieństwa sygnału p_s (ang. *signal probability*), które jest definiowane jako prawdopodobieństwo tego, że sygnał logiczny przyjmuje wartość jeden. Parametr ten został

wprowadzony w metodach testowania układów logicznych [Par75]. Autor podaje również metody obliczania prawdopodobieństwa sygnału dla podstawowych bramek logicznych. Równania w ogólnej postaci, przy założeniu, że w układzie nie ma re-zbieżnych ścieżek sygnału (ang. *reconvergent*) można znaleźć w [Ima98]:

$$\begin{aligned} \text{bramka NOT : } & p_s(o) = 1 - p_s(i) \\ \text{bramka AND : } & p_s(o) = \prod_{i \in WEJ} p_s(i) \\ \text{bramka OR : } & p_s(o) = 1 - \prod_{i \in WEJ} (1 - p_s(i)) \end{aligned} \quad (1.3)$$

gdzie: p_s – prawdopodobieństw sygnału, o – wyjście, i – wejście, WEJ – zbiór wejść bramki. Efektywną implementację algorytmu Parkera można znaleźć w pracy [Roy93]. Autorzy, stosując pewną symboliczną reprezentację prawdopodobieństwa sygnału węzłów, uniknęli pamięcio- i czasochłonnych obliczeń.

Inną definicję prawdopodobieństw sygnału oraz przełączenia podano w [Has99]. Istota tych pojęć jest taka jak uprzednio, choć być może wzór definicyjny łatwiej obrazuje znaczenie. Jeśli na wejściach układu zdefiniujemy sygnał logiczny $x[n]$, który jest zsynchronizowany według dyskretnej skali czasu jako: $x[n] = x(nT) = x(t)|_{t=nT}$, to prawdopodobieństwo sygnału określa wzór:

$$p_s = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{n=1}^k x[n] \quad (1.4)$$

Jest to po prostu liczba jedynek logicznych w danym sygnale. Natomiast prawdopodobieństwo przełączenia opisuje wzór:

$$p_t = \lim_{t \rightarrow \infty} \frac{n_x(t)}{t} \quad (1.5)$$

gdzie: $n_x(t)$ – to liczba przejść sygnału $x(t)$ w czasie t , i liczba ta może zawierać również szpilki sygnału.

Zatem wykorzystując prawdopodobieństwo sygnału, można obliczyć prawdopodobieństwo przełączenia w węźle x jako [Naj94]:

$$p_t(x) = 2 p_s(x) p_s(\bar{x}) = 2 p_s(x)(1 - p_s(x)) \quad (1.6)$$

Powyższe rozważania są słuszne przy założeniu niezależności czasowej i przestrzennej sygnałów w układzie logicznym.

Najm w pracy [Naj91] wprowadził pojęcie gęstości przełączeń $d(x)$ (ang. *transition density*), które jest zdefiniowane jako współczynnik średniej liczby przełączeń. Bardziej precyzyjnie, dla stacjonarnego stochastycznego sygnału logicznego $x(t)$, dla którego przez $n_x(T)$ oznaczono liczbę losowych przejść pomiędzy stanami logicznymi 0 i 1 w przedziale $(-\frac{T}{2}, \frac{T}{2}]$ gęstość przełączeń to

wartość oczekiwana liczby przełączeń w jednostce czasu i wyraża się wzorem:

$$d(x) = \lim_{T \rightarrow \infty} \frac{n_x(t)}{T} \quad (1.7)$$

Wtedy iloczyn częstotliwości i prawdopodobieństwa przełączenia ($f p_t$) występujący w równaniu (1.2) może być zastąpiony właśnie przez gęstość przełączeń.

W niektórych pracach, w szczególności na temat estymacji mocy strat, autorzy próbują uwzględnić korelacje czasowe i przestrzenne sygnałów w układach logicznych [Mar94], [Mar96], [Mar98], [Fre01]. Poza tym wiele z tych prac podejmuje problem opóźnień czasowych, które mają wpływ na obliczaną aktywność układu.

Obok probabilistycznych metod obliczania aktywności układu, których przykłady przytoczono powyżej, można znaleźć metody symulacyjne [Gho92], [Mon97]. Autorzy, stosując metodę symulacji symbolicznej, uwzględniają przestrzenne korelacje między sygnałami powstałe w układzie (*reconvergence*) oraz opóźnienia bramek.

Inne podejście, choć z grupy metod symulacyjnych, zaprezentowano w [Brz99a], [Brz99b], gdzie zdefiniowano całkowitą liczbę przełączeń węzłów w układzie i miarę tę wykorzystano jako wskaźnik stopnia optymalizacji energetycznej układu. Liczba przełączeń węzła jest to całkowita liczba zmian stanu logicznego tego węzła w odniesieniu do wszystkich możliwych zmian wektorów wejściowych układu. Dla każdego węzła w układzie na podstawie zwykłej tabeli prawdy (rys. 1.2) tworzona jest tabela przejść [Brz00a] odnosząca się do głównych wejść układu (rys. 1.3). Zmiana stanu logicznego jest oznaczona przez "1" a brak zmiany przez "0". Suma jedynek w tej tabeli jest zatem liczbą przełączeń węzła i może reprezentować aktywność węzła. Jeżeli w tabeli przejść dla głównych wejść układu pewne zmiany wektorów zostaną wyróżnione jako bardziej prawdopodobne za pomocą odpowiednich współczynników, to otrzymujemy opis pewnego rozkładu prawdopodobieństwa zmian. W takiej sytuacji w tabelach przejść dla węzłów należy wpisać odpowiedni współczynnik zamiast jedynki. Teraz może być modelowany dowolny rozkład prawdopodobieństwa zmian wektorów wejściowych.

		$y_i(V_m)$					
$V_m(t)$	$V_m(t+1)$	V_0	V_1	V_2	V_3	$\dots$	V_{2^n-1}
V_0		0	1	0	0	$\dots$	1
V_1		1	0	1	1	$\dots$	0
V_2		0	1	0	0	$\dots$	1
V_3		0	1	0	0	$\dots$	1
$\vdots$							
$\vdots$							
V_{2^n-1}		1	0	1	1	$\dots$	0

$V_m(t)$ – aktualny wektor wejściowy
 $V_m(t+1)$ – następny wektor wejściowy
 '0' – brak zmiany stanu w węźle
 '1' – jest zmiana stanu węzła

Rys. 1.2. Tabela przejść dla węzła y_i

Input vector	Function inputs	Gates outputs
	$x_1 x_2 x_3 \dots x_{n-1} x_n$	$y_1 y_2 y_3 \dots y_j$
V_0	0 0 0 ... 0 0	$y_1(V_0) y_2(V_0) y_3(V_0) \dots y_j(V_0)$
V_1	0 0 0 ... 0 1	$y_1(V_1) y_2(V_1) y_3(V_1) \dots y_j(V_1)$
V_2	0 0 0 ... 1 1	$y_1(V_2) y_2(V_2) y_3(V_2) \dots y_j(V_2)$
$\vdots$	$\vdots$	$\vdots$
V_{2^n-1}	1 1 1 ... 1 1	$y_1(V_{2^n-1}) y_2(V_{2^n-1}) y_3(V_{2^n-1}) \dots y_j(V_{2^n-1})$

Rys. 1.3. Tabela prawdy dla wszystkich węzłów w układzie logicznym

Zakładając równomierny rozkład prawdopodobieństwa zmian wektorów wejściowych układu można założyć, że zmiana pomiędzy każdym z wektorów wystąpi tylko raz. Wtedy liczbę przełączeń można obliczyć bez tworzenia tabel przejść dla węzłów. Wystarczy tylko znajomość liczby logicznych zer i jedynek w tabeli prawdy dla danego węzła (moce zbiorów: F – ON-set i R – OFF-set). Zatem liczba przełączeń dla danego węzła i jest dana równaniem [Brz99a]:

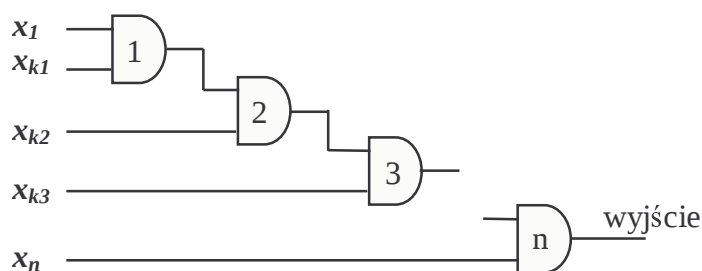
$$A_i = 2l_{F_i}l_{R_i} = 2^{n+1}l_{F_i} - 2l_{F_i}^2 \quad (1.8)$$

gdzie: l_{F_i} – moc zbioru F („jedynki”), l_{R_i} – moc zbioru R („zer”), n – liczba wejść do układu. Zakładamy, że funkcja jest w pełni określona i suma $l_{F_i} + l_{R_i} = 2^n$. Możliwa jest oczywiście modyfikacja równania (1.6) dla przypadku funkcji nie w pełni określonych. Obliczenie mocy zbiorów ON-set i OFF-set wymaga symulacji logicznej układu, co dla dużych układów może być czasochłonne i nieefektywne. Jednakże w pewnych sytuacjach można natychmiast podać równania określające aktywność węzłów. Jeżeli bramka o k wejściach jest bramką pierwszego poziomu w układzie n -wejściowym, inaczej mówiąc bramka realizuje $(n-k)$ -wymiarową kostkę¹, to aktywność jej wyjścia określa wzór [Brz99b]:

$$A_i(k, n) = 2^{2n-k+1} - 2^{2n-2k+1} \quad (1.9)$$

Zasadę tą można rozszerzyć na dalsze poziomy bramki w układzie pod warunkiem, że są to bramki tego samego typu lub pojawia się negacja. Sprowadza się to do tego, że formuła logiczna w danym węźle dotyczy jednej kostki wejściowej. Na rys. 1.4 przedstawiono kolejne bramki, dla których równanie (1.7) można zastosować. Aby obliczyć liczbę przełączeń dla bramki o numerze 1, trzeba w miejsce k w (1.7) podstawić k_1 , dla bramki drugiej k_2 , itd. [Brz00b], [Brz00c].

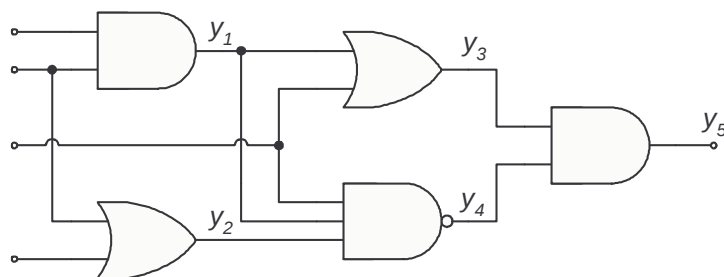
¹ Element c ze zbioru $\{0, 1, -\}^n$ nazywamy **kostką wejściową**; w przypadku, gdy c ma p składowych nieokreślonych „-”, kostkę nazywać będziemy p -wymiarową. Kostka taka może być, w typowy sposób, interpretowana jako iloczyn $n-p$ zmiennych [Łub98].



Rys. 1.4. Ilustracja do obliczania liczby przełączeń bramek w odniesieniu do pokrywanych kostek

W przypadku układów wielopoziomowych, w których są bramki różnego typu (np.: AND, OR, NAND, itd.) lub re-zbieżne ścieżki sygnału logicznego, nie można zastosować wzoru (1.7), gdyż rozpatrując równania logiczne, opisujące poszczególne węzły, może wystąpić sytuacja, że jakiś węzeł jest opisany przez więcej niż jedną kostką wejściową. W takiej sytuacji należy określić liczby zer i jedynek logicznych w danym węźle bezpośrednio z wymiarów kostek. Liczba pokrytych mintermów („jedynek”) przez kostkę jest jej wymiarem. Trzeba jednak najpierw, czy to przekształcając równanie węzłowe do postaci kanonicznej, czy korzystając z innych metod, znaleźć wszystkie kostki wejściowe dla danego węzła. Dla dużych układów może to być czasochłonne. Ponadto kostki mogą mieć części wspólne – mogą pokrywać te same mintermy funkcji realizowanej przez układ cyfrowy. Części wspólne kostek można znaleźć obliczając ich iloczyn. Następnie od sumy mintermów pokrywanych przez wszystkie kostki, czyli sumy wymiarów kostek, odejmujemy sumę wymiarów kostek powstałych z iloczynów na zasadzie „każdy z każdym”.

Rozważmy przykładowy układ z rysunku 1.5. Liczbę przełączeń węzłów y_1 i y_2 można obliczyć bezpośrednio z równania (1.7). Węzeł y_3 jest opisany równaniem: $y_3 = x_1x_2 + x_3$, co odpowiada kostkom: (11--) i (--1-), których wymiary to 4 i 8. Iloczyn tych kostek daje kostkę (111-) o wymiarze 2. Zatem liczba $l_F(y_3) = 4 + 8 - 2 = 10$. Ponieważ funkcja jest 4-wej. i zakładamy, że jest w pełni określona, to liczba maxtermów („0”) wynosi: $l_R(y_3) = 2^{2n} - l_F(y_3) = 16 - 10 = 6$. Węzeł y_4 jest opisany równaniem $y_4 = \overline{x_1x_2x_3}$, co odpowiada bramce AND i negacji, a inwerter nie zmienia liczby przełączeń. Aktywność wyjścia inwertera jest równa aktywności jego wejścia.



Rys. 1.5. Przykładowy wielopoziomowy układ kombinacyjny

W przypadku węzła y_5 otrzymujemy równanie, któremu odpowiadają trzy kostki: (110-), (0-1-) i (-01-). Iloczyny kostki pierwszej z drugą i trzecią dają kostki puste – kostka pierwsza nie ma części wspólnej z nimi. Iloczyn kostki drugiej z trzecią daje w wyniku: (001-), czyli kostkę 2-wymiarową. Liczba pokrywanych mintermów to: $I_F(y_5) = 2 + 4 + 4 - 2 = 8$.

Przedstawiony powyżej sposób modelowania aktywności układu, wykorzystujący liczbę przełączeń węzła może być alternatywą dla modeli probabilistycznych. Co więcej, w przypadku równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych, możliwe jest obliczenie aktywności układu, korzystając z zaproponowanych równań lub metody analizy rozmiarów kostek. Ponadto stał się on inspiracją autora do wprowadzenia modelu aktywności uwzględniającego wpływ zmiany sygnałów na wszystkich wejściach bramki CMOS na jej aktualny pobór mocy. Model został przedstawiony w punkcie drugim niniejszego rozdziału.

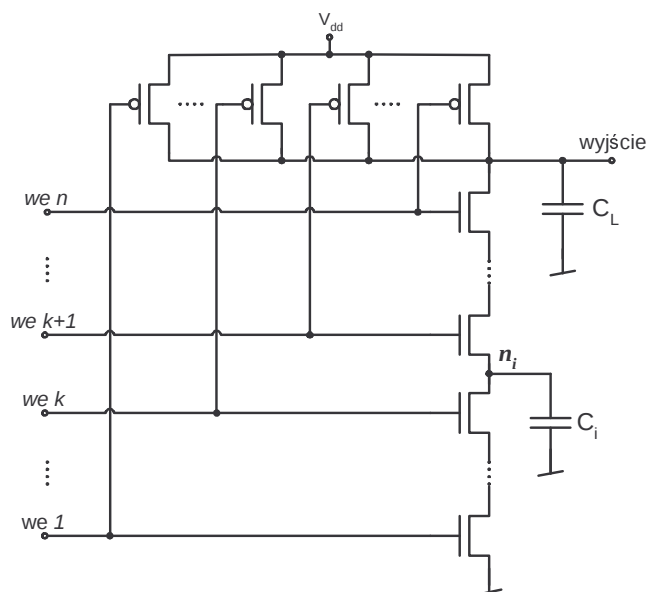
1.1.2. Pojemność węzłowa

Drugim, obok aktywności układu, ważnym parametrem, mającym duży wpływ na dokładność modelowania strat mocy jest wartość pojemności węzłowej występująca w równaniu (1.1). Stosując opisane wcześniej modele aktywności układu, pojemność węzłowa z natury musi mieć wartość stałą i niezależną od przyczyny przełączenia bramki. Gdyż rozważano tylko zmianę stanu logicznego w węźle układu, zaniedbując przyczynę tej zmiany. Jak zostanie to wykazane w rozdziale drugim nie jest to słuszne założenie (punkt 1.2.1), jednakże zasadniczo upraszcza modelowanie aktywności sieci logicznej.

Pojemność węzłowa, będąca całkowitym obciążeniem danej bramki, zwykle jest obliczana jako suma trzech składników [Bel95]: (1) pasożytnicza pojemność wyjściowa bramki sterującej, (2) całkowita pojemność połączeń z bramkami następnymi i (3) wejściowa pojemność sterowanych bramek. Pojemność wyjściowa bramki składa się z pojemności złączowych dren-podłoże tranzystorów, których dreny są bezpośrednio połączone z wyjściem bramki oraz pojemności dren-bramka (rys. 1.1.b). Wartość pojemności połączeń jest obliczana jako pojemność kondensatora płaskiego o powierzchni połączeń z uwzględnieniem efektu pojemności bocznej [Sak83]. Dane technologiczne każdego procesu CMOS podają stosowne wartości pojemności połączeń na jednostkę powierzchni oraz na jednostkę długości połączenia. Obliczając pojemność wejściową bramki CMOS zwykle uwzględnia się pojemność bramkową wszystkich tranzystorów w określonej bramce CMOS.

Autorzy wielu prac zajmujących się problemem optymalizacji układów cyfrowych pod względem poboru mocy, wykorzystując wcześniej przytoczone równanie (1.1) zakładają, że

pojemność węzłowa ma wartość stałą i jest znana [Ima98], [Ped96], [Roy93], [Wan99]. Jest to założenie upraszczające. W niektórych pracach wzięto pod uwagę pojemności węzłów wewnętrznych powstałych w łańcuchu szeregowo połączonych tranzystorów MOS [Asa95], [Bel95], [Has99], [Hos96], [Pra96], [Tsu94]. Problem ten dotyczy bramek wielowejsciowych (tj. NAND i NOR – rys. 1.6) oraz bramek złożonych (AndOrInvert i OrAndInvert). Najczęściej jako pojemność węzła wewnętrznego bierze się sumę pojemności złączowych dren-podłoże jednego z tranzystorów i źródło-podłoże następnego.



Rys. 1.6. Wielowejsciowa bramka NAND CMOS z pojemnością C_i węzła wewnętrznego n_i

Przeładowanie pojemności węzłów wewnętrznych zwykle występuje podczas zmiany stanu na wyjściu bramki. Możliwa jest także sytuacja, że podczas pewnej konfiguracji załączonych i wyłączonych tranzystorów nastąpi naładowanie C_i , a później przy innej konfiguracji nastąpi rozładowanie tej pojemności i nie będzie zmiany stanu na wyjściu bramki. Zatem aktywność wyjścia bramki nie jest wystarczającym warunkiem determinującym przeładowanie wewnętrznej pojemności. Dynamiczne starty mocy związane z i -tym węzłem wewnętrznym można opisać wzorem [Bel95]:

$$P_{d_wew} = \frac{1}{2} C_i V_i V_{dd} f a_i \quad (1.10)$$

gdzie: V_i – napięcie do jakiego ładuje się węzeł, zwykle niższe niż napięcie zasilania V_{dd} , a_i – współczynnik aktywności węzła. Pojawia się problem obliczenia aktywności węzłów wewnętrznych w bramce na podstawie wcześniej zdefiniowanych parametrów sygnału wejściowego, tj. prawdopodobieństwa sygnału lub przejść oraz gęstości przełączeń. Dodatkowo trzeba określić, jakie warunki muszą być spełnione, aby pojemność danego węzła wewnętrznego

mogła się naładować lub rozładować. Współczynnik aktywności węzła wewnętrznego może być zdefiniowany równaniem [Tsu94]:

$$a_i = \frac{p_{high}(i) p_{low}(i)}{p_{high}(i) + p_{low}(i)} \quad (1.11)$$

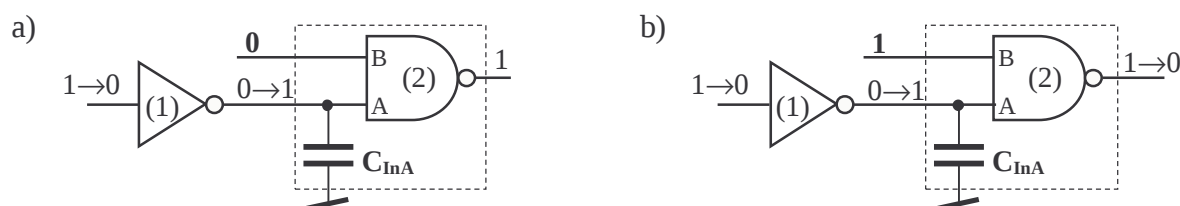
gdzie: p_{high} – prawdopodobieństwo ładowania, p_{low} – prawd. rozładowania węzła i . Aby obliczyć wartości tego prawdopodobieństwa w pracy [Tsu94], wprowadzono funkcje boolowskie, które reprezentują warunki ładowania i rozładowania węzłów wewnętrznych względem zmiennych wejściowych do bramki. Funkcje opisują warunki załączenia tranzystorów NMOS czy PMOS tak, aby powstały ścieżki pomiędzy danym węzłem a zasilaniem lub masą po to, aby możliwe było naładowanie lub rozładowanie węzła. Tak więc, korzystając z prawdopodobieństwa sygnału dla wejść do bramki, można obliczyć aktywność węzłów wewnętrznych. Inne podejście przedstawili autorzy pracy [Asa95]. Wprowadzili diagram stanów przejściowych opisujący prawdopodobieństwo zmiany stanu logicznego na danym wejściu bramki. Następnie zgodnie z tą ideą opisano zachowanie się węzła wewnętrznego i podano stosowne formuły pozwalające na obliczenie poboru mocy uwzględniając węzły wewnętrzne. W pracy [Hos96] dla łańcucha tranzystorów NMOS, takiego jak występuje w bramkach NAND, podano równania opisujące aktywność węzłów wewnętrznych. Wykorzystano pojęcie prawdopodobieństwa iloczynu zdarzeń losowych. Natomiast w [Has99] autorzy uwzględnili sytuację, gdy węzeł wewnętrzny jest nie połączony ani z zasilaniem ani z masą. Sytuacja taka ma duży wpływ na wartość pojemności wejściowej bramki nie rozważanej we wcześniej przytaczanych pracach. Określono stosowne funkcje boolowskie charakteryzujące warunki ładowania, rozładowania lub odizolowania (ang. *floating*) węzła. Wykorzystując prawdopodobieństwo sygnału i gęstość przełączeń oraz zaproponowane funkcje, podano równania definiujące pobór mocy danej bramki oraz wpływ zamiany kolejności wejść do bramki na jej pojemność wejściową a zatem na pobór mocy bramki sterującej – poprzedniej.

1.2. Nowy model strat energii w cyfrowych układach CMOS

W tradycyjnym modelowaniu strat energii w bramkach CMOS, opisanym powyżej zakłada się, że wartość stanu logicznego lub jego zmiana (zbocze sygnału) na jednym z wejść bramki nie ma wpływu na wartość pojemności wejściowej pozostałych wejść. Ponadto zakłada się również, że wewnętrzna pojemność wyjściowa bramki jest stała, a sposób, czyli przyczyna przełączenia bramki, nie ma wpływu na jej wartość. Poniżej zostanie przeprowadzona analiza zachowania się statycznej bramki CMOS podczas przełączania oraz będzie przedstawiony nowy model dynamicznych strat energii, w którym uwzględniono wpływ innych wejść.

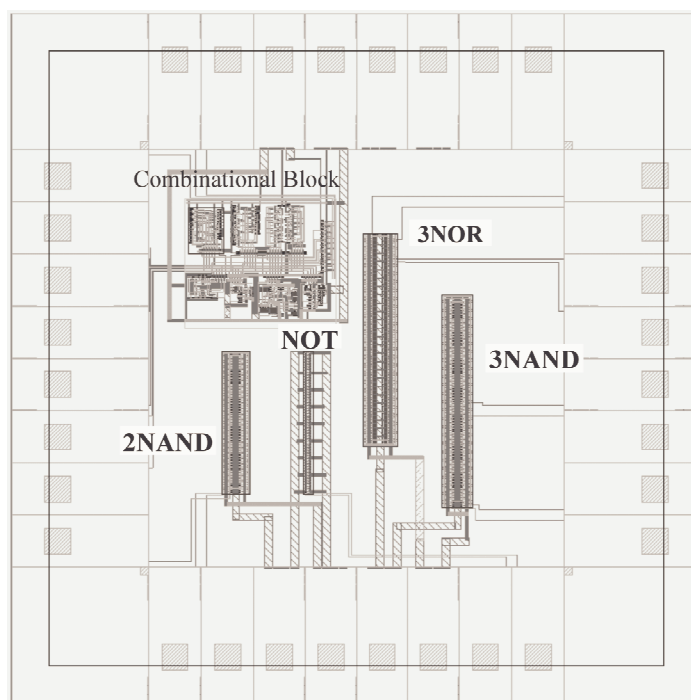
1.2.1. Motywacja

Uwzględniając nawet w najprostszym modelu tranzystora MOS pasożytnicze pojemności występujące pomiędzy jego końcówkami, nasuwa się pytanie, jaki będzie pobór energii bramki sterującej (1) z rysunku 1.7 w przypadku, gdy na drugim wejściu bramki sterowanej (2) będzie logiczny stan „0” lub „1”. Wstępna analiza schematu zastępczego bramki NAND, w którym uwzględniono pasożytnicze pojemności tranzystorów, pokazuje, że otrzymamy różne wartości poboru mocy. Zaprojektowano i wykonano układ scalony w technologii AMIS CMOS $0.7\mu\text{m}$ C07-MD [www1] zawierający kilka podstawowych bramek CMOS [Brz02]. Następnie wykonano pomiary prądu zasilania bramki NOT dla obu przypadków sterowania bramką NAND w celu odpowiedzi na wyżej postawione pytanie.



Rys. 1.7. Wpływ wejścia „B” na pojemność wejściową wejścia „A” w statycznej bramce CMOS

Badany układ scalony, którego layout przedstawiono na rysunku 1.8, składał się z czterech rodzajów bramek: inwerter, 2- i 3-wejściowa bramka NAND oraz 3-wejściowa NOR. Wymiary tranzystorów w bramkach dobrano w taki sposób, aby były jak najmniejsze, ale zapewniały maksymalny margines zakłóceń. W przypadku bramek wielowejściowych, gdy otrzymujemy rodzinę charakterystyk przejściowych zależnych od sposobu sterowania bramki – kombinacja



Rys. 1.8. Layout testowego układu scalonego

aktywnych wejść, starano się tak dobrać wymiary tranzystorów, aby skrajne charakterystyki umieścić w równych odległościach od napięcia wejściowego równego połowie napięcia zasilania. Tak, aby uśredniony margines zakłóceń był jak największy. Długość oraz szerokość kanałów tranzystorów zestawiono w tabeli 1.1. Poza tym taki dobór wymiarów zapewnia porównywalne czasy narastania i opadania napięcia wyjściowego bramki podczas przełączania. Układ został zaprojektowany w formie bloków po 100 równolegle połączonych bramek po to, aby zwiększyć wartości mierzonych prądów.

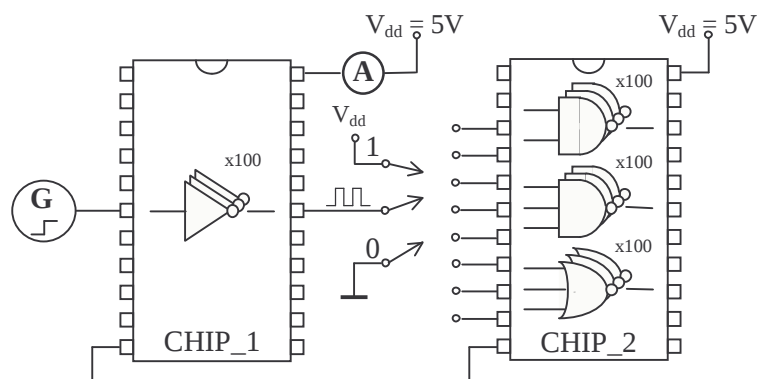
Tabela 1.1. Wymiary W/L tranzystorów bramek w testowym układzie scalonym [μm]

Typ tranzystora	NOT	2-wej. NAND	3-wej. NAND	3-wej. NOR
P	2,6/0,7	1,5/0,7	1,1/0,7	7,9/0,7
N	1,0/0,7	1,0/0,7	1,0/0,7	1,0/0,7

Pomiary przeprowadzono dla wszystkich bramek w układzie. Układ pomiarowy (rys. 1.9) składał się z dwóch układów scalonych oraz generatora i amperomierza. Prąd zasilania inwertera z pierwszego układu był mierzony dla kilku sposobów sterowania bramką z drugiego układu. Jedno z wejść bramek było sterowane z inwertera, podczas gdy na pozostałych były kolejno wszystkie możliwe kombinacje zer i jedynek. Wyniki pomiarów prądów tylko dla 3-wejściowej bramki NAND oraz obliczone pojemności ekwiwalentne zawarto w tabeli 1.2. Ostatnie dwie kolumny zawierają różnice prądów w odniesieniu do przypadku pierwszego (a00). Napięcie zasilania układów wynosiło 5V a częstotliwość sygnału wejściowego 15MHz. Wartości pojemności ekwiwalentnej obliczono według wzoru:

$$C_{weXequ} = (I_{Not+weX} - I_{Not}) / (f \cdot V_{dd}) \quad (1.12)$$

gdzie I_{Not} – średni prąd zasilania inwertera zmierzony bez obciążenia, $I_{Not+weX}$ – średni prąd zasilania inwertera sterującego wejście X bramki NAND. Małą literą a oznaczono aktywne wejście, czyli sterowane przez inwerter z pierwszego układu. Sekwencja ABC oznacza kolejność wejść do bramki w ten sposób, że wejście oznaczone przez A to bramka tranzystora, którego źródło jest podłączone do masy w bramce NAND lub do zasilania w bramce NOR.



Rys. 1.9. Układ pomiarowy do badania pojemności wejściowej bramki

Wyniki pomiarów zebrane w tabeli 1.2 mają większe wartości niż wyniki symulacji, ponieważ w rzeczywistym układzie pomiarowym występują pojemności pasożytnicze nie wzięte pod uwagę podczas symulacji. Między innymi są to pojemności: wewnętrznych połączeń w układzie, pól kontaktowych (ang. *bonding pad*), zabezpieczeń przed uszkodzeniem elektrostatycznym (ESD), końcówek układu scalonego, połączeń zewnętrznych, itd. W przypadku, kiedy jest sterowane wejście A, które jest najbardziej odległe od wyjścia, obserwujemy mały wpływ pozostałych wejść na wartość jego pojemności, za wyjątkiem przypadku (a11), kiedy to zmienia się stan wyjścia bramki. Gdy stan wyjścia bramki jest stały, następuje tylko ładowanie lub rozładowanie pojemności węzła wewnętrznego oraz pojemności między bramkami tranzystorów aktywnego wejścia a ich drenami lub źródłami. W przypadku, kiedy bramka jest przełączana i zmienia się stan na jej wyjściu, następuje przeładowywanie pojemności, które są włączone pomiędzy wejście a wyjście (np. C_{GD}), od napięcia $-V_{dd}$ do $+V_{dd}$. Zatem potrzeba dwa razy więcej ładunku, co objawia się większą wartością średnią prądu i większą efektywną pojemnością wejściową. W przypadkach sterowania pozostałymi wejściami (B lub C) można zaobserwować większe różnice dla poszczególnych kombinacji stanów logicznych na niesterowanych wejściach. Wiąże się to z tym, że załączone tranzystory NMOS powodują połączenie węzłów wewnętrznych z masą lub zasilaniem przez tranzystory PMOS i następuje ładowanie lub rozładowywanie pojemności tych węzłów oraz pojemności bramkowych sterowanych tranzystorów.

Tabela 1.2. Wyniki pomiarów prądu oraz obliczeń ekwiwalentnej pojemności wejściowej dla różnych sposobów sterowania 100 równolegle połączonych bramek NAND 3-wej.

sposób ster. ABC	pomiar		obliczenia		różnice	
	$I_{Not+weX}$ [mA]	I_{Not} [mA]	C_{weXequ} [fF]	$C_{weXequ}/\text{bramkę}$ [fF]	$I_{Not+weX}$ [%]	I_{weX} [%]
wejście A						
a00	1,97438	1,51180	6151	61,51	0,000	0,000
a10	1,97476	1,51180	6157	61,57	0,019	0,083
a01	1,97453	1,51180	6154	61,54	0,008	0,034
a11	2,11957	1,51180	8087	80,87	6,850	23,889
wejście B						
0a0	1,95899	1,51180	5964	59,64	-0,786	-3,442
1a0	1,96499	1,51180	6045	60,45	-0,478	-2,071
0a1	1,95824	1,51180	5955	59,55	-0,824	-3,615
1a1	1,98013	1,51180	6246	62,46	0,291	1,228
wejście C						
00a	1,98416	1,51180	6296	62,96	0,493	2,071
10a	1,98468	1,51180	6303	63,03	0,519	2,178
01a	1,98448	1,51180	6300	63,00	0,509	2,137
11a	2,00318	1,51180	6549	65,49	1,438	5,861

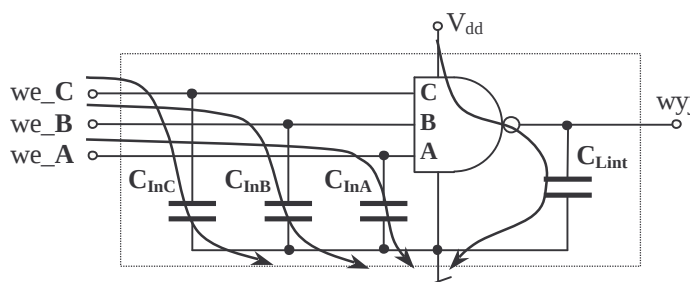
„a” – aktualnie sterowane wejście przez inwerter z pierwszego układu

Podobne obserwacje były motywem pracy [Has99], jednakże w zaproponowanym algorytmie optymalizacji mocy przez wybór najlepszej kolejności wejść do bramki wzięto pod uwagę sytuacje, gdy dren i źródło danego tranzystora NMOS nie są połączone z masą (lub zasilaniem dla PMOS). Wtedy pojemność wejściowa danego wejścia jest najmniejsza. Pozostałe przypadki połączeń drenu i źródła ujęto jako jeden. Co więcej, nie rozważano możliwości jednoczesnej zmiany stanu na kilku wejściach oraz zmian przeciwnych. Spowodowało to zawężenie rozpatrywanych przypadków sterowania bramką i pominięcie pewnych istotnych zależności, ale pozwoliło na zastosowanie tradycyjnych miar aktywności układu.

Powyżej otrzymane wyniki pomiarów nie są zbyt dokładne, gdyż nie uniknięto wpływu pewnych zjawisk pasożytniczych, jednakże obrazują skalę problemu niestalej pojemności wejściowej bramki CMOS w zależności od sposobu sterowania. Zatem przyczyna przełączenia lub nawet nieprzełączenia bramki powinna być wzięta pod uwagę w trakcie estymacji mocy strat układu cyfrowego. Ponadto należy uwzględnić wszystkie możliwe zmiany wektorów wejściowych bramki.

1.2.2. Definicja pojemności ekwiwalentnej

Przez pojemność ekwiwalentną rozumiemy pojemność kondensatora, który pobrałby tyle energii z zasilania, ile pobiera statyczna bramka CMOS w wyniku jednej zmiany wektora wejściowego. Zatem model poboru mocy przez bramkę, który zostanie dalej dokładniej zdefiniowany, będzie zbudowany z idealnej bramki, reprezentującej tylko zależności logiczne, oraz pojemności ekwiwalentnych obrazujących konsumpcję energii poprzez poszczególne końcówki (rys. 1.10). Rozpatrując wszystkie wejścia i zasilanie, określamy ekwiwalentne pojemności wejściowe i wyjściową. Kondensatory umieszczone na wejściach bramki z rys. 1.10 reprezentują pobór prądu z zasilania poprzez poprzednią bramkę. Są obciążeniem stopnia poprzedniego, sterującego. Kondensator na wyjściu to wewnętrzna pojemność obciążająca. Obrazuje on moc pobieraną poprzez końcówkę zasilania, ale nie związaną z przeładowaniem zewnętrznego obciążenia, tj. pojemności połączeń i wejściowych następnych bramek [Brz03].



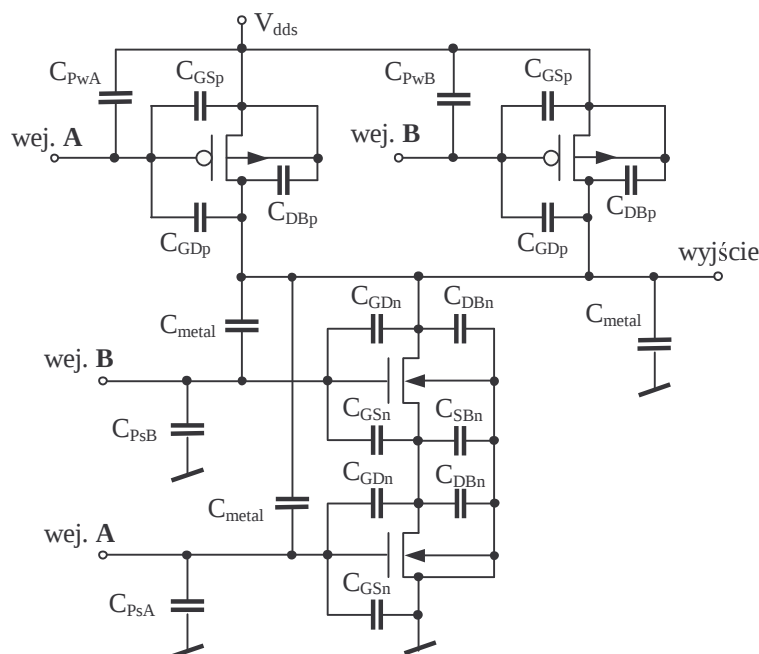
Rys. 1.10. Pojemnościowy model strat mocy 3-we. bramki NAND z pojemnościami ekwiwalentnymi

Koncepcja pojemności ekwiwalentnej została zaczerpnięta z pracy [Tur98]. Autorzy zaproponowali makromodel bazujący na koncepcji ekwiwalentnej pojemności obrazującej pobór mocy związany z quasi-zwarciami oraz z przerzutem napięcia w przełączanej bramce. Takie podejście pozwoliło autorom traktować te dwie składowe poboru mocy na równi ze składową pojemnościową. Dzięki temu mogli bezpośrednio i niezależnie od częstotliwości porównywać różne komponenty strat energii w bramce. Wartości tych pojemności obliczano jako całka ze stosownych prądów, podzielona przez napięcie.

Ze względu na często spotykany podział strat mocy w układach cyfrowych CMOS, można również zdefiniować ekwiwalentne pojemności związane z poszczególnymi składowymi poborami energii. Będziemy rozpatrywać straty pojemnościowe, quasi-zwarcia oraz statyczne.

Ekwiwalentna pojemność reprezentująca dynamiczne straty pojemnościowe obrazuje pobór mocy, związany z ładowaniem, rozładowaniem bądź przeładowaniem wewnętrznych pojemności pasożytniczych w bramce [Brz03]. Są to pojemności tranzystorów MOS (C_{GD} , C_{GS} , C_{DB} , C_{SB}) oraz pojemność złączowa obszarów dyfuzji i pojemności uformowane pomiędzy warstwami metali, polikrzemu a podłożem czy studnią (C_{metal} , C_{Ps} , C_{Pw}) lub innymi warstwami w rzeczywistym układzie scalonym (rys. 1.11).

Zmiana wektora wejściowego bramki powoduje pojawienie się pewnej kombinacji sygnałów na poszczególnych wejściach. Mogą to być zbocza narastające albo opadające lub stałe sygnały jeden albo zero. Zmieniający się stan na którymkolwiek wejściu powoduje, że następuje proces ładowania, rozładowania lub przeładowania pewnych wewnętrznych kondensatorów w bramce.



Rys. 1.11. Dwuwejściowa bramka NAND z wewnętrznymi pojemnościami pasożytniczymi

W efekcie określona ilość energii jest konsumowana bezpośrednio z zasilania lub pośrednio przez bramkę sterującą. Sytuacja ta dotyczy każdej zmiany wektorów wejściowych, nawet takiej, która nie powoduje przełączenia bramki. Ponadto zmiany pomiędzy różnymi wektorami będą powodować różne wartości pobieranej mocy. Zatem potrzeba miary określającej wszystkie możliwe zmiany wektorów wejściowych. Dlatego wprowadza się nową miarę aktywności układu, nazwaną **sposobem sterowania bramki** (punkt 1.2.3).

Wartości pojemności ekwiwalentnych można wyznaczyć na podstawie wartości prądu przepływającego przez końcówki bramki bazując na pomiarach lub symulacjach, korzystając ze wzoru:

$$C_{equ-X} = \frac{1}{V_{dd}} \int_{-T/2}^{T/2} i_X(t) dt \quad (1.13)$$

gdzie: X – końcówka bramki, przez którą przepływa prąd i_X , T – okres pomiędzy kolejnymi zmianami wektorów wejściowych. Obliczając pojemność ekwiwalentną związana tylko ze stratami dynamicznymi należy, odjąć prąd strat statycznych. Dodatkowo trzeba zapewnić warunki pracy bramki bez występowania zjawiska quasi-zwarcia.

Koncepcja pojemności ekwiwalentnej może być wykorzystana do określenia wielkości strat mocy związanych ze zjawiskiem quasi-zwarcia, czyli częściowego przewodzenia obu bloków tranzystorów w bramce podczas przełączenia (porównaj rys. 1.1). Wtedy przez bramkę przepływa prąd z zasilania do masy. Taka sytuacja ma miejsce, gdy sygnały wejściowe bramki mają relatywnie długi czas narastania lub opadania. Zakłada się, że quasi-zwarciovie straty mocy są pomijalne, gdy czasy narastania i opadania na wyjściu i wejściu bramki są porównywalne [Vee84]. Nie jest to zbyt precyzyjne określenie warunków występowania quasi-zwarcia, choć wystarczające w praktyce inżynierskiej. W pracach [Bis00], [Bis98], [Brz04b], [Brz04c], [Hed87], [Nos00] dokładniej zdefiniowano warunki występowania i wielkość strat quasi-zwarciovych.

Wartość prądu quasi-zwarciovego zależy od sposobu sterowania bramką [Brz02]. Decydujący wpływ mają dwa zjawiska. Liczba włączonych tranzystorów spośród połączonych równolegle (tj. PMOSy w NAND, i NMOSy w NOR) ma wpływ na wypadkową rezystancję danego bloku. Sterowanie tranzystorów połączonych szeregowo wpływa na napięcie progowe tych tranzystorów przez efekt podłożowy. Zatem sposób sterowania bramki również decyduje o wielkości quasi-zwarciovych straty mocy. Jednakże tylko takie sterowanie, które powoduje przełączenie bramki, jest istotne.

Równanie (1.13) może być zastosowane do obliczenia quasi-zwarcioviej pojemności ekwiwalentnej, pod warunkiem, że właściwie wydzielono prąd quasi-zwarciovowy z całkowitego prądu zasilania [Alv98], [Kan86], [Tur98], [Yac89]. Wartość strat quasi-zwarciovych zależy

wprost proporcjonalnie od czasu narastania lub opadania napięcia wejściowego. Ponieważ, przyjmując koncepcję pojemności ekwiwalentnej w modelu start mocy, nie ma informacji o czasach w pracy [Brz04c], zaproponowano wykorzystanie ekwiwalentnej pojemności strat pojemnościowych do opisu wielkości strat quasi-zwarciovych na podstawie stosunku pojemności wyjściowej do wejściowej.

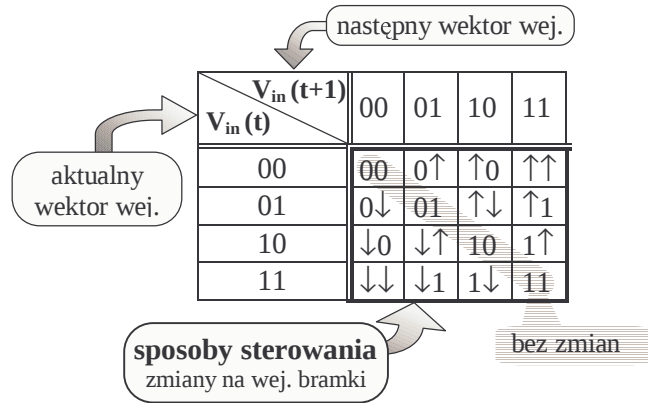
Statyczne straty występujące w układach CMOS również mogą być modelowane za pomocą pojemności ekwiwalentnej. Jednakże wartość prądu upływu, zależy między innymi od stanów logicznych na wejściach bramki [Abd04], [Roy03]. Im jest więcej odciętych tranzystorów w łańcuchu szeregowym, tym mniejsza wartość podprogowego prądu upływu. Dlatego potrzebna jest stosowna informacja o zachowaniu się układu. Taką miarą mogłoby być prawdopodobieństwo wystąpienia danego wektora na wejściu bramki, chociaż być może wystarczyłaby znajomość prawdopodobieństwa zmian wektorów oraz gęstości tych zmian. Można by wtedy obliczyć wartość wektora po zmianie oraz z gęstości zmian średni czas występowania danego wektora na wejściu.

Prawdopodobieństwo zmian wektorów wejściowych, które w następnych punktach zostanie zdefiniowane jako nowa miara aktywności układu logicznego, może być efektywnie zastosowane do opisu wszystkich składowych strat mocy, opisanych powyżej.

1.2.3. Definicja sposobu sterowania bramką

Dynamiczne pojemnościowe straty mocy w bramce CMOS zależą od aktualnie podanych sygnałów na wejścia. Ponadto straty quasi-zwarciovych również podporządkowują się tym zależnościom. Zatem proponowany model strat mocy wymaga zdefiniowania nowej miary aktywności układu. Tradycyjne miary aktywności, opisane w punkcie 1.1, nie są wystarczające, gdyż pozwalają określić jedynie liczbę przełączeń w jakimś węzle układu. Z przedstawionych powyżej rozważań (w szczególności punkt 1.2.1) wynika, że z punktu widzenia poboru mocy istotne jest zachowanie się sygnałów na wszystkich wejściach bramki CMOS. Ważna jest nie tylko przyczyna zmiany stanu na wyjściu bramki, ale każda zmiana sygnałów na wejściach.

Rozważmy dowolną bramkę dwuwejściową i wszystkie możliwe zmiany dwuelementowych wektorów binarnych, które są jej wektorami wejściowymi. Dla wygody zapiszemy tabelę ze wszystkimi możliwymi wektorami i ich zmianami (rys. 1.12). W ten sposób otrzymujemy wszystkie możliwe sposoby sterowania bramką 2-wejściową. Strzałki oznaczają narastające lub opadające zbocza sygnałów wejściowych. Można zauważyć, że przekątna tej tabeli reprezentuje sposoby sterowania bez zmiany stanów wejściowych, jednak jest to również sposób sterowania,



Rys. 1.12 Wszystkie możliwe zmiany wektorów wej. dla bramki 2-wejściowej – sposoby sterowania zwłaszcza z punktu widzenia wejść do układu. Zatem na podstawie powyższej tabeli można podać całkowitą liczbę sposobów sterowania N_{dw} dla n -wejściowej bramki jako:

$$N_{dw} = 2^{2^n} - 2^n + 1 \quad (1.14)$$

Z punktu widzenia modelu poboru mocy potrzebna jest znajomość liczby poszczególnych sposobów sterowania, potrzebujemy więc tabeli takiej jak na rysunku 1.12, wypełnionej odpowiednimi liczbami. Nie jest to zbyt wygodne podejście dla dalszych rozważań. Jeśli jednak liczbę poszczególnych sposobów sterowania odniesiemy do liczby wszystkich zmian wektorów wejściowych, czyli sumy liczb z tabeli, to otrzymamy parametr określający prawdopodobieństwo określonego sposobu sterowania. Zatem możemy podać następującą definicję:

Definicja 1.1: *Prawdopodobieństwo sposobu sterowania to prawdopodobieństwo wystąpienia określonego sposobu sterowania bramką w odniesieniu do wejść układu. Będzie obliczane jako liczba wystąpień danego sposobu sterowania dzielona przez sumę wszystkich zmian wektorów wejściowych układu logicznego.*

Powyższa definicja pozwala na uwzględnienie dowolnego rozkładu prawdopodobieństwa sposobów sterowania układem podczas estymacji lub optymalizacji strat mocy w układzie. Ponadto przypadki braku zmian na wejściach jakiegokolwiek układu są uwzględnione i suma prawdopodobieństw dla wszystkich sposobów sterowania wynosi jeden. Zatem dla dowolnego węzła j sterowanego przez n -wejściową bramkę można zapisać następujące równanie:

$$\sum_{k=0}^{2^{2^n}-2^n} p(k) = 1 \quad (1.15)$$

gdzie: k – numer sposobu sterowania, $p(0)$ – prawdopodobieństwo braku zmian wektora wejściowego, $p(1)$ – prawdopodobieństwo pierwszego sposobu sterowania, itd.

Definicję 1.1 można rozszerzyć tak, aby uwzględnić szpilki występujące w układzie, które są spowodowane opóźnieniami, wtedy prawdopodobieństwo $p(dw_j)$ sterowania węzła j w układzie jest opisane równaniem:

$$p(dw_j) = \frac{1}{k} \lim_{k \rightarrow \infty} \sum_{i=1}^k dw_j(i) \quad (1.16)$$

Jego wartość jest obliczana, podobnie jak poprzednio, jako liczba wystąpień danego sposobu sterowania (dw_j), ale zawierająca również zmiany wektorów wywołane przez szpilki, dzielona przez całkowitą liczbę zmian wektorów wejściowych układu przy k zmierzającym do nieskończoności.

1.2.4. Model statycznej bramki CMOS

W dwóch poprzednich punktach zdefiniowano parametry, które pozwolą na zbudowanie nowego modelu poboru mocy dla bramki CMOS [Brz03]. Proponowany model strat mocy jest modelem pojemnościowym, zbudowany w oparciu o koncepcję pojemności ekwiwalentnej. Schemat zastępczy bramki został już wprawdzie przedstawiony na rysunku 1.10, jednak wymaga uzupełnienia o nową miarę aktywności układu. W poprzednich punktach wykazano, że wartość pojemności ekwiwalentnej jest funkcją sposobu sterowania bramki. Zatem prawdopodobieństwo sposobu sterowania będzie współczynnikiem określającym udział danej wartości pojemności ekwiwalentnej w całkowitej mocy strat. Tak więc całkowity pobór energii przez bramkę CMOS, reprezentowany przez pojemność ekwiwalentną, będzie obliczany jako suma iloczynów wartości pojemności ekwiwalentnej i prawdopodobieństwa danego sposobu sterowania bramką dla wszystkich możliwych sposobów sterowania. Całkowita ekwiwalentna pojemność $C_{T_equ_X}$ związana z końcówką X bramki g może być wyrażona równaniem:

$$C_{T_equ_X}(g) = \sum_{dw_g} c_{equ_X}(dw_g) \cdot p(dw_g) \quad (1.17)$$

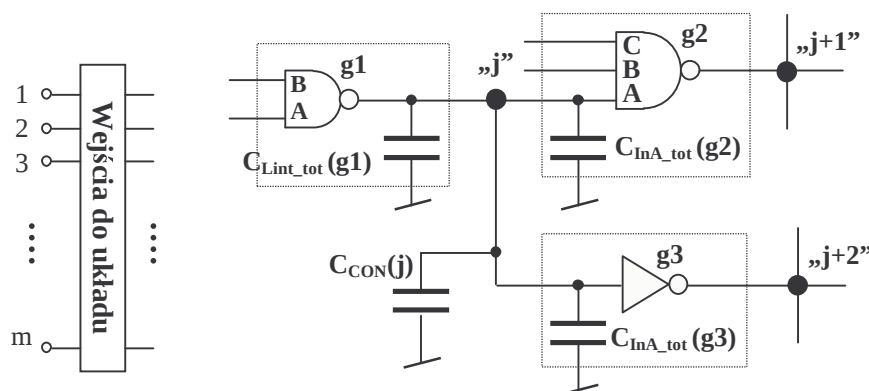
gdzie: $c_{equ_X}(dw_g)$ – cząstkowa, ekwiwalentna pojemność bramki g związana z końcówką X dla sposobu sterowania dw . Sumowanie jest przeprowadzone dla wszystkich możliwych zmian wektorów wejściowych bramki.

1.2.5. Model sieci logicznej

Model układu kombinacyjnego, zbudowanego z bramek CMOS, opiera się na powyżej przedstawionym modelu bramki, jednakże miara aktywności jest teraz traktowana nieco szerzej. Jako zmienna wejściowa dla estymacji bądź optymalizacji poboru mocy potrzebny jest rozkład prawdopodobieństwa zmian wektorów wejściowych do układu. Możemy mówić o sposobie sterowania całego układu. Każda zmiana wektora na głównych wejściach powoduje zmiany na wejściach bramek wewnątrz układu. Generowane są sposoby sterowania bramek w zależności od

struktury układu. W tym kontekście brak zmiany wewnątrz układu też jest sposobem sterowania bramki i w myśl definicji 1.1 możemy mówić o prawdopodobieństwie braku zmiany wektora. W przypadku, gdy różne sposoby sterowania układu powodują taki sam sposób sterowania pewnej wewnętrznej bramki, należy zsumować prawdopodobieństwa tych sposobów. Zgodnie z definicją 1.1 suma prawdopodobieństw wszystkich sposobów sterowania bramki bądź układu jest równa jeden (równanie (1.15)). Zatem nawet gdyby w skrajnym przypadku dowolny sposób sterowania układu powodował zawsze taki sam sposób sterowania bramki, to będzie on miał prawdopodobieństwo równe jeden, a pozostałe przypadki zero.

Kondensatory na rysunku 1.13 reprezentują prąd pobierany z zasilania poprzez bramkę sterującą węzeł j , więc zgodnie z prądowym prawem Kirchhoffa wartości pojemności mogą być zsumowane podczas obliczania całkowitej pojemności ekwiwalentnej tego węzła. Sumujemy odpowiednie wartości składowe pojemności, które zostały uprzednio obliczone jako całkowite ekwiwalentne pojemności związane z końcówkami bramek połączonych z węzłem j . Równanie (1.17) może być bezpośrednio wykorzystane do obliczenia wewnętrznej pojemności ekwiwalentnej dla bramki $g1$, sterującej rozpatrywany węzeł. Bierzemy pod uwagę zmiany wektorów na jej wejściu, czyli sposoby sterowania węzła j . Obliczając pojemności wejściowe dla bramek $g2$ i $g3$ rozpatrujemy zmiany wektorów na ich wejściach, czyli sposoby sterowania węzłów następnych $j+1$, oraz $j+2$. Uwzględniając te warunki, równanie (1.17) również może być zastosowane do obliczeń.



Rys. 1.13. Model strat mocy w węźle j przykładowego układu logicznego CMOS

Pojemność połączeń $C_{CON}(j)$ jest również traktowana w sensie pojemności ekwiwalentnej. Gdy napięcie w jakimś węźle układu narasta, wtedy następuje ładowanie pojemności połączeń i jest konsumpcja mocy z zasilania. Zatem tabela ekwiwalentnej pojemności połączeń jako funkcja sterowania węzła będzie wypełniona tylko dwoma wartościami, tj. rzeczywistą wartością pojemności dla tych przypadków sterowania bramki, kiedy napięcie w węźle jest narastające, a dla pozostałych przypadków przyjmie wartość zero. Jako miarę aktywności węzła można

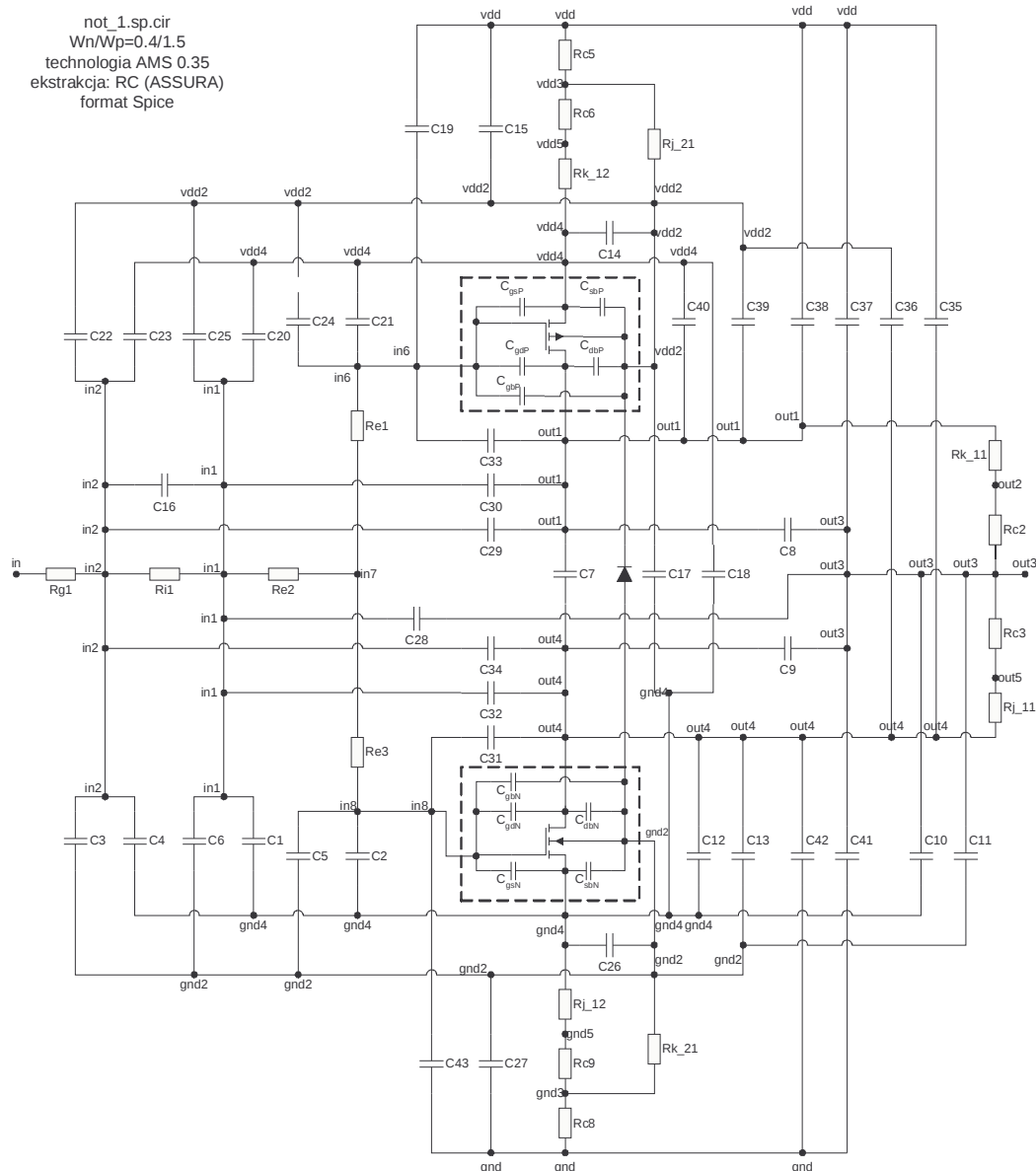
wykorzystać prawdopodobieństwo sposobu sterowania bramki sterującej dany węzeł i wtedy równanie (1.17) może być stosowane do obliczenia całkowitej ekwiwalentnej pojemności połączeń węzła.

Wejścia do układu także mogą być traktowane jako węzły wewnętrzne układu. Można zatem obliczyć ekwiwalentną pojemność wejściową układu w sposób analogiczny do obliczania pojemności wejściowej bramek. Uwzględniamy tylko pojemności wejściowe bramek pierwszego poziomu i połączenia. Analizując pobór mocy całego układu, należy wydzielić jego pojemność wejściową, gdyż stanowi ona obciążenie stopnia poprzedniego.

1.3. Ocena własności energetycznych statycznych bramek CMOS

Zarówno tradycyjny jak i nowy model poboru mocy wymaga wyznaczenia parametrów, określających własności energetyczne bramek. W przedstawionych modelach rolę tę pełnią pojemności kondensatorów ekwiwalentnych modelujących pobór energii. W punkcie 1.1.2 opisywano już pojemność węzłową dla tradycyjnego modelu.

Zaproponowany, nowy sposób modelowania strat mocy w układach cyfrowych CMOS wymaga wyznaczenia w właściwy sposób parametrów energetycznych dla bramek logicznych [Brz03]. Dla nowego modelu są to pojemności ekwiwalentne w funkcji sposobów sterowania, nie jest to jedna stała wartość jak w modelu tradycyjnym. Ponieważ będziemy rozważać dynamiczne, pojemnościowe straty mocy, można by wartości pojemności obliczyć na podstawie schematu zastępczego uzyskanego z layout'u bramki uwzględniającego parametry pasożytnicze. Rysunek 1.14 przedstawia taki schemat dla inwertera zaprojektowanego w technologii AMS 0,35 μm [www2] z uwzględnieniem pasożytniczych pojemności i rezystancji. Ponieważ pakiety oprogramowania przeznaczone do projektowania układów scalonych, zwykle wyposażone są w odpowiednie narzędzia służące do ekstrakcji netlisty, uzyskanie takiego schematu nie stanowi problemu. Zatem dysponując schematami zastępczymi dla bramek CMOS można by obliczyć pojemność ekwiwalentną dla wszystkich sposobów sterowania. Jednakże jest to podejście czasochłonne, zwłaszcza dla bardziej rozbudowanych schematów, a otrzymane wyniki będą obarczone dużym błędem, gdyż pojemności pasożytnicze tranzystora są nieliniowe, zwłaszcza złączowe (C_{DB} , C_{SB}). Oprócz tego, podczas przełączania bramki następują dynamiczne procesy rekonfiguracji układu, niektóre pojemności się przeładowują, inne nie w pełni się ładują, lub rozładowują. Z tych powodów jest prawie niemożliwe obliczenie dokładnych wartości pojemności ekwiwalentnej przy użyciu analitycznych metod rozwiązywania obwodów, a na pewno bardzo czasochłonne.



Rys. 1.14. Schemat zastępczy inwertera z parametrami pasożytniczymi (tech. AMS 0,35μm)

Zastosowanie właściwych metod pomiarowych lub symulacji pozwala na łatwe i dokładne wyznaczenie parametrów energetycznych bramek dla potrzeb nowego modelu. Wyniki otrzymane dla nowego modelu posłużą do wyliczenia pojemności węzłowych dla tradycyjnego modelowania poboru mocy. Wartości obliczone w ten sposób będą dokładniejsze, gdyż w pewnym zakresie uwzględniają wpływ sposobu sterowania bramki na jej pobór mocy.

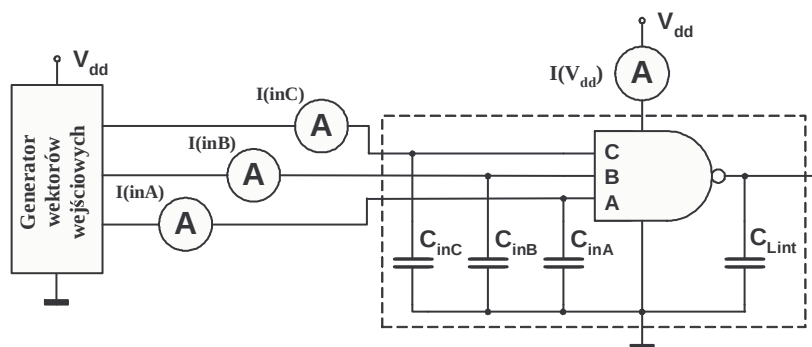
1.3.1. Parametry zależne od sposobu sterowania bramki – nowy model

Zanim zostaną dokładnie określone warunki pomiarów mocy, rozważymy zachowanie się statycznej bramki CMOS podczas przełączania. Wtedy to powstają największe straty mocy w cyfrowych układach CMOS. Ponadto trzeba określić warunki, pozwalające na wydzielenie dynamicznych, pojemnościowych strat mocy, ponieważ tymi będziemy się głównie zajmować.

Gdy napięcie wejściowe bramki jest większe od napięcia progowego tranzystora NMOS (V_{TN}) i mniejsze niż napięcie zasilania pomniejszone o wartość napięcia progowego tranzystora PMOS ($V_{dd} - |V_{TP}|$), wtedy istnieją warunki do wystąpienia zjawiska quasi-zwarcia. Przez bramkę, z zasilania do masy może przepływać prąd quasi-zwarcia. Jednakże nie zawsze quasi-zwarcie musi występować w przełączanej bramce. Jeśli zbocza napięcia wejściowego będą wystarczająco szybkie, to przez bramkę popłyną tylko prądy związane z wewnętrznymi i zewnętrznymi pojemnościami obciążeń i nie będzie quasi-zwarcia. W celu dokładnego wyjaśnienia tego efektu rozważmy narastające zbocze na wejściu inwertera CMOS. Gdy wejściowe napięcie wynosi zero, to napięcie wyjściowe jest równe V_{dd} , tranzystor NMOS jest wyłączony a PMOS jest włączony i pracuje w obszarze liniowym bez prądu drenu, pomijając prądy upływu. Kiedy narastające napięcie wejściowe osiągnie V_{TN} – napięcie progowe NMOS, wtedy tranzystor NMOS włącza się i wchodzi w zakres nasycenia, podczas gdy PMOS jest ciągle w zakresie liniowym. Zaczyna przepływać prąd przez oba przewodzące tranzystory. Dalsze zwiększanie napięcia wejściowego powoduje, że obydwa tranzystory są nasycone. Następnie sytuacja odwraca się i tranzystor PMOS wyłącza się, gdy napięcie wejściowe osiągnie wartość napięcia zasilania minus napięcie progowe tranzystora PMOS ($V_{dd} - |V_{TP}|$). Tranzystor NMOS pracuje teraz w obszarze liniowym z zerowym prądem drenu po rozładowaniu się pojemności obciążających. Jednakże z powodu pojemności sprzęgającej, występującej pomiędzy wejściem a wyjściem bramki, która składa się głównie z pojemności bramka-dren tranzystorów (C_{GD}), w początkowym etapie przełączania występuje przerzut napięcia wyjściowego bramki. Podczas przerzutu napięcie wyjściowe jest większe niż napięcie zasilania, zatem prąd nie płynie z zasilania do masy przez tranzystory, pomimo tego, że oba są włączane. Jeśli zbocze napięcia wejściowego jest na tyle szybkie, że tranzystor PMOS wyłączy się przechodząc z obszaru liniowego z pominięciem nasycenia, zanim przerzut się zakończy, to zjawisko quasi-zwarcia nie wystąpi. Zatem mierząc prąd zasilania dla wystarczająco szybkich zboczy napięcia wejściowego można wyodrębnić dynamiczną pojemnościową moc pobieraną przez bramkę.

Ocena parametrów energetycznych bramki CMOS dla nowego modelu konsumpcji mocy wymaga: (i) zastosowania odpowiednio szybkich zboczy napięcia wejściowego, tak, aby nie było quasi-zwarcia, (ii) wykonania pomiarów dla narastających i opadających zboczy oddzielnie oraz (iii) rozpatrzenia wszystkich możliwych zmian wektorów wejściowych – sposobów sterowania.

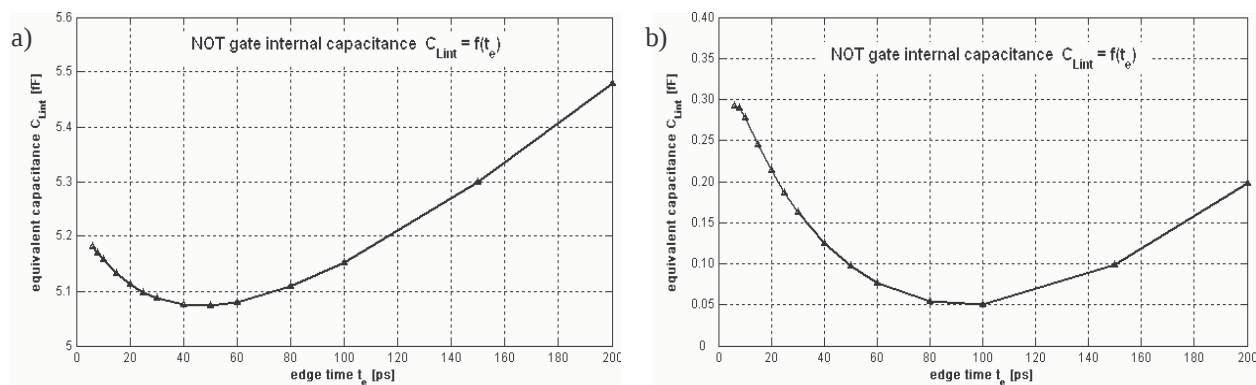
Spełnienie tych wymogów jest dużo łatwiejsze przy zastosowaniu technik symulacyjnych niż pomiarowych. Symulacje były przeprowadzone przy użyciu programu do symulacji układów elektronicznych PSPICE. Zastosowano układ pomiarowy przedstawiony na rysunku 1.15.



Rys. 1.15. Układ pomiarowy do wyznaczania wartości pojemności ekwiwalentnych

Badana bramka jest sterowana przez generator wektorów zbudowany ze źródeł dostarczających napięcia o kształcie trapezowym i krótkich czasach trwania zboczy. Źródła opisano za pomocą deklaracji odcinkowo-liniowych (PWL). Kolejne wektory były tak generowane, aby w jednym cyklu symulacyjnym zapewnić wszystkie możliwe sposoby sterowania bramki. Jako amperomierze do pomiaru prądów wejściowych i zasilania zastosowano, zgodnie z wymogami symulatorów bazujących na SPICE, zerowoltowe źródła napięcia. Dane otrzymane z analizy czasowej typu TRANSIENT dla wszystkich zmian wektorów były przeniesione do środowiska MATLAB w celu dalszych obliczeń. Na podstawie wartości prądów obliczono pojemności ekwiwalentne uwzględniając prąd statyczny. Na podstawie napięć wejściowych rozpoznawano kolejne wektory i określano sposoby sterowania bramki. Tak, więc w jednym cyklu symulacyjno-obliczeniowym otrzymywano wartości wejściowych i wewnętrznych pojemności ekwiwalentnych dla wszystkich sposobów sterowania badanej bramki. Wyniki zostały zorganizowane w tabele podobne do tabeli przejść z rysunku 1.12.

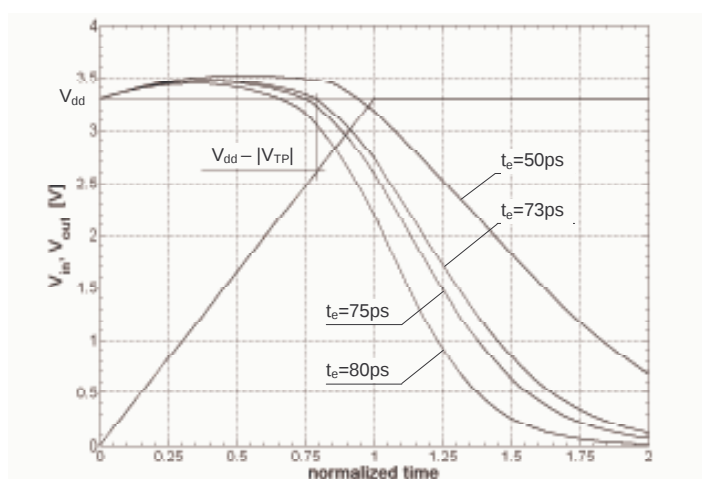
Pewnym problemem, który pojawił się podczas badania bramek, było dobranie właściwego czasu trwania zboczy sygnałów wejściowych. Zgodnie z opisem zachowania się bramki CMOS podczas przełączania, który był przedstawiony powyżej, żeby oddzielić składową pojemnościową od quasi-zwarciowej mocy trzeba stosować krótkie zbocza. Można spotkać w literaturze zastosowanie skoku napięcia w tym celu [Tur98]. Jednakże wraz ze skracaniem czasu trwania zbocza napięcia wejściowego, od pewnego momentu następuje wzrost poboru mocy (rys. 1.16). Efekt ten można wyjaśnić następująco. Krótsze zbocza wejściowe powodują większą wartość napięcia przerzutu – krótszy czas to większe napięcie spowodowane rozładowaniem pojemności sprzęgającej. Ponieważ podczas przerzutu wyjściowe napięcie bramki jest większe niż napięcie zasilania, to pojemność łączowa dren-podłoże tranzystora PMOS (C_{DBp} na rysunku 1.11) ładuje się wstępnie do ujemnego napięcia względem zasilania o wartości napięcia przerzutu ($-V_{ov}$). Po zakończeniu przerzutu i później, gdy napięcie wyjściowe maleje, następuje przeładowanie tej pojemności tak, że jej napięcie zmienia się od $-V_{ov}$ do $+V_{dd}$. Zatem gdy amplituda przerzutu



Rys. 1.16. Wewnętrzna pojemność ekwiwalentna w funkcji czasu trwania (a) opadającego i (b) narastającego zbocza napięcia wejściowego dla inwertera CMOS w tech. 0,35μm

rośnie potrzeba zasilania więcej energii na przeładowanie pojemności C_{DB} tranzystorów PMOS i w konsekwencji przepływa większy prąd. Efekt ten jest silniejszy w bramkach NAND, ponieważ tranzystory typu p są w nich połączone równolegle, co zwiększa wartość pojemności złączowej dren-podłoże.

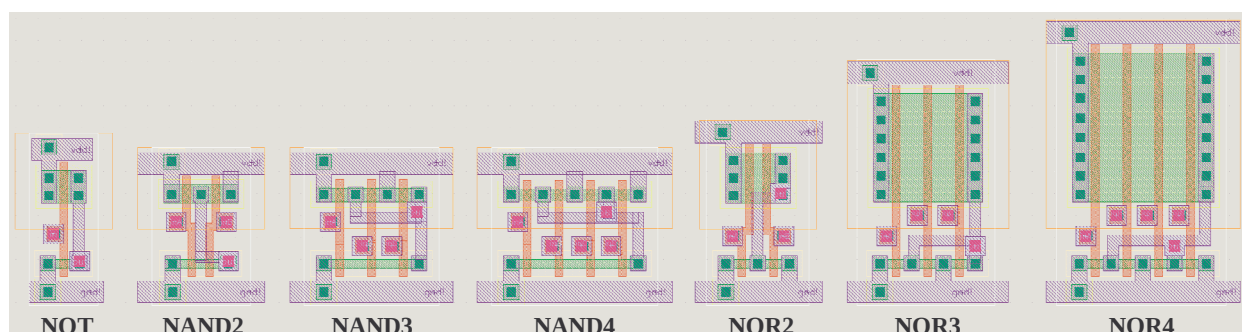
Wartość minimalna, otrzymana z powyższych wykresów, jako pojemność ekwiwalentna nie jest dobra, ponieważ wtedy występuje quasi-zwarcie. Zatem zdecydowano się określić czas zbocza wejściowego w taki sposób, że jest on możliwie najdłuższy, ale taki, aby nie występowało quasi-zwarcie. Obserwując unormowane przebiegi napięcia wyjściowego inwertera dla kilku czasów trwania zbocza wejściowego (rys. 1.17) oraz pamiętając wcześniejszy opis zachowanie się inwertera podczas przełączania, można zauważyć, że dla zboczy krótszych niż 73ps nie ma quasi-zwarcia w przypadku inwertera. W praktyce dla bramek wielowojściowych stosowano zbocza o czasie trwania $t_e=80$ ps.



Rys. 1.17. Przebiegi napięcia wyjściowego dla różnych czasów zbocza wejściowego w inwerterze

1.3.2. Wyniki oceny parametrów energetycznych bramek dla nowego modelu

Spełniając powyżej opisane wymagania, co przebiegów czasowych, warunków symulacji i układów pomiarowych, dokonano oceny parametrów energetycznych dla siedmiu statycznych bramek CMOS. Wyznaczono wartości ekwiwalentnych pojemności wejściowych i wewnętrznych dla następujących bramek: NOT, 2-, 3- i 4-wejściowy NAND oraz NOR. Bramki zostały zaprojektowane w środowisku CADENCE [www3] w technologii *Austriamicrosystems 0,35μm CMOS C35B3C0* [www2]. Topografie bramek przedstawiono na rysunku 1.18. Wymiary



Rys. 1.18. Layout'y bramek zaprojektowanych w technologii AMS 0,35μm, dla których wyznaczono pojemności ekwiwalentne

tranzystorów dobrano według tych samych zasad, jak dla układu testowego przedstawionego w punkcie 1.2.1. Tranzystory były jak najmniejsze ze względu na minimalizację poboru mocy, ale z drugiej strony takie, aby uzyskać maksymalny margines zakłóceń. W tabeli 1.3 zebrano wymiary W/L (szerokość/długość) kanałów tranzystorów w zaprojektowanych bramkach.

Tabela 1.3. Wymiary W/L [μm] tranzystorów badanych bramek (AMS 0,35μm)

typ tranzystora	NOT	NAND			NOR		
		2-wejścia	3-wejścia	4-wejścia	2-wejścia	3-wejścia	4-wejścia
P	1,5/0,35	0,85/0,35	0,65/0,35	0,55/0,35	3,2/0,35	4,95/0,35	6,8/0,35
N	0,4/0,35	0,4/0,35	0,4/0,35	0,4/0,35	0,4/0,35	0,4/0,35	0,4/0,35

Używając programu ASSURA z pakietu CADENCE do ekstrakcji elementów z layout'u, wygenerowano netlisty w formacie PSPICE dla bramek z uwzględnieniem pasożytniczych rezystancji i pojemności. Tranzystory MOSFET były modelowane z użyciem modelu BSIM3v3. Poczym dokonano symulacji, a wartości pojemności ekwiwalentnych obliczono według wzoru:

$$C_{Lx} = \frac{I_{dd_av}}{f \cdot V_{dd}} \quad (1.18)$$

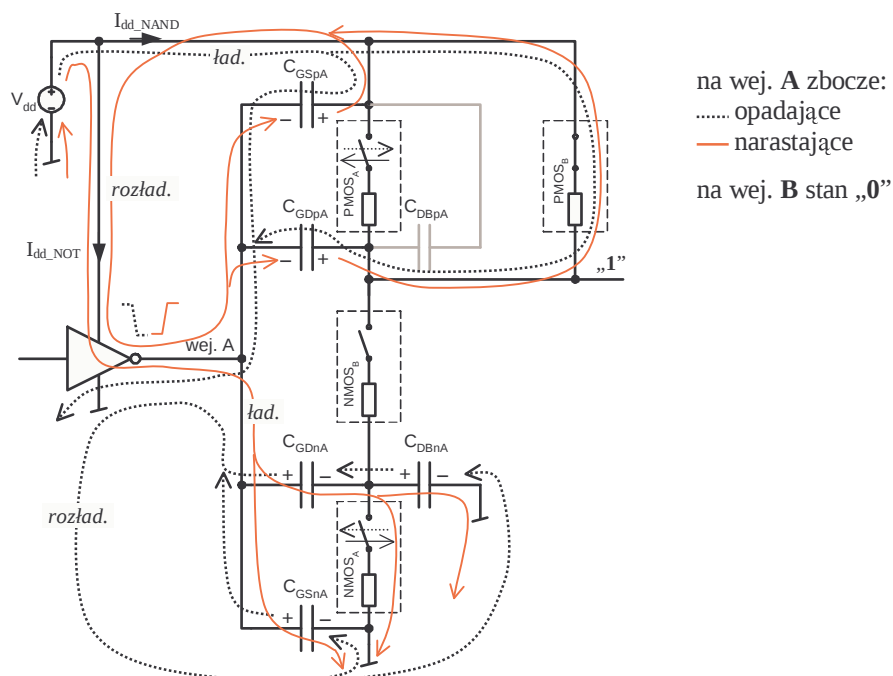
gdzie: I_{dd_av} – średni prąd przepływający przez końcówkę x badanej bramki w czasie jednego okresu z uwzględnieniem prądu statycznego. Aby mieć pewność, że procesy ładowania lub rozładowania kondensatorów w bramce zakończyły się, stosowano odpowiednią częstotliwość.

Dla wybranej technologii była to częstotliwość równa 100MHz. Czasy narastania i opadania zboczy wejściowych ustalono zgodnie z wnioskami z poprzedniego punktu. Jako prąd statyczny przyjmowano średnią wartość prądu płynącego pod koniec okresu.

Równanie (1.18) reprezentuje cały ładunek, który przepłynął przez daną końcówkę bramki bezpośrednio z zasilania lub przez bramkę poprzednią. W rzeczywistości połowa tej energii jest rozpraszana a połowa magazynowana w kondensatorach w bramce. W czasie następnej zmiany wektora wejściowego zmagazynowana energia jest rozpraszana lub wraca do zasilania przez rozważaną końcówkę bramki. To zależy od rekonfiguracji kondensatorów w bramce spowodowanej przez włączające lub wyłączające się tranzystory. Zatem pomiar prądu jako efektu zmiany sterowania bramki określa aktualną konsumpcję mocy z zasilania. Ilość energii zmagazynowanej w bramce po przełączeniu, jak i sposób jej rozpraszania, nie jest istotny w tych rozważaniach. Ważna jest tylko całkowita ilość energii pobieranej lub zwracanej do zasilania w wyniku jednej zmiany wektorów wejściowych bramki. Tak więc, zgodnie z wcześniej przyjętymi założeniami modelu oraz sposobem wyznaczania parametrów energetycznych bramek, możliwe jest otrzymanie ujemnych wartości pojemności ekwiwalentnych. Co więcej – jest to poprawny wynik.

Rozpatrując moc pobieraną przez bramkę bezpośrednio z zasilania, ekwiwalentna wewnętrzna pojemność będzie miała ujemne wartości w przypadku, gdy bramka jest sterowana bez zmiany stanu na wyjściu. Taka sytuacja ma miejsce, gdy przynajmniej na jednym wejściu bramki NAND jest „1”, lub „0” w przypadku NOR, i jest przynajmniej jedno narastające zbocze na dowolnym innym wejściu. W celu wyjaśnienia tego zjawiska uczynimy jeden krok wstecz, gdy na wejściu bramki było zbocze opadające. Wtedy prąd płynął z zasilania do bramki spowodowany ładowaniem pojemności bramka-źródło i bramka-dren sterowanego tranzystora PMOS (C_{GSp} i C_{GDp}) oraz kondensatora pasożytniczego uformowanego pomiędzy warstwą metalu lub polikrzemu linii wejściowej a dodatnio spolaryzowaną studnią. Na rysunku 1.19, na którym przedstawiono fragment bramki NAND tak właśnie sterowanej, prądy związane z narastającym zboczem wejściowym zaznaczono linią ciągłą, a z opadającym przerywaną. Narysowano tylko najważniejsze kondensatory. Jak zwykle, połowa pobranej energii została rozproszona na rezystancjach a połowa zmagazynowana w pojemnościach. Otrzymana z równania (1.18) dodatnia wartość pojemności ekwiwalentnej reprezentuje całą pobraną energię z zasilania. Następnie, gdy napięcie wejściowe narasta, uprzednio naładowane kondensatory tranzystora PMOS oraz linii wejściowej rozładowują się, a prąd rozładowania wypływa z bramki przez końcówkę zasilania. Obwód rozładowania zamyka się przez poprzednią, sterującą bramkę i aktywne wejście. Mierzac teraz ten prąd i uwzględniając jego kierunek, otrzymujemy ujemną

wartość ekwiwalentnej pojemności wewnętrznej i dodatnią wejściowej. Energia, uprzednio zmagazynowana w kondensatorach, jest teraz rozpraszana. Suma wartości wewnętrznej pojemności ekwiwalentnej dla obu przypadków jest równa zero, gdyż następuje ładowanie i rozładowanie tych samych kondensatorów.



Rys. 1.19. Rozpływ prądów w bramce NAND w przypadku sterowania ($A=\downarrow$, $B=0$) i ($A=\uparrow$, $B=0$) – ujemna wewnętrzna pojemność obciążenia

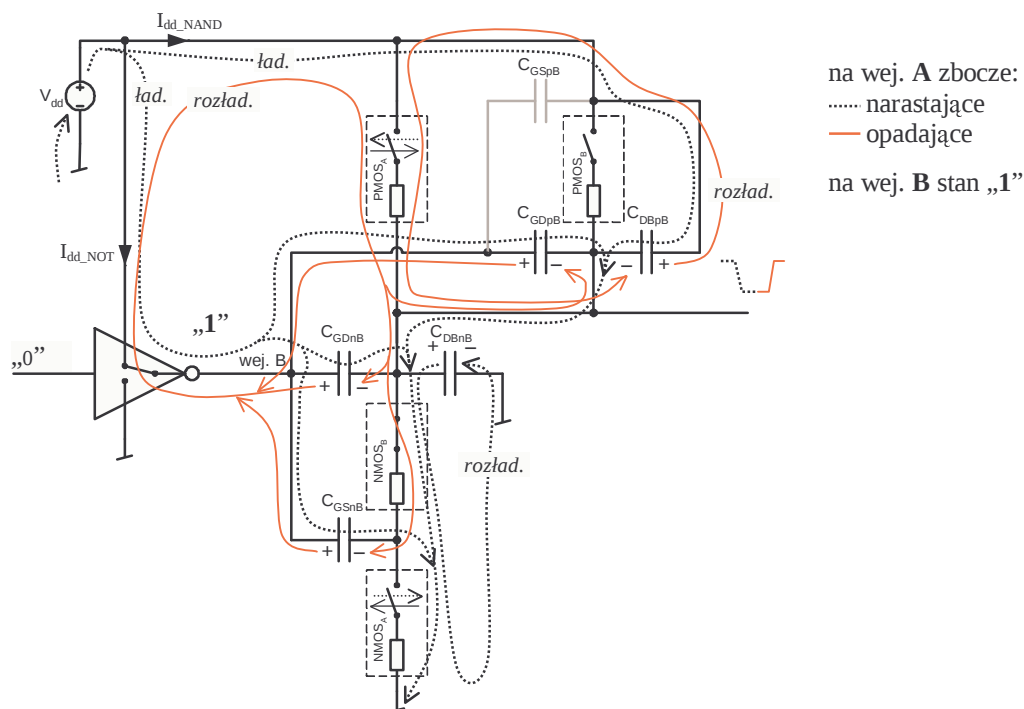
W tym momencie nasuwa się pytanie o poprawność takiego podejścia, przecież nastąpił przepływ prądu i to dwukrotnie, zatem pewna określona energia została rozproszona na rezystancjach. Więc wartość energii pobranej z zasilania nie może być równa zero, a taki właśnie wynik sugerują otrzymane wartości wewnętrznej pojemności ekwiwalentnej. Aby wyjaśnić ten problem, musimy przyjrzeć się wejściowej pojemności ekwiwalentnej.

Wejściowa pojemność ekwiwalentna reprezentuje konsumpcję energii z zasilania, ale przez poprzednią bramkę; tak założono w modelu i jest to zgodne z rzeczywistością. Podczas opadającego zbocza na wejściu prąd wypływa z bramki w związku z rozładowywaniem tych kondensatorów pasożytniczych, które są w jakiś sposób połączone z masą (głównie pojemności tranzystorów NMOS) i ładowaniem tych związanych z zasilaniem (dotyczy PMOS). Jednakże prąd ten nie wraca do zasilania przez końcówkę zasilającą poprzednią bramkę, lecz jego droga zamyka się przez masę. W tym przypadku przyjmujemy wartość zerową dla wejściowej ekwiwalentnej pojemności. Ponieważ wejście jest połączone z masą przez załączone tranzystory NMOS bramki sterującej, więc rozważana bramka nie ma możliwości poboru energii z zasilania przez poprzednią bramkę i rozpatrywane wejście.

W przypadku, kiedy wejściowe napięcie jest narastające, następuje ładowanie pojemności wejściowej bramki. Prąd płynie z zasilania przez załączone tranzystory PMOS bramki sterującej i wpływa przez aktywne wejście do rozważanej bramki (linia ciągła na rys. 1.19). Zatem wejściowa pojemność ekwiwalentna przyjmuje wartość dodatnią.

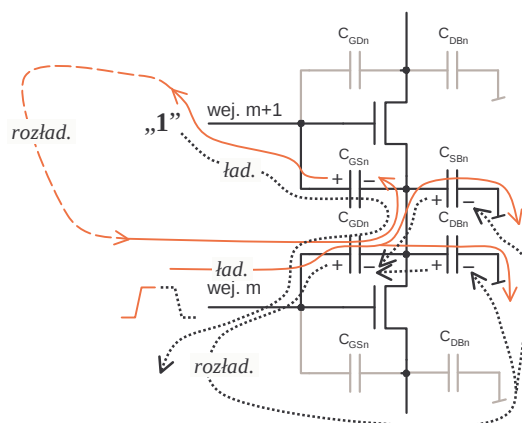
Możemy teraz wrócić do przypadku ujemnych wartości pojemności wewnętrznych. Jeśli zrobimy bilans energii dla obu przypadków, to widać, że dla każdego zbocza prąd wypływał ze źródła zasilania. Rozpatrując drogi przepływu prądu ze szczególnym uwzględnieniem końcówki zasilania bramki i wejścia, widać, że ekwiwalentna pojemność wejściowa jest przeszacowana a wewnętrzna zerowa (rys. 1.19). Uwzględniając rzeczywiste przepływy energii należałoby uwzględnić tylko prąd ładowania wpływający do bramki przez końcówkę zasilania w czasie opadającego zbocza. Natomiast rozładowanie i ujemną wartość pojemności wewnętrznej dla narastającego zbocza – pominąć. Odnośnie wejścia, to wpływający prąd należałoby pomniejszyć o wartość prądu wypływającego z bramki przez końcówkę zasilania, gdyż jest to prąd rozładowania. W ten sposób uwzględniamy rzeczywiste prądy związane z ładowaniem kondensatorów i wypływające ze źródła zasilania. Jednakże pozostawiając ujemne wartości, takie jak otrzymano z obliczeń, suma prądów, czyli bilans dla obu zboczy, dla źródła zasilania i obu bramek jest taki sam, jak w przypadku przyjęcia dodatnich wartości pojemności. Ostatecznie zdecydowano się na pozostawienie ujemnych wartości, gdyż nie ma to wpływu na wyniki estymacji mocy i optymalizacji układu (tabela 1.4). Chociaż ujemne wartości pojemności mogą czytelnika nieco razić, to pamiętając, że odzwierciedlają pobór mocy z zasilania, są to poprawne wartości i opisują rzeczywiste kierunki przepływu prądu.

W pewnych przypadkach sterowania bramki, również wejściowa pojemność ekwiwalentna może przyjmować ujemne wartości i tym razem też jest to poprawny wynik. Taka sytuacja ma miejsce, gdy na danym wejściu bramki jest stan „1” a zmiany na pozostałych wejściach powodują, że w wyniku rozładowywania się jakichś pojemności pasożytniczych prąd wypływa z tego wejścia i przez załączone tranzystory PMOS bramki sterującej wraca do zasilania. W bramce NAND, w takim przypadku sterowania, zmienia się stan wyjścia i rozładowują się pojemności tlenkowe (C_{GD} , C_{GS}). Na rysunku 1.20 zaznaczono drogi prądów ładowania i rozładowania dla takiego przypadku. Suma prądu zasilania inwertera I_{dd_NOT} dla obu zboczy wynosi zero. Sytuacja dualna w bramkach NOR występuje, gdy na wejściu jest stan „0”, a to z punktu widzenia nowego modelu start mocy jest nie istotne, gdyż wtedy, gdy na wejściu jest „0” prąd wypływający z wejścia trafia do masy nie do źródła zasilania, więc wejściowa pojemność ekwiwalentna wynosi zero.



Rys. 1.20 Rozpływ prądów w bramce NAND w przypadku sterowania ($A=\uparrow, B=1$) i ($A=\downarrow, B=1$) – ujemna ekwiwalentna pojemność wejściowa C_{inB}

W układach CMOS występuje jeszcze inny przypadek, który wygląda podobnie do opisywanego, ale dotyczy bramek o większej liczbie wejść i związany jest z szeregowo połączonymi tranzystorami. Jeśli na bramce jednego z tranzystorów w łańcuchu szeregowym (NMOS lub PMOS) jest stan „1”, a na bramce sąsiedniego tranzystora pojawia się zbocze narastające i te tranzystory są odizolowane od masy i zasilania (przez wyłączone sąsiednie), to w wyniku ładowania pojemności bramkowych jednego i rozładowania pojemności drugiego tranzystora wypływa prąd przez wejście będące w stanie „1” (rys. 1.21). Powstaje swego rodzaju dzielnik pojemnościowy pomiędzy wejściami a węzłem wewnętrznym i masą. W związku z tym na wartości prądów mają wpływ napięcia węzłów wewnętrznych przed zmianą wektora wejściowego, a więc konfiguracja tranzystorów dla poprzednich sposobów sterowania bramką,



Rys. 1.21. Rozpływ prądów wejściowych w wielowejściowej bramce NAND w przypadku sterowania ($1\uparrow$) – ujemna pojemność wejściowa

czyli historia zmian. Wyniki symulacji wykonanych dla 3-wejściowej bramki NAND pokazują różnicę w oszacowanej wartości pojemności wejściowej poniżej 2%.

W przypadku bramek wielowejściowych możliwe są jeszcze bardziej skomplikowane sytuacje, np. zmiany „na przemian”, których nie ma sensu szczegółowo analizować. Powyżej przedstawione analizy są pewnym uproszczeniem, ponieważ zakładamy pewien stan, napięcie kondensatorów przed i po przełączeniu. W trakcie samego przełączania bramki w związku z przeładowywaniem kondensatorów jak i przełączaniem tranzystorów następuje dynamiczne przekonfigurowanie układu. Uwzględnienie tych wszystkich procesów w analizie „na papierze” jest dość trudne. Chociaż ta analiza może być pewnego rodzaju weryfikacją otrzymanych wyników. Dlatego należy pozostać przy pomiarze średniego prądu przepływającego przez daną końcówkę bramki. Przyjęta zasada, uznająca ujemne wartości za właściwe, pozwala pominąć skomplikowane analizy ścieżek prądów. Ponadto przeanalizowane przykłady pokazują, że energia, która wypłynęła ze źródła bilansuje się z energią „krążącą” w bramkach.

Wyniki pojemności ekwiwalentnej wewnętrznej i wejściowej z uwzględnieniem wniosków z powyższych rozważań, tj. wartości ujemnych i zera dla zboczy opadających na wejściu, dla dwuwejściowych bramek NAND i NOR zebrano w tabeli 1.4. Sposoby sterowania, dla porządku ponumerowano oraz podzielono na trzy grupy: brak zmiany na wyjściu, zmiana na jednym wejściu oraz zmiany sygnałów na obu wejściach. Strzałki oznaczają zbocz sygnałów.

Tabela 1.4. Wyniki oceny parametrów energetycznych 2-wejściowych bramek NAND i NOR – wewnętrzna i wejściowa pojemność ekwiwalentna [fF] (technologia AMS 0,35μm)

Sposób sterowania		NOT			
		nr	[weA]	wyj.	
		1	↑	↓	0,054
		2	↓	↑	5,108
Sposób sterowania		2-wej NAND			
nr	[weA, weB]	wyj.	C _{Lint}	C _{In_A}	C _{In_B}
1	0 ↑	1	-2,136	0	3,002
2	0 ↓	1	2,387	0	0
3	↑ 0	1	-2,131	3,521	0
4	↓ 0	1	2,140	0	0
5	1 ↑	↓	-0,083	0,183	4,093
6	1 ↓	↑	4,930	-0,183	0
7	↑ 1	↓	-0,072	3,872	0,944
8	↓ 1	↑	5,558	0	-0,939
9	↑ ↓	1	0,280	3,690	0
10	↓ ↑	1	0,539	0	3,156
11	↑ ↑	↓	-1,847	3,651	3,929
12	↓ ↓	↑	7,230	0	0

Analizując otrzymane wyniki z powyższej tabeli można wyciągnąć następujące wnioski dla statycznych bramek CMOS:

Ekwiwalentna wewnętrzna pojemność obciążająca przyjmuje największe wartości w przypadkach, gdy napięcie wyjściowe bramki narasta. Sterowanie numer 6 i 8 dla bramki NAND oraz 2 i 4 dla NOR. Duży pobór prądu z zasilania jest spowodowany ładowaniem pojemności złączowej dren-podłoże C_{DB} tranzystorów NMOS oraz pojemności warstwy metalu linii wyjściowej. Na dodatek, pojemność sprzęgająca między aktywowanymi wejściami a wyjściem bramki jest przeładowywana (C_{GD} , C_{GS}). W przypadku wejść nieaktywnych następuje tylko rozładowanie (NMOS) lub ładowanie (PMOS) pojemności sprzęgającej (por. rys. 1.11). Dla bramki NOR wartości pojemności ekwiwalentnej są większe niż dla NAND z powodu połączonych równolegle tranzystorów NMOS (suma C_{DB}) oraz relatywnie większej pojemności sprzęgającej tranzystorów PMOS spowodowanej większymi wymiarami kanałów w bramce NOR (tabela 1.3).

W uproszczonej analizie poboru mocy przez bramkę CMOS, zwykle spotykanej w literaturze, zakłada się, że wewnętrzna pojemność obciążająca jest zerowa w przypadku opadającego napięcia na wyjściu bramki, ponieważ wtedy pojemność obciążająca rozładowuje się przez załączone tranzystory NMOS. Jednak dokładna analiza pokazuje, że w tej sytuacji prąd rozładowania pojemności bramka-źródło tranzystorów PMOS dla wejść z narastającymi zboczami, wypływa z bramki NAND przez końcówkę zasilania. Z drugiej strony do bramki wpływa prąd ładowania pojemności dren-podłoże tranzystorów PMOS. Tak więc w tabeli 1.4 możliwe są ujemne wartości, ale występuje tu silna zależność od topografii bramki. Obszar drenów dla obydwu tranzystorów PMOS w 2-wejściowej bramce NAND został zaprojektowany jako wspólny (rys. 1.18), zatem pojemność złączowa C_{DSp} nie jest zbyt duża i w tabeli 1.4 obserwujemy ujemne wartości pojemności C_{Lint} w przypadkach sterowania o numerach 5 i 7. Dla 3-wejściowej bramki NAND ujemną wartość otrzymano tylko w przypadku trzech zboczy narastających na wejściach, ponieważ w pozostałych przypadkach prąd wpływający do bramki był większy niż wypływający z powodu znacznej pojemności złączowej dren-podłoże tranzystorów PMOS (por. rys. 1.18).

Ujemne wartości wewnętrznej pojemności C_{Lint} w przypadkach sterowania bez zmiany stanu na wyjściu zostały już wyjaśnione wcześniej (rys. 1.19).

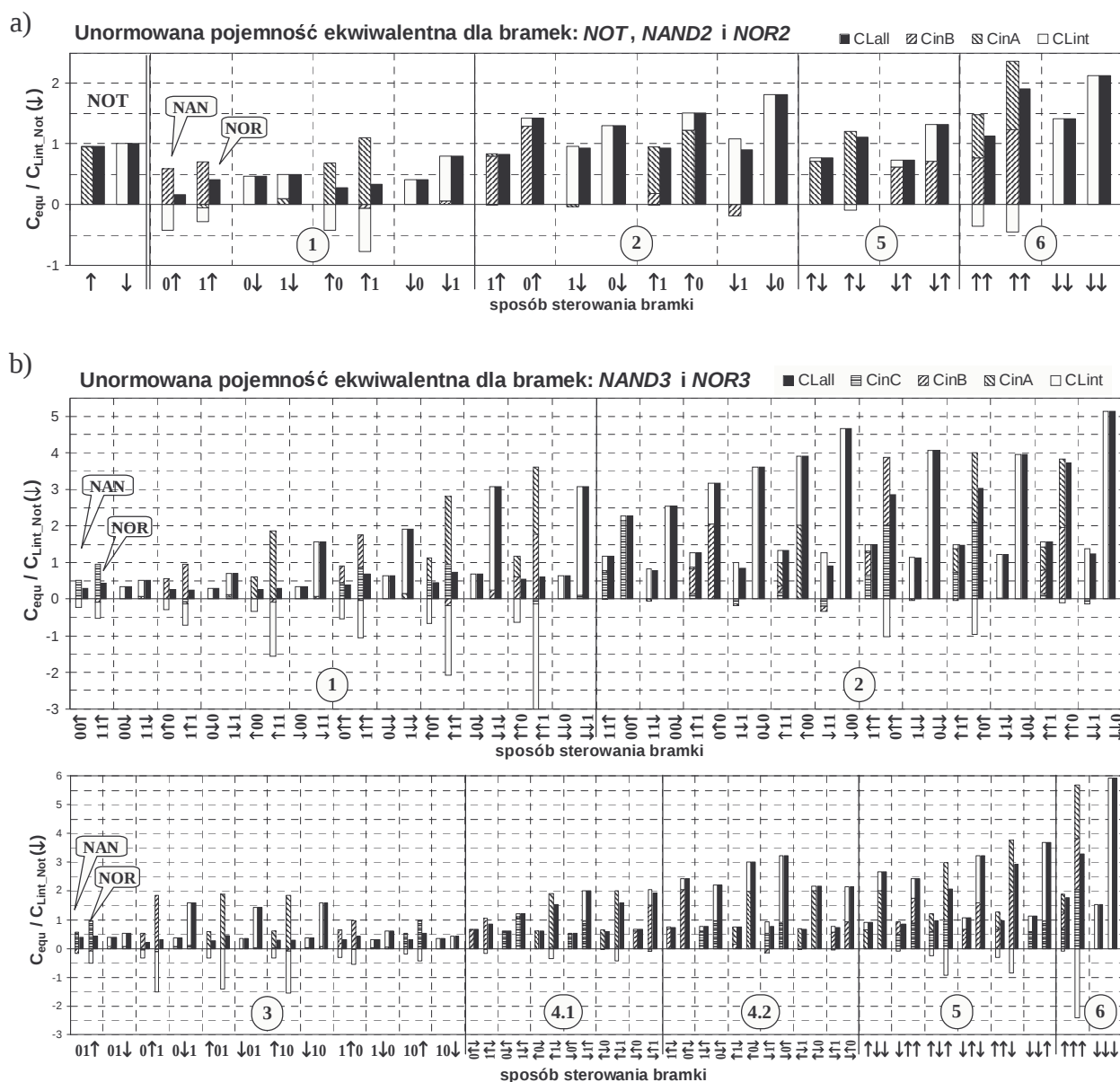
Ekwiwalentna pojemność wejściowa jest największa w przypadkach narastającego napięcia wejściowego, gdyż następuje ładowanie wypadkowej pojemności wejściowej bramki. Gdy na wejściu jest zbocze opadające, jak już wyjaśniano, założono zerową wartość pojemności ekwiwalentnej.

W przypadku sterowania wejść przemiennie ($\uparrow\downarrow$ lub $\downarrow\uparrow$) zastosowano symetryczne kształty przebiegów. W rzeczywistych układach zwykle przebiegi wejściowe są asymetryczne i wartość ekwiwalentnej pojemności może być nieco inna. Jednakże wyniki przeprowadzonych symulacji pokazują mały wpływ asymetrii sygnałów wejściowych na wartości wewnętrznej i wejściowej pojemności ekwiwalentnej.

Uogólniając, zastosowana w pracy metoda oceny własności energetycznych bramek CMOS pozwala na wyznaczenie parametrów dla nowego modelu poboru mocy. Metoda bazuje na pomiarze średniego prądu, który jest efektem zmiany wektora na wejściu bramki. Innymi słowy, pomiarze prądu zależnego od sposobu sterowania bramki. Na podstawie wartości prądu obliczamy pojemność ekwiwalentną według wzoru (1.18). Możliwe są zarówno dodatnie jak i ujemne wartości pojemności. Jeśli bramkę potraktujemy jako sieć kondensatorów z tranzystorami jako wyłącznikami, to podanie nowego wektora na wejście powoduje rekonfigurację takiego obwodu i możliwe jest, że prąd wypływa przez końcówki z bramki. W efekcie otrzymujemy ujemne wartości pojemności, które nie mają wpływu na przedstawione w dalszej części pracy metody estymacji i optymalizacji układów cyfrowych.

Na rysunku 1.22 zebrano wyniki oceny parametrów energetycznych bramek: NOT oraz 2- i 3-wejściowej NAND i NOR. Na wykresach przedstawiono wartości wewnętrznej C_{Lint} , wejściowej C_{inX} oraz będącej sumą całkowitej C_{Lall} pojemności ekwiwalentnej dla wszystkich sposobów sterowania bramek. Wartości pojemności unormowano do pojemności inwertera sterowanego zboczem opadającym. Odpowiednie wartości dla bramek NAND i NOR zostały umieszczone obok siebie z uwzględnieniem dualności bramek. Ponadto wyniki pogrupowano według sposobów sterowania. W danej grupie znalazły się przypadki dające taką samą odpowiedź bramki. Uwzględniając zatem stan wyjścia bramki, na rysunku 1.22 wydzielono następujące grupy sterowania:

- 1) bez zmiany na wyjściu,
- 2) ze zmianą na wyjściu,
- 3) brak zmiany na wyjściu, jedno zbocze na wejściach i stałe stany przeciwne,
 - 4.1) brak zmiany, lecz możliwa szpilka na wyjściu NOR,
 - 4.2) brak zmiany, lecz możliwa szpilka na wyjściu NAND,
- 5) brak zmiany, lecz możliwe szpilki na wyjściu NAND i NOR,
- 6) ze zmianą przy zgodnych zmianach na wszystkich wejściach.



Rys. 1.22. Wyniki oceny parametrów energetycznych (unormowane wartości poj. ekw.) dla bramek: (a) NOT, 2-wej. NAND i NOR, (b) 3-wej. NAND i NOR; technologia AMS 0,35 μ m

Porównanie własności bramek jest teraz nieco łatwiejsze. Dla bramek dwuwejściowych nie wszystkie grupy sterowania występują, a dla bramek o większej liczbie wejść można by wydzielić więcej charakterystycznych grup.

1.3.3. Parametry niezależne od sposobu sterowania bramki – model tradycyjny

Bazując na otrzymanych powyżej wynikach można wyliczyć pojemność ekwiwalentną dla tradycyjnego modelowania strat energii opartego na aktywności układu, które zostało opisane w punkcie 1.1. Jak już wspomniano, model ten wymaga jednej wartości pojemności węzłowej niezależnej od sposobu sterowania bramki, gdyż nie rozróżnia przyczyny przełączenia bramki a jedynie uwzględnia sam fakt przełączenia. Można by oczywiście szukać pojemność węzłową

obliczyć bazując na schemacie zastępczym otrzymanym z layout'u bramki. Jednakże wtedy nieliniowości pojemności pasożytniczych tranzystorów nie będą wzięte pod uwagę. Ponadto, jak można zauważyć, analizując wyniki z poprzedniego paragrafu, nawet w przypadku przełączenia bramki wartości pobieranej mocy nie są takie same, więc nie można założyć, że każde przełączenie powoduje przeładowanie takiej samej wartości pojemności. Poniżej zostanie przedstawiony algorytm pozwalający na obliczenie wartości pojemności węzłowych na podstawie wyników otrzymanych przy rozważeniu sposobów sterowania bramki. W ten sposób przynajmniej częściowo będzie uwzględniony wpływ sterowania bramki na wartość pobieranej mocy, co poprawi dokładność metody tradycyjnej.

W uproszczonej analizie zakłada się, że pobór energii przez bramkę CMOS następuje tylko wtedy, gdy jej napięcie wyjściowe jest narastające. Wyniki zebrane na rysunku 1.22 i w tabeli 1.4 pokazują, że w przypadku opadającego zbocza na wyjściu również następuje konsumpcja mocy z zasilania. Aby uwzględnić to zjawisko, zdecydowano, że wyjściowa pojemność węzłowa dla metody tradycyjnej będzie obliczana jako suma średnich wartości wyjściowej pojemności ekwiwalentnej dla zboczy opadających i narastających. Przypadki bez zmiany stanu wyjścia nie są brane pod uwagę, gdyż tradycyjny model uwzględnia tylko przełączenia. Jednakże analizy ścieżek prądów ładowania lub rozładowania (przedstawione w poprzednim punkcie) wskazują, że można te przypadki uwzględnić obliczając pojemności wejściowe.

Pojemność wejściowa dla modelu tradycyjnego będzie obliczana podobnie jak wyjściowa, jako średnia z wartości pojemności dla przypadków zboczy narastających na danym wejściu, ale z pewnym uwzględnieniem wewnętrznej pojemności wyjściowej (C_{Lint}).

Analizując rozptył prądów dla bramek NAND i NOR można zauważyć, że w przypadkach, gdy stan wyjścia nie zmienia się, to dla odwrotnych sposobów sterowania może przez końcówkę zasilania przepływać prąd w przeciwnych kierunkach o tych samych wartościach lub nie (rys. 1.19, rys. 1.20). Wtedy odpowiednie wartości pojemności ekwiwalentnych albo są równe, albo nie (tab. 1.4, rys. 1.22). Przez odwrotne sposoby sterowania rozumiemy takie przypadki, że na aktywnych wejściach zbocza są przeciwne a stany stałe pozostają bez zmian. Jeśli zatem wartości bezwzględne prądów są równe przy stałym stanie logicznym na wyjściu bramki, to mamy do czynienia z taką sytuacją, jak przedstawiono na rysunku 1.19. Sumując wartości prądu zasilania bramki NAND dla obu zboczy, otrzymujemy zero. Rozpatrując natomiast prąd wejściowy, widać, że jego wartość jest przeszacowana dla przypadku zbocza narastającego z powodu rozładowywania się pojemności tlenkowych tranzystora PMOS. Zatem w tej sytuacji możemy pominąć pojemność ekwiwalentną związaną z końcówką zasilania bramki NAND, gdyż

jest ona automatycznie uwzględniona przy obliczaniu pojemności wejściowej bramki. Patrząc z punktu widzenia końcówki zasilania można by powiedzieć, że w bramce nastąpiło tymczasowe przechowanie energii, chociaż w rzeczywistości każdy przepływ prądu przez rezystancje powoduje rozpraszanie energii. Równanie (1.18) zapewnia poprawne wartości pojemności ekwiwalentnej odzwierciedlające rzeczywisty pobór mocy z zasilania.

W przypadkach, gdy przy stałym stanie na wyjściu, dla odwrotnych sposobów sterowania wartości wewnętrznych pojemności ekwiwalentnych nie są równe, co do modułu, czyli pobrana i oddana przez bramkę energia nie bilansuje się, wartość ekwiwalentnej pojemności wyjściowej jest dzielona przez liczbę wejść ze zboczami narastającymi i dodawana do pojemności wejściowej tych wejść. Dopiero potem oblicza się średnią. Taka sytuacja oznacza, że w wyniku sekwencji danego sterowania i sterowania odwrotnego, została skonsumowana jakaś ilość energii z zasilania przez końcówkę zasilającą bramkę. Ponieważ stan wyjściowy bramki nie zmieniał się i ta energia nie byłaby uwzględniona w tradycyjnym modelu, a przyczyną tego była zmiana na wejściach, więc "obciążymy" za to wejścia.

Przypadki bez zmiany stanu na danym wejściu są pomijane. Analizując rozptyw prądów dla takich przypadków (rys. 1.20, rys. 1.21) widać, że konsumpcja mocy jest uwzględniana albo przez ekwiwalentną pojemność wyjściową, albo przez pojemność aktywnych, sąsiednich wejść.

Listing programu obliczającego ekwiwalentne pojemności dla tradycyjnego modelu, bazując na wynikach dla nowego modelu (uwzględniającego sposoby sterowania) i powyższych rozważaniach, napisanego w środowisku Matlab, umieszczono w załączniku. Tabela 1.5 zawiera wyniki obliczeń ekwiwalentnych pojemności węzłowych dla modelu tradycyjnego wyznaczone z wartości uwzględniających sposoby sterowania dla bramek NOT, 2-, 3- i 4-wejściowych NAND i NOR zaprojektowanych w technologii AMS 0,35 μ m

Tabela 1.5. Ekwiwalentna pojemność dla tradycyjnego modelu poboru mocy

bramka	C_{Lint} [fF]	C_{InA} [fF]	C_{InB} [fF]	C_{InC} [fF]	C_{InD} [fF]
NOT	5,162	4,860	—	—	—
NAND2	5,238	3,753	3,680	—	—
NAND3	7,131	3,126	3,238	3,106	—
NAND4	8,048	2,931	2,874	2,904	2,864
NOR2	8,878	5,831	5,501	—	—
NOR3	20,767	9,501	8,490	8,131	—
NOR4	30,031	12,252	10,418	9,887	9,572

Przedstawiony sposób obliczeń pojemności węzłowych pozwala na częściowe uwzględnienie wpływu sposobu sterowania na pobór mocy przez bramkę CMOS. Jednakże dopiero nowy model, zaproponowany w pracy, dokładniej opisuje te zależności.

ROZDZIAŁ 2.

Estymacja strat energii w układach kombinacyjnych

W procesie estymacji mocy wymaga się odpowiedniej dokładności i szybkości, ponieważ najczęściej jest on etapem weryfikacji projektu. Estymacja mocy jest wielokrotnie wykonywana jako pętla sprzężenia zwrotnego i od jej wyników zależy kierunek dalszego rozwoju projektu. Niemniej jednak określenie poziomu poboru mocy może być również wykorzystane do oceny parametrów energetycznych układu już zaprojektowanego lub będącego w końcowym etapie projektowania. Wtedy celem jest właściwy dobór innych elementów systemu np. stosownego źródła zasilania, systemu chłodzenia, innych układów współpracujących, itd. W pierwszym przypadku ważniejsza jest szybkość oceny poboru mocy, ponieważ czasami trzeba rozważać wiele wariantów układu, chociaż dokładność może mieć znaczący wpływ na dobór parametrów w czasie projektowania i w efekcie końcowy wynik. W drugim przypadku szybkość estymacji mocy wydaje się mniej istotna, a ważniejsza jest raczej dokładność. Poza tym na efektywność projektowania ma wpływ zastosowanie właściwych narzędzi do oceny poboru mocy na różnych poziomach opisu układu, gdyż znalezienie rozwiązania optymalnego na poszczególnych etapach projektowania zapewni dobry efekt końcowy [Che00]. Spotykamy metody estymacji poboru mocy poczynając od opisu funkcjonalnego układu na wysokim poziomie abstrakcji [Buy05], [Gup00a], [Nem96], poprzez niższe poziomy, np. przesłań rejestrowych [Gup00b], [Lin05], lub na poziomie bramek [Dev92], [Din98], [Naj94], [Naj95], [Tur98], a nawet na poziomie tranzystorów [Hua96].

Z punktu widzenia stosowanych rozwiązań estymację mocy można podzielić na estymację bazującą na metodach symulacyjnych i probabilistycznych. Metody symulacyjne mogą bardzo dobrze modelować zależności przestrzenne i czasowe między sygnałami wejściowymi i wewnątrz układu, dając bardzo dokładny wynik estymacji. Jednakże wymagają dostarczenia ciągu wektorów testowych, który właściwie odzwierciedla warunki pracy układu. Taki ciąg wektorów może być bardzo długi, zwłaszcza, gdy warunki pracy układu zmieniają się w szerokim zakresie, a to powoduje znaczne wydłużenie czasu estymacji mocy. Również skonstruowanie takiego ciągu wektorów może nie być łatwe. Obok prostych metod symulacyjnych [Kan86] często stosowaną metodą jest metoda Monte Carlo polegająca na losowym generowaniu wektorów testowych i symulacji pracy układu aż zostanie osiągnięty założony poziom dokładności [Fis96], [Bur93]. Poprzez dobór odpowiedniego kryterium

zakończenia symulacji można przyspieszyć proces estymacji mocy [Hil96], [Yua98]. Innym sposobem przyspieszenia tego typu estymacji jest takie skrócenie ciągu wektorów wejściowych, aby bez straty na dokładności odzwierciedlał zależności pomiędzy wejściowymi sygnałami, dając akceptowalny poziom błędu [Liu04], [Mar00], [Mar97]. Spotyka się też zastosowanie różnych teorii z dziedziny statystyki, np. teoria ekstremalnych wartości została wykorzystana w [Evm02] lub teoria entropii w [Mar96]. W pracy [Din00] przedstawiono sposoby poprawy efektywności metody Monte Carlo polegające na podziale układu na mniejsze bloki, które są próbkowane w odpowiedni sposób, aby wyznaczyć charakterystyczne wartości mocy, a następnie na podstawie tych wyników określa się pobór mocy dla całego układu. To pozwala na znaczne skrócenie czasu testowania układu.

Probabilistyczne metody estymacji mocy strat w układach cyfrowych polegają na zastąpieniu wielokrotnych symulacji układu dla kolejnych wektorów wejściowych i uśredniania wyników prądu przez obliczenie najpierw miary probabilistycznej określającej aktywność sygnału wejściowego a potem użycie jej do estymacji mocy. Rysunek 2.1 zaczerpnięty z [Naj94] przedstawia ideę estymacji bazującej na metodach symulacyjnych i probabilistycznych. Innymi słowy w metodzie probabilistycznej zamiast ciągu wektorów wejściowych trzeba określić cechy sygnału wejściowego używając pojęć probabilistycznych i zamiast wielokrotnych cykli symulacji wystarczy jeden cykl obliczeń układu, aby oszacować konsumpcję mocy. Niewątpliwą cechą pozytywną metod probabilistycznych jest szybkość, jednak wiele z nich nie uwzględnia zależności czasowych i przestrzennych w układzie [Cir87], [Dev90], [Dev92]. Niektóre prace uwzględniają opóźnienia bramek lub wewnętrzne korelacje między sygnałami [Gho92], [Mon97], [The01], [Tsu93]. Wiele prac dotyczy właściwego modelowania zależności czasowych między wektorami wejściowymi [Fre01], [Fre99], [Mar95], [Mar98].

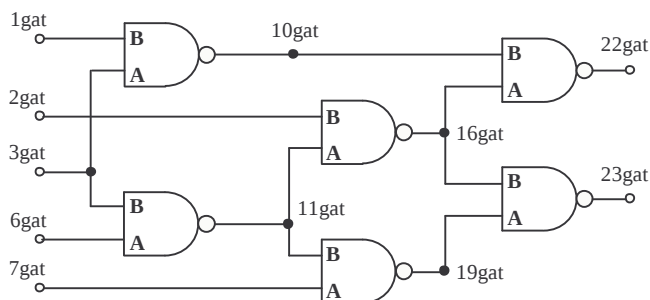
Rys. 2.1. Alternatywne drogi estymacji mocy [Naj94]

Estymację mocy z wykorzystaniem zaproponowanego nowego modelu strat energii można zaliczyć do grupy metod symulacyjnych, gdyż wykonywana jest symulacja logiczna układu w celu określenia sposobów sterowania bramek dla zadanych zmian wektorów wejściowych. Jednakże poprzez określenie prawdopodobieństwa zmian wektorów nie trzeba wykonywać długich symulacji dla ciągu wektorów testowych, wystarczy jeden raz rozważyć każdą zmianę. Nie ma też problemu właściwego doboru wektorów wejściowych, a otrzymany wynik nie zależy od ich kolejności. Zatem zaproponowana metoda estymacji ma cechy metod probabilistycznych.

Poniżej zostaną przedstawione wyniki estymacji mocy dla układu testowego *C17* [Brg85] z wykorzystaniem nowego modelu poboru mocy. Następnie porównano wyniki estymacji mocy z modelem nowym i tradycyjnym dla kilkunastu układów kombinacyjnych.

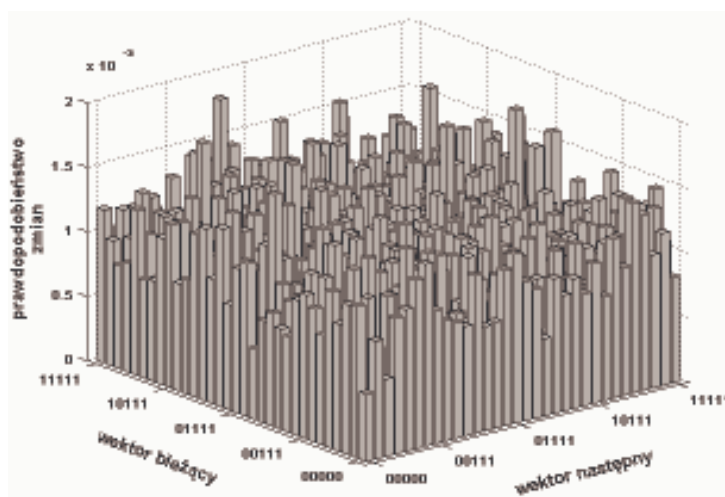
2.1. Zastosowanie nowego modelu w estymacji mocy na przykładzie układu *C17*

Opisany w rozdziale pierwszym nowy model strat energii oraz wyniki oceny własności energetycznych bramek zostały zastosowane do estymacji średniej mocy strat w testowym układzie *C17* przedstawionym na International Symposium on Circuits and Systems (ISCAS'85) w maju 1985 [Brg85]. Układ *C17* jest zbudowany z sześciu bramek NAND (rys. 2.2). Ponieważ ze względu na nowy model ważna jest kolejność wejść do bramek, na poniższym rysunku zaznaczono wejścia A i B według wcześniej przyjętych zasad, tj. wejście A to bramka tranzystora NMOS najbliższego do masy.



Rys. 2.2. Schemat ideowy układu testowego *C17*

Stosując symulacje logiczne określono prawdopodobieństwo sposobów sterowania dla wszystkich węzłów w układzie przy założeniu równomiernego oraz nierównomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych (rys. 2.3). Kwadratowa tabela z rysunku 2.3 zawiera losowo wygenerowane wartości prawdopodobieństwa przejść pomiędzy wszystkimi możliwymi wektorami dla dowolnego układu 5-wejściowego. Suma tych wartości wynosi jeden. Ta tabela jest podobna do tabeli z rysunku 1.12 i może być traktowana jako rozkład prawdopodobieństwa sposobów sterowania układu. Dla każdego węzła w układzie powstaje



Rys. 2.3. Losowy rozkład prawdopodobieństwa zmian wejściowych wektorów 5-bitowych

tabela o rozmiarach zależnych od liczby wejść do bramki sterującej dany węzeł. Wartość prawdopodobieństwa danego sposobu sterowania bramki jest sumą wartości prawdopodobieństw tych sposobów sterowania układu, które wywołały dany sposób sterowania węzła. Dla wektora 5-bitowego mamy możliwych 2^{10} zmian, w tym 32 zmiany między tymi samymi wektorami, więc zgodnie z równaniem (1.14) daje to 993 sposoby sterowania. Załóżmy dla rozkładu równomiernego, że każda zmiana wystąpi tylko raz, więc jest 1024 możliwe zmiany, licząc również zmiany między tymi samymi wektorami. Dla równomiernego rozkładu sposobów sterowania, powiemy, że prawdopodobieństwo wystąpienia każdej ze zmian jest równe $\frac{1}{1024}$. Zatem stosowna tabela określająca prawdopodobieństwo sposobów sterowania 5-wejściowego układu będzie wypełniona liczbami $\frac{1}{1024}$. W czasie symulacji logicznej układu dla każdej zmiany wektorów wejściowych określamy zmianę wektorów bramek i sumujemy stosowne wartości prawdopodobieństw. W przykładowym układzie węzły *10gat* i *11gat* będą miały równomierny rozkład prawdopodobieństwa sterowania o wartości 0,0625, ponieważ wejścia bramek są wejściami układu. Natomiast pozostałe węzły nie mają równomiernego rozkładu. Dla przykładu w tabeli 2.1 przedstawiono wartości prawdopodobieństw sterowania węzłów *16gat* i *22gat*.

Tabela 2.1. Wartości prawdopodobieństw sterowania wybranych węzłów w układzie C17

16gat					22gat				
V_{n+1}	00	01	10	11	V_{n+1}	00	01	10	11
V_n					V_n				
00	0,0156	0,0156	0,0469	0,0469	00	0,0039	0,0195	0,0117	0,0273
01	0,0156	0,0156	0,0469	0,0469	01	0,0195	0,0977	0,0586	0,1367
10	0,0469	0,0469	0,1406	0,1406	10	0,0117	0,0586	0,0352	0,0820
11	0,0469	0,0469	0,1406	0,1406	11	0,0273	0,1367	0,0820	0,1914

Otrzymane w ten sposób wartości prawdopodobieństwa są współczynnikami określającymi udział pojemności ekwiwalentnej związanej z danym sposobem sterowania w całkowitej

pojemności układu. Zatem stosując równanie (1.17) zostały obliczone węzłowe pojemności ekwiwalentne dla równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych i dla rozkładu z rysunku 2.3. Wyniki zebrano w tabeli 2.2.

Tabela 2.2. Ekwiwalentna pojemność węzłowa $C_{Ln_{eqv}}$ [fF] dla wszystkich węzłów w układzie C17

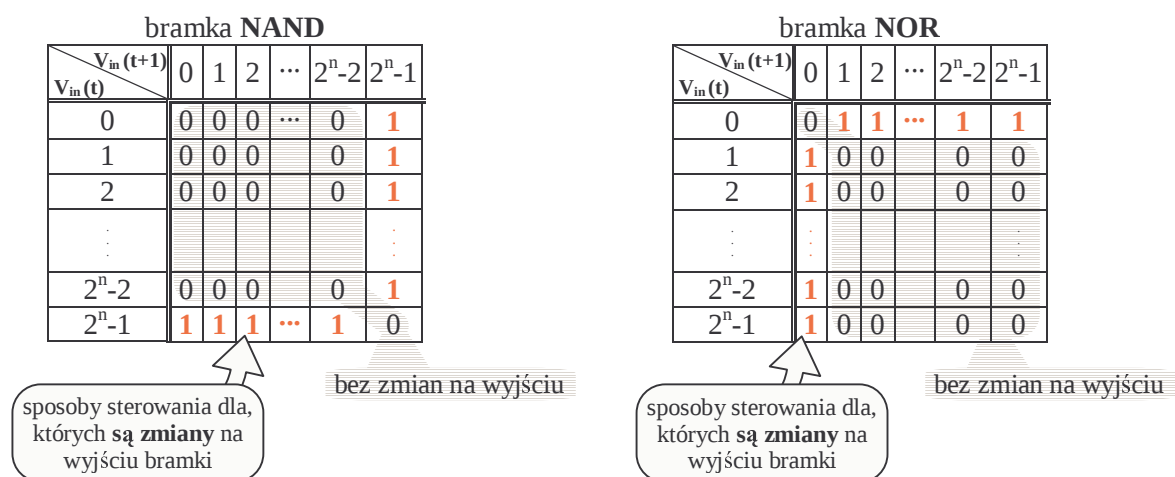
rozkład wej.	1gat	2gat	3gat	6gat	7gat	10gat	11gat	16gat	19gat	22gat	23gat	suma
równomierny	0,887	0,955	1,807	0,920	0,944	1,737	2,406	2,995	2,172	1,348	1,327	17,498
losowy	0,896	0,965	1,839	0,949	0,966	1,698	2,339	2,994	2,168	1,392	1,351	17,557

Powyższe wyniki estymacji pojemności ekwiwalentnej pozwalają na obliczenie rzeczywistej konsumpcji mocy P_{node_tot} w dowolnym węźle układu dla zadanej częstotliwości f zmian wektorów wejściowych według następującego równania:

$$P_{node_tot} = f C_{Ln_{eqv}} V_{dd}^2 \quad (2.1)$$

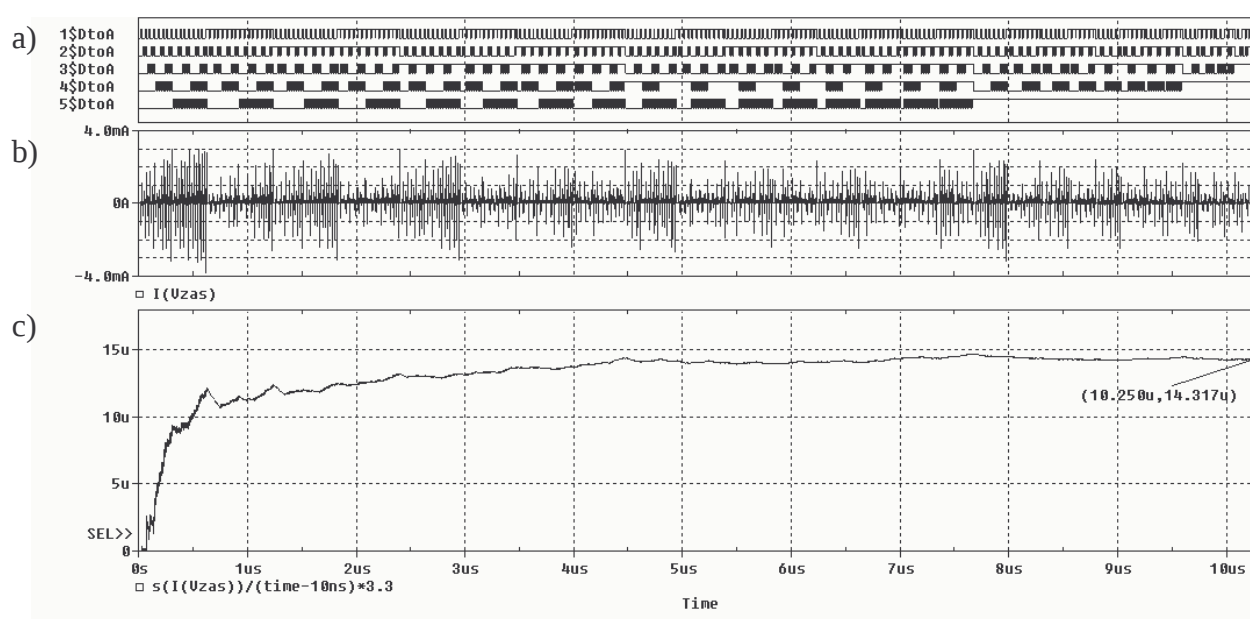
Uwzględniono również węzły wejściowe, a całkowitą moc pobieraną przez układ można obliczyć sumując moce węzłowe.

W celu oceny dokładności estymacji mocy wykorzystującej nowy model strat energii porównano powyższe wyniki z wynikami estymacji z modelem tradycyjnym oraz z wynikami symulacji przy użyciu symulatora układów elektronicznych PSPICE. Dla tradycyjnego modelu wykorzystano pojemności węzłowe obliczone według zasad przedstawionych w punkcie 1.3.3 i zebrane w tabeli 1.5. Korzystając z obliczonych powyżej prawdopodobieństw sterowania obliczono aktywność przełączeniową węzłów. Prawdopodobieństwo przełączenia węzła obliczono w ten sposób, że wzięto pod uwagę te sposoby sterowania bramek, które powodują zmianę stanu w węźle. Na rysunku 2.4 zaznaczono jedynekami sposoby sterowania bramek NAND i NOR powodujące zmianę stanu na wyjściu. Mnożąc elementy tych tabel z elementami tabel określających prawdopodobieństwo sterowania węzłów na zasadzie „każdy z każdym” i



Rys. 2.4. Sposoby sterowania bramek NAND i NOR powodujące zmiany stanu na wyjściach

następnie sumując, otrzymujemy prawdopodobieństwo przełączenia węzłów. Korzystając z równania (1.2) obliczono ekwiwalentną pojemność układu oraz średnią moc dla równomiernego rozkładu zmian wektorów wejściowych. W załączniku zamieszczono wydruki procedur użytych do obliczania aktywności przełączeniowej układu oraz estymacji mocy. Zakładając częstotliwość zmian wektorów wejściowych równą 100MHz otrzymujemy średni pobór mocy równy $12,745\mu\text{W}$ w przypadku modelu tradycyjnego i $13,051\mu\text{W}$ dla nowego modelu strat energii. Wartość średniej mocy strat uzyskana z symulacji wyniosła $14,317\mu\text{W}$ (rys. 2.5). Pomimo tego, że obliczając pojemności węzłowe dla modelu tradycyjnego częściowo uwzględniono wpływ sposobów sterowania, to wynik jest nieco gorszy. Zatem można stwierdzić, że nowy model pozwala na dokładniejszą estymację strat energii w cyfrowych układach CMOS.



Rys. 2.5. Wyniki symulacji poboru mocy przez układ C17: (a) ciąg wektorów wejściowych, (b) przebieg prądu zasilania, (c) średnia moc pobrana przez układ

Symulacje układu były przeprowadzone dla równomiernego rozkładu zmian wektorów wejściowych. Wygenerowano odpowiedni ciąg wektorów uwzględniający wszystkie możliwe zmiany (rys. 2.5a), następnie z przebiegu prądu (rys. 2.5b) obliczono średnią moc jako całkę z wartości prądu dzielną przez czas symulacji (rys. 2.5c).

2.2. Wyniki estymacji mocy z zastosowaniem modelu tradycyjnego i nowego

Dla kilkunastu układów ze zbioru układów testowych MCNC [Lis88] dokonano estymacji mocy wykorzystując tradycyjny i nowy model strat energii oraz dla porównania wykonano symulacje w programie PSPICE [Brz04a]. Układy testowe są opisane przez funkcje logiczne jako zbiory wektorów wejściowych i wyjściowych. Dlatego wymagają najpierw syntezy i mapowania technologicznego z określoną biblioteką bramek. Do syntezy zastosowano program

SIS oraz przygotowane skrypty *script.algebraic* i *script.rugged* [Sen92]. Listę poleceń programu SIS składającą się na użyte skrypty oraz opis biblioteki umieszczono w załączniku C. Biblioteka bramek składała się z siedmiu bramek: NOT, 2-, 3-, i 4-wejściowy NAND i NOR. Ponieważ symulacje wykonano w taki sam sposób jak poprzednio dla układu C17 z równomiernym rozkładem prawdopodobieństwa zmian, więc również estymacje przeprowadzono dla równomiernego rozkładu. Podobnie częstotliwość zmian wektorów wejściowych wynosiła 100MHz. W tabeli 2.3 zebrano wyniki estymacji oraz symulacji średniej mocy w układach. Obliczono różnicę wyrażoną w procentach oraz porównano między sobą wyniki estymacji z zastosowaniem nowego i tradycyjnego modelu. Dodatkowo podano informacje o strukturze układów, tj. liczbę: wejść, wyjść, poziomów oraz bramek.

Tabela 2.3. Wyniki estymacji i symulacji średniej mocy strat w układach testowych – porównanie

nazwa układu	średnia moc[μ W]			błąd [%]			liczba:			
	SPICE	tradyc.	nowy	tradyc.	nowy	nowy a trad.	we	wy	poz.	bra.
dk27_alg	47,64	40,35	41,52	-15,30	-12,85	2,46	4	5	5	21
dk27_rug	63,53	46,93	48,37	-26,13	-23,86	2,27	4	5	12	23
lion_alg	51,79	43,04	45,72	-16,90	-11,72	5,17	4	3	5	17
lion_rug	42,42	36,87	38,32	-13,08	-9,67	3,42	4	3	7	15
shift_alg	63,43	55,51	56,37	-12,49	-11,13	1,36	4	4	5	24
shift_rug	74,070	57,93	59,18	-21,79	-20,10	1,69	4	4	11	24
tra04_alg	46,40	39,64	40,42	-14,57	-12,89	1,68	4	3	7	18
tra04_rug	47,61	38,75	41,05	-18,61	-13,78	4,83	4	3	6	14
bbtas_alg	92,90	74,91	76,73	-19,36	-17,41	1,96	5	5	8	31
bbtas_rug	90,68	69,56	72,40	-23,29	-20,16	3,13	5	5	9	27
bw_alg	521,20	363,94	335,71	-34,03	-31,75	2,28	5	28	9	145
bw_rug	643,26	384,71	390,05	-41,73	-39,36	2,37	5	28	27	126
dk15_alg	171,17	135,64	143,67	-20,76	-16,07	4,69	5	7	7	54
dk15_rug	212,34	142,44	146,62	-32,92	-30,95	1,97	5	7	19	51
dk17_alg	122,65	100,25	103,80	-18,26	-15,37	2,89	5	6	6	47
dk17_rug	192,40	126,76	136,67	-34,12	-28,97	5,15	5	6	18	44
dk512_alg	238,76	179,78	186,54	-24,70	-21,87	2,83	5	7	7	59
dk512_rug	210,84	142,01	146,23	-32,65	-30,64	2,00	5	7	17	56
mod12_alg	93,73	77,62	79,77	-17,19	-14,89	2,29	5	5	7	36
mod12_rug	210,84	128,66	133,62	-38,98	-36,62	2,35	5	5	12	38
rd53_alg	159,32	136,57	126,41	-22,76	-20,66	2,10	5	3	8	48
rd53_rug	132,34	95,29	102,89	-26,24	-22,25	3,99	5	3	12	33
squar5_alg	175,08	155,02	143,59	-21,07	-17,99	3,08	5	8	7	52
squar5_rug	222,47	149,21	155,43	-32,93	-30,13	2,80	5	8	22	53
xor5_alg	74,01	42,96	43,74	-41,95	-40,90	1,05	5	1	12	18
xor5_rug	70,60	39,05	40,14	-44,69	-43,14	1,54	5	1	12	16
con1_alg	53,08	47,28	48,07	-10,93	-9,44	1,49	7	2	6	18
con1_rug	69,39	55,21	59,79	-20,44	-13,83	6,60	7	2	5	21

Analizując wyniki zebrane w powyższej tabeli można powiedzieć, że zastosowanie nowego modelu strat energii w układach CMOS do estymacji mocy strat daje wyniki o kilka procent bliższe rzeczywistości. Można przypuszczać, że różnice tylko kilkuprocentowe pomiędzy wynikami estymacji są efektem uwzględnienia sposobów sterowania podczas obliczania pojemności węzłowych dla modelu tradycyjnego (punkt 1.3.3). Natomiast duże różnice wyników estymacji w stosunku do symulacji wynikają z nieuwzględnienia opóźnień bramek i z tego wynikających szpilek oraz hazardów w układzie. Pomimo że w symulacjach stosowano sygnały o odpowiednich czasach trwania zboczy, to jednak w większych układach parametry czasowe sygnałów mogły się pogorszyć, powodując wystąpienie quasi-zwarciovych strat mocy, dodatkowo powiększając wyniki symulacji, a ten efekt również nie był uwzględniony w estymacji.

Doświadczenia z oceną własności energetycznych układów cyfrowych CMOS poprzez estymację i symulację nasuwają następujące spostrzeżenia. Ponieważ, jak opisywano wcześniej, symulacyjne metody estymacji mocy mają wiele wad, a probabilistyczne są mniej dokładne, to zaproponowaną metodę oceny parametrów energetycznych bramek można by zaadaptować dla całych układów. Jeśli wykonamy symulacje układu dla wszystkich możliwych zmian wektorów wejściowych, to możemy analogicznie do bramek obliczyć ich pojemność ekwiwalentną. Ponadto w takich symulacjach będą uwzględnione opóźnienia bramek i związany z tym pobór energii. W ten sposób określimy pewne porcje konsumowanej energii dla każdej zmiany wektorów. Następnie, korzystając ze statystycznej charakterystyki strumienia wejściowego danych do układu, możemy bardzo dokładnie obliczyć konsumpcję mocy. Wystarczy znajomość prawdopodobieństwa zmian wektorów. Jednakże takie rozwiązanie jest dobre tylko dla małych układów ze względu na czas symulacji. Podczas symulacji, których wyniki zebrano w tabeli 2.3, potrzeba było nawet kilku godzin dla większych układów przy zastosowaniu komputera klasy IBM z procesorem Pentium 4 2,4GHz. Na domiar złego, w procesie projektowania układu potrzeba czasami wielokrotnie ocenić pobór mocy przez układ.

ROZDZIAŁ 3.

Optymalizacja układów cyfrowych z zastosowaniem nowego modelu strat energii

Celem optymalizacji jest ogólna poprawa jakości układu. W znaczeniu ogólnym optymalizacja może polegać na maksymalizacji lub minimalizacji pewnej określonej wielkości fizycznej, technologicznej lub ekonomicznej. Matematyczny model optymalizacji formułuje zadanie poprawy określonej wielkości charakterystycznej, zwanej kryterium optymalizacji, wskaźnikiem jakości lub funkcją celu, przy odpowiednich ograniczeniach. Te ostatnie określają zazwyczaj dopuszczalne zakresy zmian parametrów technicznych lub zasobów. Optymalizacja polega zatem na znalezieniu takich warunków pracy systemu lub układu, w których wartość określonego wskaźnika jakości będzie ekstremalna a jednocześnie parametry pracy układu nie przekroczą narzuconych na nie ograniczeń.

W przypadku układów cyfrowych zwykle optymalizacja skupiała się na minimalizacji powierzchni oraz zwiększaniu szybkości pracy poprzez poprawę parametrów czasowych układu. Często, zwłaszcza w przypadku układów scalonych, celem optymalizacji była poprawa łatwości testowania, co skutkowało zwiększeniem uzysku produkcyjnego i poprawą niezawodności pracy. W ostatnich kilkunastu latach, gdy w coraz bardziej rozbudowywanych i zaawansowanych technologicznie układach scalonych problemy termiczne zaczęły odgrywać znaczącą rolę, celem optymalizacji stały się parametry energetyczne układów. Bardzo szybki rozwój technologii mikroelektronicznych jak również bezprzewodowych i ciągle podnoszenie walorów użytkowych produkowanych urządzeń wymogło rozwój metod projektowania układów oszczędnych energetycznie.

Jedną z metod optymalizacji kombinacyjnych układów logicznych CMOS jest metoda, polegająca na ustaleniu takiej kolejności wejść do bramek, aby w efekcie minimalizować pobór mocy przez układ. Uwzględniając pojemność pasożytniczą węzłów wewnętrznych, powstałych między szeregowo połączonymi tranzystorami w bramce oraz wykorzystując tradycyjne miary aktywności układów, autorzy wielu prac proponują różne techniki determinujące najlepszą kolejność podłączenia sygnałów do wejść bramki. O ile obliczenie pojemności węzła wewnętrznego nie sprawia większego problemu, gdyż jest to suma pojemności złączowej obszaru drenu i źródła dwóch sąsiednich tranzystorów, to obliczenie aktywności przełączeniowej tego węzła jest problemem. Jak już wspomniano w punkcie 1.1.2, przełączenie węzła

wewnętrznego może pojawić się nie tylko wtedy, gdy zmienia się stan wyjścia bramki, ale w wyniku dowolnej zmiany sygnałów wejściowych. W związku z tym stosuje się różne techniki pozwalające określić aktywność węzłów wewnętrznych na podstawie prawdopodobieństwa lub gęstości przełączeń sygnałów wejściowych [Asa95], [Has99], [Hos96], [Pra96], [She95].

W pracy [Pra96] zastosowano metodę „wyczerpującego wyliczania” w celu znalezienia optymalnej kolejności wejść do bramki. Obliczono pobór mocy dla każdej z możliwych kolejności wejść. Autorzy zastosowali heurystyczną metodę, w której dla każdego potencjalnego sygnału wejściowego najpierw oblicza się prawdopodobieństwo zdarzenia, polegającego na tym, że wystąpi zmiana sygnału i pozostałe sygnały wejściowe bramki są w stanie „włączenia” („1” dla tranzystorów NMOS, „0” dla PMOS). Następnie sygnał z tak obliczonym, największym prawdopodobieństwem jest podłączany na wejście najbliższe wyjściu. Autorzy pracy [She95] zamiast dokładnie obliczać aktywność węzłów wewnętrznych, bazując na ogólnych analizach, zaproponowali zestaw reguł określających, do których wejść bramek prostych (NAND i NOR) oraz złożonych (AOI, OAI) podłączać sygnały opisane przez prawdopodobieństwo jedynki i gęstość przejść. W pracy [Asa95], jak wspomniano w punkcie 1.1.2, do opisu aktywności bramki i również węzłów wewnętrznych wykorzystano graf stanów definiując prawdopodobieństwo zera i jedynki oraz przejść między nimi. Na tej podstawie wyprowadzono stosowane równania opisujące prawdopodobieństwo ładowania i rozładowania węzła wewnętrznego i podano formułę opisującą pobór mocy przez bramkę. Jest to praktycznie model klasyczny w oparciu, o który zdefiniowano kryterium rozstrzygające o lepszej kolejności wejść. Rozważania przeprowadzono dla 2-wejściowej bramki NAND a następnie rozszerzono na bramki wielowejściowe. Bazując na prawdopodobieństwie sygnału wejściowego i pojęciu prawdopodobieństwa warunkowego, autorzy pracy [Hos96] określili warunki połączenia węzła wewnętrznego z zasilaniem oraz masą i prawdopodobieństwo stanu „0” lub „1”. Na tej podstawie wyprowadzili równania definiujące oczekiwaną aktywność przełączeniową węzłów wewnętrznych i pobór mocy przez bramkę. Zadanie optymalizacyjne zdefiniowano jako minimalizację równań opisujących pobór mocy w poszczególnych węzłach wewnętrznych. Zaproponowano heurystyczną technikę zamiany wejść tak, aby znaleźć rozwiązanie najbliższe optymalnemu przy możliwie najmniejszej liczbie iteracji algorytmu. Pominięty tu przypadek odizolowania węzła uwzględnili autorzy w pracy [Has99] (por. punkt 1.1.2). Analizując grafy bramek podano warunki logiczne połączenia węzła wewnętrznego z masą, zasilaniem lub odizolowania. Równania opisujące pobór mocy określono wykorzystując te zależności logiczne i gęstość przełączeń oraz prawdopodobieństwo sygnału. Poszukiwanie najlepszej kolejności wejść do bramki polega na obliczeniu konsumpcji mocy dla wszystkich możliwych permutacji wejść i wybraniu kolejności o najmniejszej wartości mocy.

W pracy [Bor95] bazując na wynikach symulacji zaobserwowano, że jeśli sygnał krytyczny czasowo będzie podany na wejście tranzystora najbliższego wyjściu, to nastąpi poprawa parametrów czasowych i energetycznych bramki. W rzeczywistości autorzy zmniejszali wpływ czasu trwania zbocza sygnału wejściowego na quasi-zwarciove straty energii. Podobne wyniki uzyskano podczas testowania układów scalonych w pracy [Brz02]. Ponadto kolejność włączonych i wyłączonych tranzystorów w łańcuchu szeregowym ma wpływ na wartość prądu upływu, płynącego przez podbramkowy tlenek krzemu w tranzystorach [Sul04], a więc na straty statyczne.

W niniejszym rozdziale zostanie opisana metoda polegająca na znanej z literatury technice zamiany kolejności wejść do bramek logicznych, ale wykorzystująca nową uprzednio opisaną miarę aktywności układu – prawdopodobieństwo sposobu sterowania bramki.

3.1. Procedura optymalizacji

Przedstawione w drugim rozdziale wyniki estymacji mocy, chociaż dobrze odzwierciedlają całkowity pobór mocy, jak i straty w poszczególnych węzłach układu, to jednak nie mogą być bezpośrednio wykorzystane do optymalizacji układu prowadzącej do zmniejszenia konsumpcji energii. Analizując wyniki oceny parametrów bramek, przedstawione na rysunku 1.22 można zauważyć, że istnieje możliwość zmniejszenia strat mocy w układzie przez optymalny dobór kolejności wejść do bramek. Z punktu widzenia pracy bramki logicznej kolejność wejść nie ma znaczenia, natomiast ma istotny wpływ na jej pobór mocy i parametry czasowe. Opisanie powyżej metody optymalizacji układów bazują na zmianie kolejności wejść, ale wykorzystują tradycyjne miary aktywności sygnałów (prawdopodobieństwo przejść, gęstość przełączeń, itp.). Poniżej przedstawiono procedurę optymalizacji kombinacyjnych układów logicznych, zbudowanych z bramek CMOS, wykorzystującą nowy model strat mocy oraz wnioski wynikające z analizy wyników oceny parametrów energetycznych bramek. Zaproponowany model pozwala uwzględnić nie tylko pojemność węzłów wewnętrznych, ale również pojemności sprzęgające tranzystorów (C_{GD} , C_{GS} , C_{GB}) oraz komentowane w rozdziale pierwszym (p. 1.3.2) przypadki ujemnych prądów.

Celem zaproponowanej procedury optymalizacji jest minimalizacja średniej mocy pobranej przez układ cyfrowy w pewnym okresie czasu. Optymalizacja polega na poszukiwaniu lokalnych minimów iloczynów danego prawdopodobieństwa sterowania i odpowiedniej pojemności ekwiwalentnej. Efektem jest określenie najlepszej kolejności wejść do bramek w układzie. Układ logiczny musi być uprzednio zmapowany z określoną biblioteką bramek, dla których oszacowano parametry energetyczne zgodnie z zasadami przedstawionymi w punkcie 1.3 rozdziału pierwszego. Zatem proponowana optymalizacja może być włączona w proces projektowania jako ostatni

element. Poszukując optymalnej kolejności wejść do bramek trzeba przeanalizować wszystkie możliwe kombinacje kolejności wejść dla wszystkich bramek w układzie. Zmiana kolejności wejść jakiejś bramki w układzie skutkuje powstaniem nowego układu z punktu widzenia sposobów sterowania rozważanego węzła. Trzeba ponownie przeprowadzić analizę aktywności układu i obliczyć ekwiwalentne pojemności węzłowe. Takie podejście wymaga bardzo dużej liczby analiz układu i wielokrotnego obliczania prawdopodobieństwa sterowania węzłów i węzłowych pojemności ekwiwalentnych. Dlatego zaproponowano bardziej efektywny algorytm optymalizacji wymagający tylko jednokrotnego obliczenia prawdopodobieństwa sterowania węzłów oraz nieco zmodyfikowano sposób obliczania pojemności ekwiwalentnej związanej z węzłami w układzie.

Procedurę optymalizacji można podzielić na następujące etapy:

- 1) **analiza aktywności układu** – określenie prawdopodobieństwa sposobów sterowania węzłów dla określonego rozkładu prawdopodobieństwa zmian wektorów wejściowych; wynikiem są tabele aktywności z prawdopodobieństwem sposobów sterowania dla wszystkich węzłów w układzie,
- 2) **obliczenie nowych tabel aktywności węzłów** dla wszystkich możliwych kolejności wejść do bramek bazując tylko na wynikach otrzymanych w punkcie pierwszym,
- 3) **obliczenie całkowitej pojemności ekwiwalentnej** dla bramek w układzie i wszystkich możliwych kolejności ich wejść,
- 4) **wyбір najlepszej kolejności wejść do bramek** – najmniejsza wartość pojemności ekwiwalentnej wskazuje najlepszą kolejność (lokalne minima).

Poszczególne etapy powyższego algorytmu wymagają szczegółowego wyjaśnienia. W pierwszym punkcie dla wszystkich możliwych zmian wektorów wejściowych określa się sposoby sterowania węzłów w układzie. Można tu z powodzeniem zastosować technikę symulacji logicznych. Następnie dla zadanego rozkładu prawdopodobieństwa zmian wektorów wejściowych oblicza się prawdopodobieństwo wyznaczonych sposobów sterowania węzłów, sumując wartości prawdopodobieństw tych zmian wektorów, które wywołały dany sposób sterowania. W efekcie otrzymujemy kwadratowe tabele, takie jak otrzymano dla układu C17 (tabela 2.1). Nazwiemy je tabelami aktywności węzłów. Wymiary tych tabel zależą oczywiście od liczby wejść do bramki sterującej dany węzeł.

W drugim punkcie na podstawie tabel aktywności otrzymanych poprzednio tworzone są nowe tabele dla wszystkich możliwych kolejności wejść do bramek. Nowa kolejność dla danej bramki wymaga zamiany miejscami odpowiednich wartości z wyjściowej tabeli aktywności. Wektory aktualny i następny, opisujące tabelę, mogą być potraktowane jako indeksy. Jeśli

zamienimy kolejność wejść, to nastąpi zamiana pozycji bitów w wektorach i odpowiednia zmiana wartości indeksów. Teraz indeksy wskazują nowe miejsce w nowej tabeli. Przykładowo, jeśli wejściowy wektor ABC zmieni swą wartość z 101 na 111 , to mamy sposób sterowania oznaczony przez $(1\bar{1}1)$, a zamiana miejscami wejść A i B spowoduje, że otrzymamy nowy sposób sterowania $(\bar{1}11)$. W tej sytuacji prawdopodobieństwo wystąpienia nowego sposobu jest takie jak starego. Zatem, w nowej tabeli należy zamienić miejscami wartości prawdopodobieństw tych sposobów sterowania. Są to miejsca o indeksach $(6,7)$ i $(3,7)$. Należy zauważyć, że dla tych konkretnych wektorów zamiana miejscami wejść A i C nie spowoduje zmiany sposobu sterowania. Rysunek 3.1 wyjaśnia zasadę zamiany miejscami wartości w tabeli aktywności dla zamiany wejść A i B w przypadku bramki trójwejściowej. Stosując tę technikę zamiany miejscami wartości obliczonego w pierwszym punkcie prawdopodobieństwa, można łatwo i szybko utworzyć tabele prawdopodobieństw sposobów dla wszystkich możliwych kombinacji wejść do bramki sterującej dany węzeł.

ABC								
	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

BAC								
	0	1	4	5	2	3	6	7
0								
1								
4								
5								
2								
3								
6								
7								

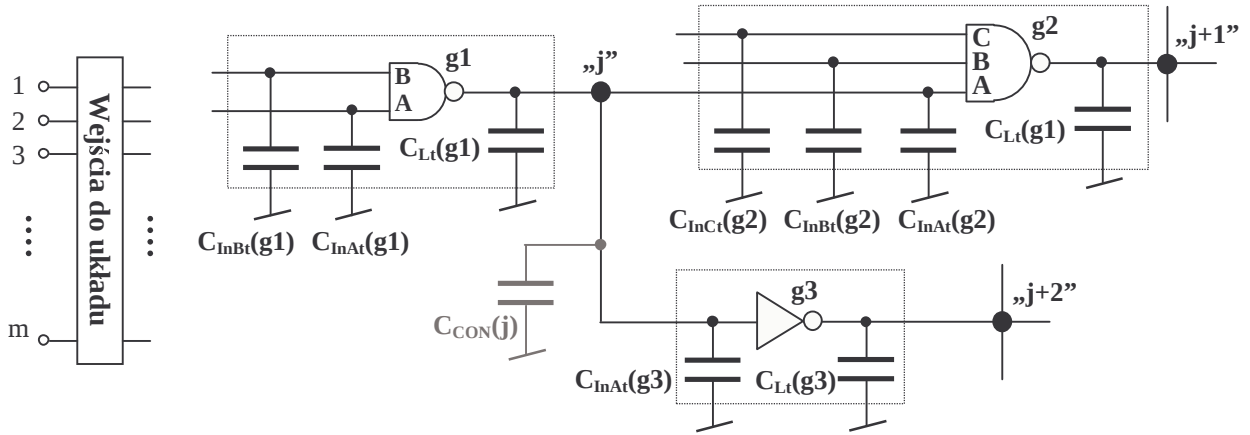
Zmiana kolejności
wejść A i B
powoduje, że trzeba
zamienić miejscami:
kolumny 2 z 4 oraz 3 z 5
i wiersze 2 z 4 oraz 3 z 5

Rys. 3.1. Zasada tworzenia nowej tabeli aktywności dla bramki 3-wej. – zamiana wejść A i B

W następnym etapie optymalizacji następuje obliczenie pojemności ekwiwalentnych dla wszystkich możliwych sposobów sterowania bramek na podstawie uprzednio otrzymanych tabel aktywności dla wszystkich węzłów w układzie. Na rysunku 3.2 przedstawiono, uzupełniony o wejściowe i wewnętrzne pojemności wszystkich bramek, przykładowy układ logiczny z rysunku 1.13. Całkowita ekwiwalentna pojemność, która jest sumą pojemności węzłowych C_{Lne} wszystkich trzech węzłów $j, j+1, j+2$ i pojemności wejściowych C_{InXt} bramki $g1$ oraz wejść B i C bramki $g2$, dla tego fragmentu układu wyraża się wzorem:

$$\begin{aligned}
 C_{tot} &= C_{Lne}(j) + C_{Lne}(j+1) + C_{Lne}(j+2) + C_{InAt}(g1) + C_{InBt}(g1) + C_{InBt}(g2) + C_{InCt}(g2) = \\
 &= C_{Lt}(g1) + C_{InAt}(g2) + C_{InAt}(g3) + C_{Lt}(g2) + C_{Lt}(g3) + C_{InAt}(g1) + C_{InBt}(g1) + C_{InBt}(g2) + C_{InCt}(g2)
 \end{aligned} \quad (3.1)$$

Oznaczone indeksem t całkowite pojemności wejściowe jak i wyjściowe bramek można zapisać w funkcji prawdopodobieństwa sposobów sterowania korzystając ze wzoru (1.17), który opisuje



Rys. 3.2. Fragment układu logicznego z pojemnościami ekwiwalentnymi

pojemność ekwiwalentną związaną z określoną końcówką bramki uwzględniając rozkład prawdopodobieństwa sposobów sterowania. Zatem powyższe równanie może być przepisane w postaci:

$$\begin{aligned}
 C_{tot} = & \sum_{dw_j} C_{L_{int_g1}}(dw_j) p(dw_j) + \sum_{dw_{j+1}} C_{InA_g2}(dw_{j+1}) p(dw_{j+1}) + \sum_{dw_{j+2}} C_{InA_g3}(dw_{j+2}) p(dw_{j+2}) + \\
 & + \sum_{dw_{j+1}} C_{L_{int_g2}}(dw_{j+1}) p(dw_{j+1}) + \sum_{dw_{j+2}} C_{L_{int_g3}}(dw_{j+2}) p(dw_{j+2}) + \sum_{dw_j} C_{InA_g1}(dw_j) p(dw_j) + \quad (3.2) \\
 & + \sum_{dw_j} C_{InB_g1}(dw_j) p(dw_j) + \sum_{dw_{j+1}} C_{InB_g2}(dw_{j+1}) p(dw_{j+1}) + \sum_{dw_{j+1}} C_{InC_g2}(dw_{j+1}) p(dw_{j+1})
 \end{aligned}$$

gdzie: dw_j , dw_{j+1} , dw_{j+2} oznaczają wszystkie sposoby sterowania węzłów j , $j+1$, $j+2$, a $p(dw_i)$ to prawdopodobieństwo określonego sposobu sterowania. Następnie grupując pojemności względem sposobów sterowania poszczególnych węzłów otrzymujemy następujące równanie:

$$\begin{aligned}
 C_{tot} = & \sum_{dw_j} (C_{L_{int_g1}}(dw_j) + C_{InA_g1}(dw_j) + C_{InB_g1}(dw_j)) p(dw_j) + \\
 & + \sum_{dw_{j+1}} (C_{L_{int_g2}}(dw_{j+1}) + C_{InA_g2}(dw_{j+1}) + C_{InB_g2}(dw_{j+1}) + C_{InC_g2}(dw_{j+1})) p(dw_{j+1}) + \quad (3.3) \\
 & + \sum_{dw_{j+2}} (C_{L_{int_g3}}(dw_{j+2}) + C_{InA_g3}(dw_{j+2})) p(dw_{j+2})
 \end{aligned}$$

W powyższych równaniach całkowita pojemność ekwiwalentna układu jest obliczana zgodnie ze wcześniej zdefiniowanym sposobem jako suma iloczynów prawdopodobieństwa sposobów sterowania i odpowiednich wartości pojemności ekwiwalentnej. Równania (3.1) i (3.2) dokładnie odzwierciedlają strukturę układu i rzeczywiste straty mocy w poszczególnych węzłach. Mogą też być wykorzystane do obliczenia strat energii w poszczególnych punktach układu. Jednakże jakakolwiek zmiana struktury układu, np. zamiana kolejności wejść do bramki, powoduje konieczność modyfikacji równań. Rozpatrując całkowitą średnią moc pobieraną przez cały układ, równanie (3.2) może być przekształcone, dając równanie (3.3). Teraz wszystkie wartości pojemności ekwiwalentnych związanych z bramką, sterującą dany węzeł, są mnożone

przez tę samą wartość prawdopodobieństwa sterowania. Zatem te pojemności mogą być zsumowane, dając całkowitą pojemność ekwiwalentną bramki C_{Lall_gi} . Równanie (3.2), opisujące całkowity pobór mocy przez układ z rysunku 3.2, może być teraz zapisane jako:

$$C_{tot} = \sum_{dw_j} C_{Lall_g1}(dw_j)p(dw_j) + \sum_{dw_{j+1}} C_{Lall_g2}(dw_{j+1})p(dw_{j+1}) + \sum_{dw_{j+2}} C_{Lall_g3}(dw_{j+2})p(dw_{j+2}) \quad (3.4)$$

Całkowita pojemność ekwiwalentna bramki jako suma wewnętrznej pojemności obciążającej oraz wejściowej pojemności ekwiwalentnej dla wszystkich wejść jest parametrem reprezentującym własności energetyczne bramki, niezależnie od układu, w którym bramka pracuje. Na rysunku 1.22 pojemność C_{Lall} została dodatkowo zaznaczona.

Wprowadzenie pojęcia całkowitej pojemności ekwiwalentnej bramki znacząco upraszcza wykonywane w trzecim punkcie procedury, obliczanie ekwiwalentnej pojemności dla wszystkich możliwych kolejności wejść do bramek w układzie. Zamiana wejść miejscami wpływa tylko na obliczanie tabel aktywności węzłów, a tabela pojemności ekwiwalentnych dla bramek jest taka sama. Również obliczanie, a właściwie tworzenie nowych tabel aktywności dla nowych kolejności wejść do bramek bez potrzeby analizy nowego układu powoduje, że proponowana metoda optymalizacji układu jest efektywna i możliwa do zastosowania nawet dla dużych układów.

Ostatnim etapem procedury jest wybranie najlepszej kolejności wejść do bramki. Ponieważ przedstawiony powyżej sposób obliczania całkowitej pojemności ekwiwalentnej układu pozwala na traktowanie bramek jako niezależnych elementów, to minimalna wartość tak obliczonej ekwiwalentnej pojemności bramki wyznacza najlepszą kolejność wejść. Zatem można powiedzieć, że w przedstawionej procedurze znalezienie lokalnego optimum poprawia parametry całego układu.

W tabeli 3.1 zamieszczono wyniki obliczeń ekwiwalentnej pojemności dla wszystkich możliwych kolejności wejść do bramek sterujących węzły w układzie C17 (rys. 2.2). Wszystkie bramki są dwuwejściowe, więc możliwe są tylko dwie kolejności wejść. Kolejność pierwsza jest taka jak na rysunku 2.2, zgodnie z opisem układu [Brg85]. Minimalne wartości pojemności dla

Tabela 3.1. Ekwiwalentna pojemność bramek sterujących węzły w układzie C17 C_{Lg_eqv} [fF] dla wszystkich możliwych kolejności wejść

rozkład prawdopodob. zmian wektorów wej.	kolejność wejść	węzły w układzie						suma
		10gat	11gat	16gat	19gat	22gat	23gat	
równomierny	1-sza	2,857	2,857	2,879	2,910	2,922	3,073	17,498
	2-ga	2,857	2,857	2,910	2,879	2,901	3,073	17,477
losowy	1-sza	2,876	2,867	2,877	2,917	2,930	3,090	17,557
	2-ga	2,871	2,869	2,911	2,881	2,904	3,088	17,524

poszczególnych węzłów wyznaczają najlepsze kolejności wejść, a ich suma jest minimalną wartością pojemności dla całego układu. Otrzymana tabela jest nieco inna niż tabela 2.2, gdyż przedstawiono tu całkowitą pojemność ekwiwalentną bramek a nie węzłów. Wartość minimalnej pojemności ekwiwalentnej dla całego układu to 17,45fF dla równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych i 17,49fF dla losowego.

3.2. Wyniki optymalizacji układów kombinacyjnych

Procedura optymalizacji opisana w poprzednim podrozdziale została zaimplementowana w środowisku Matlab. W załączniku A zamieszczono wydruk programu. W celu oceny efektywności zaproponowanego algorytmu wykonano optymalizację kilkunastu układów testowych MCNC. Podobnie jak w przypadku estymacji mocy (rozdział 2.2) układy zostały zsyntezowane za pomocą programu SIS i zmapowane z naszą biblioteką, która składała się z siedmiu bramek. Jako efekt optymalizacji przedstawiono wyniki estymacji minimalnej i maksymalnej pojemności ekwiwalentnej dla najlepszej i najgorszej kolejności wejść do bramek. Wyniki zamieszczono w tabelach 3.3 i 3.4 dla równomiernego i losowego rozkładu prawdopodobieństwa zmian wektorów wejściowych. Tabelę 3.3 podzielono, dla wygody, na dwie części. Pierwsza część (tab. 3.3.a) zawiera wyniki dla układów, których syntezy dokonano z użyciem skryptu *rugged*, a druga (tab. 3.3.b) dla układów zsyntezowanych za pomocą skryptu *algebraic*. W tabelach podano także dodatkowe informacje o strukturze układów tj. liczbę: wejść, wyjść, poziomów i węzłów (bramek). Ponieważ własności energetyczne układów zależą od rodzaju bramek dodatkowo w tabelach wpisano również liczby poszczególnych bramek zastosowanych w układach. Zamieszczono również informacje o powierzchni układów, ale z pominięciem powierzchni połączeń między bramkami. Powierzchnie bramek odczytano z layout'u (rys. 1.18) i unormowano do powierzchni inwertera (tabela 3.2).

Tabela 3.2. Powierzchnia bramek unormowana do powierzchni inwertera

	NOT	2-we NAND	3-we NAND	4-we NAND	2-we NOR	3-we NOR	4-we NOR
powierzchnia	1	1,423	1,945	2,404	1,755	3,099	4,443

Wartości pojemności ekwiwalentnych zebrane w tabelach 3.3 i 3.4 reprezentują własności energetyczne układów testowych oraz pokazują potencjalne możliwości poprawy tych własności przez właściwy dobór sygnałów do wejść bramek, czyli ustalenie najlepszej kolejności wejść.

Rzeczywistą dynamiczną moc pobieraną przez układy dla danej częstotliwości f zmian wektorów wejściowych można obliczyć wykorzystując znaną formułę:

$$P_{dyn_tot} = f C_{eqv_tot} V_{dd}^2 \quad (3.5)$$

Tabela 3.3.a. Wyniki optymalizacji układów testowych MCNC dla rozkładu równomiernego (synteza: program SIS, *script.rugged*)

nazwa układu	we. licz.	wy. licz.	poz. licz.	wz. licz.	liczba bramek							pow.	całkowita $C_{equiv.}$ [fF]		różnica	
					not	na2	na3	na4	no2	no3	no4		MIN	MAX	[fF]	[%]
dk27	4	5	12	23	8	8	4	0	3	0	0	34,4	60,80	61,19	0,39	0,63
lion	4	3	7	15	3	6	1	0	5	0	0	25,5	49,06	49,58	0,52	1,05
shift	4	4	11	24	5	17	1	0	1	0	0	33,5	66,82	67,14	0,31	0,47
tra04	4	3	6	14	5	6	0	0	2	1	0	20,1	47,21	48,01	0,80	1,66
bbtas	5	5	9	27	8	9	4	0	3	3	0	41,1	89,19	91,69	2,49	2,72
dk15	5	7	19	51	10	18	6	0	16	1	0	87,5	163,83	165,79	1,96	1,18
dk17	5	6	18	44	8	18	3	0	10	5	0	72,3	152,23	158,81	6,58	4,14
dk512	5	7	17	56	14	24	9	1	5	2	1	87,5	163,23	167,12	3,89	2,33
mod12	5	5	12	38	9	7	2	0	14	5	0	65,3	147,54	154,30	6,76	4,38
bw	5	28	27	126	34	42	15	3	23	7	2	201,1	388,89	401,92	13,03	3,24
rd53	5	3	12	33	8	15	0	1	8	0	1	50,2	113,50	115,37	1,87	1,62
squar5	5	8	22	53	13	25	4	0	8	3	0	80,9	166,43	169,02	2,59	1,53
xor5	5	1	12	16	4	8	0	0	4	0	0	25,0	48,60	48,60	0,00	0,00
dk14	6	8	25	72	14	12	1	2	27	12	4	140,2	335,70	351,60	15,90	4,52
ex7	6	6	20	55	12	22	8	0	11	2	0	88,8	132,20	135,24	3,04	2,25
lion9	6	5	14	45	12	4	1	0	21	7	0	82,4	182,63	189,82	7,19	3,79
tav	6	6	5	22	9	5	2	0	4	2	0	33,1	73,29	73,83	0,54	0,73
tra11	6	5	19	57	12	18	6	1	11	7	2	101,6	181,56	193,89	12,34	6,36
5xp1	7	10	33	103	24	51	7	0	17	4	0	158,1	322,59	327,50	4,91	1,50
con1	7	2	6	18	6	7	1	0	3	1	0	26,9	74,14	77,39	3,25	4,20
inc	7	9	23	77	20	36	11	0	8	2	0	115,4	203,54	207,63	4,09	1,97
rd73	7	3	23	60	17	26	1	2	10	3	1	92,0	189,98	198,64	8,66	4,36
Z5xp1	7	10	33	102	25	45	9	4	15	3	1	156,2	307,35	309,56	2,21	0,71
ex5	8	63	18	244	62	90	20	15	37	19	1	393,3	563,18	609,39	46,21	7,58
misex1	8	7	15	49	14	13	1	0	18	3	0	83,0	164,25	168,59	4,35	2,58
rd84	8	4	18	128	29	44	3	1	38	9	4	212,2	437,52	460,54	23,03	5,00
9sym	9	1	15	159	26	45	13	4	44	18	9	297,9	612,13	652,77	40,65	6,23
apex4	9	19	66	1548	197	367	74	45	557	229	78	3005,2	3297,09	3627,76	330,67	9,12
clip	9	5	17	99	21	55	12	0	8	3	0	147,1	294,08	296,93	2,85	0,96
Z9sym	9	1	13	159	31	40	13	6	45	18	6	289,1	577,80	606,87	29,07	4,79
sao2	10	4	23	106	21	37	8	1	27	8	4	181,6	340,89	357,08	16,18	4,53

Różnice między minimalną i maksymalną wartością pojemności ekwiwalentnej dla układów pokazują, że istnieje możliwość redukcji pobieranej mocy przez układ nawet o ponad 10% w przypadku dużych układów. Średni możliwy zysk wynosi 3,10% w przypadku zastosowania skryptu *rugged* i 2,51% w przypadku *algebraic*. Układ *apex4* po syntezie ze skryptem *algebraic* (tab. 3.3.b) jest zbudowany z ponad 1800 bramek, a różnica między maksymalną i minimalną wartością pojemności ekwiwalentnej wynosi ponad 11%. Lecz nie tylko rozmiary układu są tu istotne, ale również rodzaj zastosowanych bramek w układzie.

Tabela 3.3.b. Wyniki optymalizacji układów testowych MCNC dla rozkładu równomiernego (synteza: program SIS, *script.algebraic*)

nazwa układu	licz. we.	licz. wy.	licz. poz.	licz. wez.	liczba bramek						pow.	całkowita C _{equiv.} [fF]		różnica		
					not	na2	na3	na4	no2	no3		no4	MIN	MAX	[fF]	[%]
dk27	4	5	5	21	6	9	5	0	1	0	0	30,9	55,07	55,22	0,15	0,27
lion	4	3	5	17	5	6	1	0	4	1	0	26,9	59,23	60,22	0,99	1,65
shift	4	4	5	24	4	20	0	0	0	0	0	32,5	67,42	67,63	0,21	0,31
tra04	4	3	7	18	5	12	1	0	0	0	0	24,0	48,90	49,00	0,10	0,20
bbtas	5	5	8	31	8	13	2	0	7	1	0	49,0	87,26	88,68	1,43	1,61
dk15	5	7	7	54	11	22	8	3	8	1	1	86,7	164,54	168,29	3,75	2,23
dk17	5	6	6	47	8	24	10	0	4	1	0	73,0	128,32	130,46	2,14	1,64
dk512	5	7	7	59	9	26	7	1	12	3	1	96,8	204,01	208,03	4,02	1,93
mod12	5	5	7	36	9	17	6	0	2	1	0	51,4	98,78	99,76	0,98	0,99
bw	5	28	9	145	32	57	15	0	35	6	0	237,0	381,86	393,01	11,15	2,84
rd53	5	3	8	48	9	32	5	0	1	1	0	68,4	140,83	141,49	0,66	0,47
squar5	5	8	7	52	9	23	7	0	11	2	0	85,3	168,67	170,57	1,90	1,11
xor5	5	1	12	18	6	10	0	0	2	0	0	25,0	51,28	51,28	0,00	0,00
dk14	6	8	9	92	19	32	7	3	24	5	2	151,9	290,54	302,86	12,32	4,07
ex7	6	6	9	53	16	16	10	0	6	5	0	81,4	138,55	142,19	3,64	2,56
lion9	6	5	9	57	12	20	8	1	13	2	1	91,9	172,35	175,72	3,37	1,92
tav	6	6	5	22	9	5	2	0	4	2	0	33,1	73,29	73,83	0,54	0,73
tra11	6	5	11	68	16	12	3	5	23	8	1	120,5	187,45	196,88	9,43	4,79
5xp1	7	10	10	93	13	68	9	0	2	1	0	133,8	270,46	273,00	2,54	0,93
con1	7	2	5	21	5	15	1	0	0	0	0	28,3	59,30	59,47	0,17	0,29
inc	7	9	9	100	20	28	12	7	27	5	1	167,3	297,10	303,71	6,61	2,18
rd73	7	3	11	107	18	46	11	3	22	6	1	173,7	342,19	350,62	8,43	2,41
Z5xp1	7	10	11	115	20	62	6	1	23	2	1	173,3	364,78	370,40	5,62	1,52
ex5	8	63	9	274	48	74	23	18	72	30	9	500,6	901,95	960,89	58,94	6,13
misex1	8	7	7	40	14	12	8	0	6	0	0	61,1	104,52	106,90	2,38	2,23
rd84	8	4	15	136	24	58	8	1	32	9	4	226,3	465,71	493,75	28,03	5,68
9sym	9	1	13	170	22	45	15	6	56	23	3	312,5	671,94	701,23	29,29	4,18
apex4	9	19	17	1811	192	621	145	78	537	155	82	3332,4	2811,93	3160,05	348,12	11,02
clip	9	5	12	106	22	53	9	1	18	2	1	159,6	307,72	319,66	11,94	3,74
Z9sym	9	1	13	172	21	47	10	2	67	18	7	316,6	692,87	727,22	34,35	4,72
sao2	10	4	11	121	14	32	8	1	50	11	5	221,6	438,19	453,69	15,51	3,42

Analizując wyniki oceny parametrów energetycznych bramek przedstawione na rysunku 1.22, można zauważyć, że ogólnie, parametry bramek 3-wejściowych są gorsze niż 2-wejściowych porównując podobne sposoby sterowania. Dla bramek 3-wejściowych są większe wartości całkowitej pojemności ekwiwalentnej C_{Lall} . W przypadku bramek NAND dla niektórych grup sterowania różnice są minimalne (np. grupa 1), ponieważ rozmiary tranzystorów są porównywalne. Sytuacja pogarsza się w przypadku jednoczesnej zmiany sygnałów na kilku wejściach (grupa 2 lub 5). Druga obserwacja jest taka, że parametry bramek NOR są gorsze niż bramek NAND. Spowodowane jest to głównie dużo większymi rozmiarami tranzystorów, a więc

Tabela 3.4. Wyniki optymalizacji układów testowych MCNC dla rozkładu losowego

nazwa układu	licz. we.	licz. wy.	<i>script.rugged</i>						<i>script.algebraic</i>					
			NAND NOR		całkow. $C_{equiv.}$ [fF]		różnica		NAND NOR		całkow. $C_{equiv.}$ [fF]		różnica	
			[%]	[%]	MIN	MAX	[fF]	[%]	[%]	[%]	MIN	MAX	[fF]	[%]
dk27	4	5	52,2	13,0	59,69	60,28	0,58	0,97	66,7	4,8	54,33	54,60	0,27	0,49
lion	4	3	46,7	33,3	48,17	48,73	0,56	1,15	41,2	29,4	58,32	59,58	1,26	2,12
shift	4	4	75,0	4,2	66,37	66,81	0,44	0,66	83,3	0,0	66,86	67,17	0,31	0,46
tra04	4	3	42,9	21,4	46,65	47,42	0,77	1,63	72,2	0,0	48,29	48,48	0,19	0,39
bbtas	5	5	48,1	22,2	91,00	93,48	2,48	2,65	48,4	25,8	88,35	89,93	1,58	1,76
bw	5	28	47,6	25,4	393,06	406,52	13,46	3,31	49,7	28,3	386,08	398,05	11,97	3,01
dk15	5	7	47,1	33,3	165,47	167,59	2,12	1,26	61,1	18,5	166,25	170,06	3,81	2,24
dk17	5	6	47,7	34,1	155,20	162,10	6,90	4,25	72,3	10,6	130,26	132,61	2,35	1,77
dk512	5	7	60,7	14,3	163,15	167,74	4,58	2,73	57,6	27,1	205,14	209,88	4,74	2,26
mod12	5	5	23,7	50,0	147,86	155,13	7,27	4,69	63,9	8,3	99,80	100,85	1,05	1,04
rd53	5	3	48,5	27,3	114,39	116,62	2,23	1,91	77,1	4,2	142,30	143,18	0,88	0,61
squar5	5	8	54,7	20,8	167,95	171,03	3,08	1,80	57,7	25,0	169,95	172,18	2,23	1,30
xor5	5	1	50,0	25,0	48,76	48,88	0,12	0,24	55,6	11,1	51,67	51,74	0,07	0,13
dk14	6	8	20,8	59,7	334,10	350,46	16,37	4,67	45,7	33,7	288,94	301,37	12,43	4,13
lion9	6	5	54,5	23,6	182,11	189,63	7,52	3,97	49,1	20,8	171,77	175,42	3,65	2,08
tav	6	6	11,1	62,2	73,12	73,77	0,65	0,89	50,9	28,1	73,12	73,77	0,65	0,89
tra11	6	5	31,8	27,3	182,03	194,51	12,49	6,42	31,8	27,3	187,31	196,79	9,48	4,82
5xp1	7	10	43,9	35,1	321,69	326,63	4,94	1,51	29,4	47,1	269,70	272,28	2,58	0,95
con1	7	2	56,3	20,4	73,98	77,23	3,25	4,20	82,8	3,2	59,18	59,35	0,18	0,30
inc	7	9	44,4	22,2	202,75	206,84	4,09	1,98	76,2	0,0	296,25	302,97	6,72	2,22
rd73	7	3	61,0	13,0	189,53	198,25	8,72	4,40	47,0	33,0	341,48	350,05	8,57	2,45
Z5xp1	7	10	48,3	23,3	306,56	308,91	2,34	0,76	56,1	27,1	363,92	369,59	5,67	1,53
ex5	8	63	56,9	18,6	563,20	609,52	46,32	7,60	60,0	22,6	901,99	961,13	59,15	6,15
ex7	8	63	51,2	23,4	131,02	134,12	3,10	2,31	42,0	40,5	137,34	141,24	3,90	2,76
misex1	8	7	28,6	42,9	164,27	168,64	4,37	2,59	50,0	15,0	104,53	106,93	2,40	2,24
rd84	8	4	37,5	39,8	437,68	460,81	23,13	5,02	49,3	33,1	465,93	494,02	28,09	5,69
9sym	9	1	39,0	44,7	612,05	652,67	40,62	6,22	38,8	48,2	671,99	701,41	29,42	4,19
apex4	9	19	31,4	55,8	3297,90	3628,78	330,88	9,12	46,6	42,7	2812,31	3160,92	348,61	11,03
clip	9	5	67,7	11,1	294,15	297,02	2,87	0,96	59,4	19,8	307,80	319,77	11,97	3,74
Z9sym	9	1	37,1	43,4	577,80	606,95	29,15	4,80	34,3	53,5	692,79	727,25	34,46	4,74
sao2	10	4	43,4	36,8	340,79	356,99	16,20	4,54	33,9	54,5	438,01	453,54	15,52	3,42

większymi pojemnościami pasożytniczymi. Zatem zmiana kolejności wejść do bramek NOR może spowodować większe różnice między minimalną i maksymalną wartością pojemności. Te obserwacje są potwierdzone przez wyniki zawarte w tabelach 3.3 i 3.4. Układy, w których jest znaczna liczba bramek NOR, mają większe wartości całkowitej pojemności ekwiwalentnej (np. *apex4*, *9sym*, *Z9sym*). Również różnice między minimalną i maksymalną wartością pojemności są większe (np. *apex4*, *ex5*, *tra11*). Dla lepszego porównania w tabeli 3.4 wpisano procentowy udział bramek NAND i NOR w całkowitej liczbie bramek w układzie. Można zaobserwować, że różnice wartości pojemności są większe dla układów uzyskanych po syntezie ze skryptem *rugged*, w których jest większy udział bramek NOR. Oczywiście liczba bramek w układzie również ma wpływ na wielkość różnic, ale w mniejszym stopniu niż rodzaj bramek.

W przypadku układu *xor5*, który ma symetryczną strukturę z powodu realizowanej funkcji, zamiana kolejności wejść do bramek nie zmienia wartości pojemności ekwiwalentnej dla rozkładu równomiernego. Układ jest zbudowany tylko z bramek 2-wejściowych i inwerterów w ten sposób, że przeciwstawne sposoby sterowania mają to samo prawdopodobieństwo. Zmiana wejść miejscami nie daje więc redukcji pojemności ekwiwalentnej. Jednakże dla rozkładu nierównomiernego ta symetria zostaje zaburzona i pojawia się możliwość optymalizacji układu.

Wyniki zebrane w powyższych tabelach pokazują, że możliwy do uzyskania poziom redukcji pobieranej mocy, przez właściwy dobór kolejności wejść do bramek, silnie zależy od struktury układu tj. liczby i rodzaju bramek. Jednakże bazując na nowym, zaproponowanym modelu poboru mocy dynamicznej możliwa jest optymalizacja układów logicznych, prowadząca do redukcji konsumowanej energii.

Liczba wejść ma duży wpływ na własności energetyczne bramki. Wraz z jej wzrostem pogarszają się parametry bramki. Z drugiej strony większa liczba wejść do bramki daje większe możliwości w poszukiwaniu optymalnej kolejności i w efekcie redukcji poboru mocy. Staje się to bardziej widoczne dla nierównomiernych rozkładów prawdopodobieństwa zmian wektorów wejściowych (układ *xor5*). Chociaż nasuwa się pytanie, czy nie byłoby lepiej, właśnie dla nierównomiernych rozkładów, pójść dalej i dokonać głębszej restrukturyzacji układu niż tylko zamiana kolejności wejść do bramek, lub też, już na etapie syntezy logicznej, uwzględniać parametry energetyczne bramek.

ROZDZIAŁ 4.

Redukcja strat energii na etapie syntezy logicznej

W poprzednim rozdziale przedstawiono możliwości wykorzystania nowego modelu strat mocy w statycznych układach CMOS w celu poprawy parametrów energetycznych układu przez zamianę kolejności wejść do bramek. Ta optymalizacja jest wykonywana w ostatnim etapie projektowania, gdy znana jest już dokładna struktura układu, czyli liczba oraz rodzaj bramek i połączenia między nimi. Jedynie tylko wejścia bramek zamieniamy między sobą. Jednakże, gdyby na wcześniejszym etapie projektowania uwzględnić parametry energetyczne bramek i rozkład prawdopodobieństwa zmian wektorów wejściowych, to możliwe jest zaprojektowanie układu o jeszcze mniejszym poborze mocy.

Dowolna dwuwartościowa funkcja logiczna (funkcja Boole'a) może być zrealizowana za pomocą dwóch poziomów bramek logicznych. Pierwszy poziom składa się z bramek AND realizujących iloczyny termalne, a drugi z bramki OR – suma. Zwykle wymagane jest jeszcze użycie inwerterów, aby uzyskać negacje zmiennych. Jest to tzw. dwupoziomowa realizacja funkcji logicznej. Uzyskuje się ją z kanonicznej postaci dysjunkcyjnej (suma iloczynów) lub z postaci dualnej – kanonicznej postaci koniunkcyjnej (iloczyn sum). Chociaż w drugim przypadku pierwszy poziom logiczny składa się z bramek OR a drugi z AND.

Dwupoziomowe funkcje logiczne zostały w przeszłości gruntownie przestudiowane i metody ich optymalnych implementacji układowych można znaleźć już we wczesnych pracach [McC56], [Qui52], [Qui55]. Pojawienie się układów programowalnych spowodowało dalszy rozwój metod projektowania układów dwupoziomowych [Fle75]. Opracowano wiele zaawansowanych technik optymalizacji dających dokładny lub przybliżony wynik. Choćby wspomnieć system ESPRESSO [Bra84], w którym zastosowano heurystyczne metody w celu redukcji powierzchni układu. Natomiast w ESPRESSO EXACT [Rud89] zawarto metody pozwalające uzyskać dokładne rozwiązanie zadania minimalizacji powierzchni. Optymalizując powierzchnię układu jako kryterium stosuje się liczbę iloczynów termalnych i liczbę literałów². Taki wybór funkcji celu jest dość dobry i niezależny od technologii wykonania układu, tzn., że te same metody mogą być zastosowane do układów statycznych i dynamicznych. Jednakże w przypadku redukcji poboru mocy nie można stosować tych samych kryteriów dla układów

² **literal** – to zmienna lub zmienna zanegowana występująca w formule boolowskiej

statycznych i dynamicznych [Dev92]. Zatem różne strategie projektowania muszą być zastosowane dla różnych technologii.

4.1. Metody redukcji poboru mocy w dwupoziomowych układach logicznych

W literaturze można znaleźć bardzo dużo prac na temat redukcji poboru mocy w układach logicznych. Zwykle wykorzystywany jest tradycyjny model konsumpcji mocy w układach CMOS, który był opisany w pierwszym rozdziale niniejszej pracy. Zakładając stałą pojemność węzłową w wielu metodach redukcję poboru mocy sprowadza się do zmniejszania aktywności układu. Wykorzystywano przy tym, prawdopodobieństwo sygnału lub gęstość przełączeń albo obydwa parametry.

W pracy [She92] rozważano wpływ prawdopodobieństwa przełączeń na średni pobór mocy oraz na testowalność układu. Zaproponowano metodę restrukturyzacji układu oraz inne metody syntezy redukujące aktywność węzłów. Jeśli, zgodnie z równaniem (1.6), prawdopodobieństwo sygnału będzie bliskie jeden lub zero to prawdopodobieństwo zmiany stanu węzła będzie najmniejsze. Bazując na tej obserwacji starano się wykorzystując lokalne nieokreśloności funkcji włączyć je do zbioru ON-set lub OFF-set tak, aby prawdopodobieństwo sygnału było możliwie najbliższe jeden lub zero. Zatem, najpierw optymalizowano funkcję otrzymując układ wielopoziomowy a następnie optymalizowano każdy z węzłów stosując metody dwupoziomowe, wykorzystując wspomniane nieokreśloności a także rozłączne pokrycia.

Metodę opartą na zmodyfikowanych procedurach systemu ESPRESSO przedstawiono w pracy [Bah95]. W układzie, który był najpierw zoptymalizowany pod względem powierzchni i opóźnień czasowych oraz zmapowany z określoną biblioteką, wydzielano następnie grupy bramek przez składanie (ang. *collapse*) dwóch lub więcej poziomów w pojedynczy węzeł. Podczas optymalizacji każdej z grup stosowano dwupoziomową optymalizację bazującą na algorytmach ESPRESSO zmodyfikowanych przez odpowiednie procedury heurystyczne prowadzące do redukcji poboru mocy.

Autorzy pracy [Vru94] pokazali, jak balansować pomiędzy potencjalnie kolidującymi celami, jakimi są minimalizacja powierzchni i poboru mocy. Przedstawiono metodę zmniejszającą rozmiary zachodzących na siebie fragmentów kostek w pokryciu realizowanej funkcji, dążąc w efekcie do pokryć rozłącznych. Jednakże przyjęto upraszczające założenie, że wartości prawdopodobieństwa sygnałów wejściowych do układu są jednakowe i równe 0,5.

W pracach [Ima95] oraz [Ima98] poprawiono główną słabość powyższego rozwiązania i uwzględniono dowolne rozkłady prawdopodobieństwa sygnałów wejściowych. Zdefiniowano

ogólny problem minimalizacji poboru mocy przez projektowany układ następująco: mając daną funkcję boolowską f wraz ze zbiorem wejść V oraz wartości prawdopodobieństwa sygnału na każdym z wejść trzeba znaleźć dwupoziomową implementację tej funkcji, taką, aby pobór mocy, określony przyjętym modelem, był minimalny. Autorzy tych prac stosowali tradycyjny model poboru mocy (rozdział 1.1). Następnie wprowadzono, przez analogię do metod minimalizacji powierzchni, pojęcie *mocowych implikantów prostych* (ang. *Power Prime Implicant*). Dowolny implikant funkcji jest *PPI*, jeśli wszystkie implikanty, które zawierają dany implikant mają większy pobór mocy określony przyjętym modelem. Prosty algorytm znajdowania wszystkich *PPI* dla funkcji N -wejściowej polega na wygenerowaniu wszystkich n -kostek funkcji zaczynając od $n=1$ do N i porównaniu poboru mocy kostek ze wszystkimi zawierającymi daną kostkę. Przy czym przez n -kostkę rozumiemy kostkę zbudowaną z n literałów. Taki sposób szukania implikantów staje się szybko nieakceptowalny wraz ze wzrostem liczby wejść do układu, gdyż wymaga oszacowania konsumpcji mocy dla bardzo dużej liczby kostek. Dlatego, bazując na przyjętym modelu, wyprowadzono warunki pozwalające szybko określić potencjalnych kandydatów na *PPI*. Wzięto pod uwagę relacje pomiędzy wartościami prawdopodobieństwa sygnałów wejściowych i sygnału wyjściowego dla danego implikanta. Należy nadmienić, że implikanty funkcji, które nie są implikantami prostymi z punktu widzenia minimalizacji powierzchni mogą być użyteczne w procesie minimalizacji poboru mocy stając się *PPI*.

Autorzy uzyskali redukcję mocy do 10% w porównaniu z realizacją układów o minimalnej powierzchni dla przypadków funkcji o większej liczbie wejść i dużym współczynniku *on-set*. Współczynnik *on-set* został zdefiniowany jako stosunek mocy zbioru „jedynek” do liczby wszystkich punktów określonych w przestrzeni funkcji. Średni zysk wyniósł 2,3%.

W przedstawionej metodzie optymalizacji układów dwupoziomowych przyjęto pewne założenia upraszczające odnośnie pojemności węzłowych. Założono, że wartości pojemności wejściowych i wyjściowych bramek AND i OR są równe, przez co metoda sprowadza się do redukcji aktywności układu. Poza tym wzięto pod uwagę tylko statyczne prawdopodobieństwo sygnałów, pomijając czasowe i przestrzenne korelacje między sygnałami wejściowymi.

Wykorzystując prawdopodobieństwo przełączeń sygnałów wejściowych do układu, można uwzględnić ich czasowe korelacje. Autorzy prac [The98] i [The99] bazując na statystycznych własnościach sygnałów, tj. prawdopodobieństwie sygnału (nazwanym statycznym) i prawdopodobieństwie przejść, zaproponowali metodę redukcji aktywności, a za tym również poboru mocy, w układach dwupoziomowych. Wykorzystano tradycyjny model konsumpcji mocy. Redukcja aktywności węzłów sieci logicznej była osiągnięta przez zwiększenie liczby wejść

pewnych bramek pierwszego poziomu (tj. AND) i dołączenie dodatkowych zmiennych wejściowych. Specjalnie wybrane sygnały wymuszają logiczny stan zero na wyjściach bramek AND dla pewnej liczby wektorów wejściowych do układu. Te zmienne zwane są zmiennymi blokującymi. Zaproponowano metodę grupowania sygnałów wejściowych w klasy o podobnych własnościach statystycznych i sposób wyboru właściwych zmiennych blokujących. W oparciu o koncepcję zmiennych blokujących opracowano narzędzie do syntezy dwupoziomowych układów z obniżonym poborem mocy, wykorzystując główne procedury programu ESPRESSO.

Przedstawione przez autorów wyniki eksperymentów pokazują, że możliwa jest średnia redukcja poboru mocy do 7% w porównaniu z układami zoptymalizowanymi pod względem powierzchni. W niektórych przypadkach uzyskano zysk do 20%.

W pracach [Tse97] i [Tse99] przedstawiono metodę redukcji mocy rozpraszanej w układach dwupoziomowych bazującą na gęstości przełączeń i prawdopodobieństwie sygnału. Obliczając gęstość przełączeń w węzłach układu uwzględniono przypadki jednoczesnej zmiany stanów na kilku wejściach bramki. Dla zaproponowanej metody zmodyfikowano algorytmy programu ESPRESSO. Wykorzystano nieokreśloności funkcji w celu redukcji aktywności bramek realizujących poszczególne implikanty. Procedurę ekspansji tak zmodyfikowano, że powiększanie kostek jest prowadzone, jeśli powoduje zmniejszenie gęstości przełączeń. W tym celu określono wagi kostek. W pracach rozważano różne technologie wykonania układów logicznych, mianowicie statyczne i dynamiczne PLA oraz typowy, statyczny układ dwupoziomowy AND-OR. W związku z tym wykorzystano stosowne parametry jako wagi kostek. Były to, gęstość przełączeń sygnału wyjściowego kostki dla statycznych układów PLA, statyczne prawdopodobieństwo „0” dla dynamicznych oraz gęstość przełączeń kostki i jej zmiennych wejściowych w przypadku układów AND-OR. Zatem stosowano tradycyjne modele konsumpcji mocy. Wybierając pokrycie kolumnowe macierzy blokującej obliczano sumę gęstości przełączeń zmiennych. W ten sposób obliczano wagi powiększanych kostek. Następnie wybierano pokrycia z możliwie najmniejszymi wagami – gęstościami przełączeń, rozwiązując zarazem wagowy problem pokrycia.

Autorzy uzyskali średnią redukcję pobieranej mocy o ponad 8% dla układów typu AND-OR w porównaniu z realizacją wykorzystującą procedurę *Symplify* z programu ESPRESSO. Podobne wyniki uzyskano dla statycznych układów PLA – średni zysk ponad 11%. W przypadku dynamicznych układów PLA uzyskano poprawę własności energetycznych układów zaledwie o 1,4%.

Opisane powyżej metody optymalizacji dwupoziomowych układów logicznych pod względem poboru mocy, bazując na tradycyjnym modelu, nie uwzględniają zależności

konsumpcji mocy od sytuacji na wszystkich wejściach bramki traktowanej jako zdarzenie sterowania bramki. Ponadto w wielu pracach nie przykładano należytej wagi do wartości pojemności węzłowych, koncentrując się tylko na redukcji aktywności układu. Z drugiej strony przyjmując w rozważaniach tradycyjny model poboru mocy nie było możliwości uwzględnienia wszystkich subtelnych i złożonych zależności wpływu sposobu sterowania bramki na jej pobór mocy. Nowy, zaproponowany w niniejszej rozprawie, model uzupełnia tę lukę i pozwala na zaprojektowanie lepszych układów pod względem energetycznym.

4.2. Synteza logiczna dwupoziomowych układów o obniżonym poborze mocy z wykorzystaniem nowego modelu

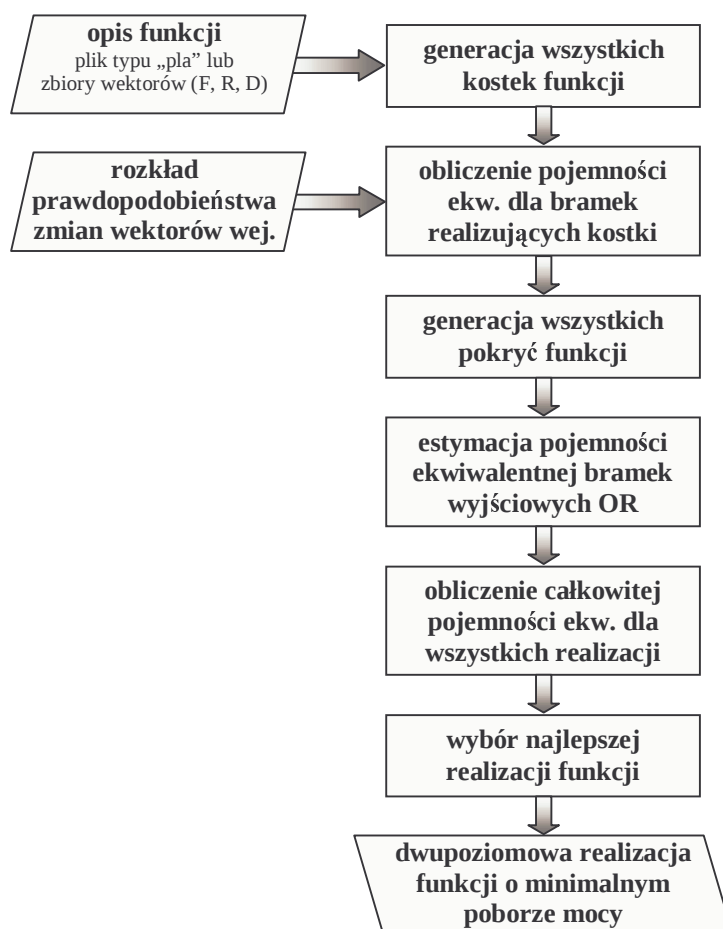
Zdefiniowany przez Iman'a i Pedram'a w [Ima95], [Ima98] ogólny problem dokładnej minimalizacji poboru mocy dla dwupoziomowej realizacji funkcji logicznej znajduje również zastosowanie przy wykorzystaniu nowego, zaproponowanego w niniejszej pracy, modelu poboru mocy. Szczegółowo problem ten może być podany następująco: mając daną funkcję boolowską f , wraz ze zbiorem wejść v oraz prawdopodobieństwo zmian wektorów wejściowych p_v i bibliotekę bramek z wyznaczonymi wartościami pojemności ekwiwalentnych należy znaleźć dwupoziomową realizację funkcji o najmniejszym poborze mocy określonym za pomocą nowego modelu. Z przyjętych założeń modelu (opis w rozdziale pierwszym, punkt 1.2.4) wynika, że zadanie minimalizacji sprowadza się do znalezienia układu o najmniejszej całkowitej pojemności ekwiwalentnej $C_{T_equ_2l}$, będącej sumą ekwiwalentnych pojemności bramek AND realizujących implikanty funkcji oraz bramki wyjściowej OR. Wykorzystując równanie (1.17) określające przyjęty model całkowitą pojemność ekwiwalentną układu dwupoziomowego można zapisać jako:

$$C_{T_equ_2l} = \sum_i \sum_{k=1}^{2^{2n}} c_{equ_AND}^i(dw_k^i) \cdot p(dw(k)) + \sum_{l=1}^{2^{2m}} c_{equ_OR}(dw_l) \cdot p(dw_{OR}(l)) \quad (4.1)$$

gdzie: $c_{equ_AND}^i(dw_k^i)$ – ekwiwalentna pojemność i -tej bramki AND dla k -tego sposobu jej sterowania, c_{equ_OR} – ekw. poj. wyjściowej bramki OR, dw_k^i – k -ty sposób sterowania i -tej bramki, $p(dw(k))$ – prawdopodobieństwo wystąpienia k -tego sposobu sterowania, n – liczba wejść do i -tej bramki AND, m – liczba wejść do bramki OR. Przy czym przez pojemność ekwiwalentną bramki w tym równaniu, rozumiemy całkowitą pojemność ekwiwalentną bramki C_{Lall} , czyli sumę pojemności wewnętrznej i wejściowych, tak jak to było w równaniu (3.4).

Stosownie do powyższego opisu dokładny algorytm syntezy układu dwupoziomowego o minimalnej mocy strat z uwzględnieniem nowego modelu energetycznego można szczegółowo przedstawić jak na rysunku 4.1. W pierwszym etapie dla funkcji opisanej za pomocą pliku „pla”

lub zbiorów wektorów (F, R, D) następuje wygenerowanie wszystkich kostek. Uwzględniając zadany rozkład prawdopodobieństwa zmian wektorów wejściowych układu, jest obliczana pojemność ekwiwalentna bramek realizujących kostki. W punkcie 4.2.2 opisano metody obliczania aktywności kostek nie wymagające żmudnych symulacji logicznych układu. Następnie należy wygenerować wszystkie możliwe pokrycia funkcji bazując na wcześniej otrzymanym zbiorze kostek. Po obliczeniu pojemności ekwiwalentnej dla bramek OR we wszystkich otrzymanych pokryciach i zsumowaniu wartości pojemności dla wszystkich bramek danej realizacji następuje wybór najlepszej realizacji funkcji, czyli tej o najmniejszej wartości całkowitej pojemności ekwiwalentnej.



Rys. 4.1. Algorytm syntezy układu dwupoziomowego o minimalnej mocy strat z uwzględnieniem nowego modelu energetycznego bramek

Tak postawiony problem jest dość trudny w realizacji ze względu na dużą liczbę możliwych kostek oraz pokryć funkcji. Dla dowolnej funkcji o N zmiennych wejściowych można określić górną granicę liczby kostek. Jest to liczba N -elementowych wariacji z powtórzeniami ze zbioru 3-elementowego, która wynosi: 3^N . Jednakże jest to słuszne dla funkcji zawsze prawdziwej, czyli przyjmującej „1” dla wszystkich wektorów wejściowych. Jeśli moc zbioru „jedynek” (on-set) jest mniejsza niż 2^N , to maksymalna liczba kostek znacząco maleje. Na przykład dla funkcji

4-wejściowej maksymalna liczba kostek wynosi: 3^4 czyli 81. Ale jeśli pojawi się choćby jedno zero, to liczba kostek wynosi 65. W przypadku dwóch zer sąsiadujących (tzn. mogą być pokryte jedną kostką) maksymalna liczba kostek wynosi 57, w przypadku 4 sąsiadujących zer 45 itd.

Równie skomplikowanym problemem jest generacja wszystkich możliwych pokryć funkcji dla uzyskanego zbioru kostek. Ze względu na potencjalnie dużą ich liczbę problem ten staje się bardzo czasochłonny i praktycznie nierozwiązywalny dla funkcji o większej liczbie zmiennych wejściowych.

Zarówno zadanie generacji wszystkich kostek jak i analiza wszystkich pokryć funkcji nie są problemami o złożoności wielomianowej. Należą one do klasy problemów NP-trudnych i wymagają zastosowania algorytmów o złożoności wykładniczej. Dlatego poszukuje się algorytmów heurystycznych o złożoności wielomianowej niskiego rzędu. Jednakże nie gwarantują one znalezienia rozwiązania optymalnego [DMi94].

Z tych powodów w dalszej części niniejszego rozdziału przedstawiono analizę parametrów energetycznych bramek (pojemności ekwiwalentnej) oraz zachowania się układów logicznych pod kątem poboru mocy w układach dwupoziomowych. Wnioski pozwolą na modyfikację ogólnego algorytmu syntezy (rys.4.1) tak, aby ograniczyć liczbę rozpatrywanych przypadków i w efekcie przyspieszyć pracę algorytmu. Następnie przedstawiono wyniki syntezy kilkunastu funkcji logicznych poprzedzone opisem efektywnych algorytmów obliczania nowej miary aktywności – prawdopodobieństwa sposobów sterowania – w układzie logicznym.

4.2.1. Praktyczny algorytm syntezy

Równomierny rozkład prawdopodobieństwa zmian wektorów wejściowych jest szczególnym przypadkiem, choć może obrazować pewne zależności w układach cyfrowych. Dlatego na wstępie zostaną przeanalizowane zależności pomiędzy liczbą wejść do bramki pierwszego poziomu a jej średnim poborem.

Zakładając równomierny rozkład zmian wektorów wejściowych oraz to, że każda zmiana wystąpi jeden raz, to dla N -wejściowego układu możliwych będzie 2^{2N} zmian. Jeśli zmiany wektorów będą występować w odstępach czasu T , to, aby wszystkie wystąpiły potrzeba:

$$t_{sym} = 2^{2N} T \quad (4.2)$$

Zatem średnią moc pobraną przez układ w tym czasie można obliczyć jako:

$$P_{sr} = \frac{1}{t_{sym}} \int_0^{t_{sym}} p(t) dt \quad (4.3)$$

gdzie: $p(t)$ – chwilowa moc strat w układzie. Uwzględniając to, że moc chwilowa to iloczyn chwilowych wartości napięcia i prądu, napięcie jest stałe i równe V_{dd} , oraz, że prąd to zmiana ładunku w czasie, powyższe równanie można następująco przekształcić:

$$P_{sr} = \frac{1}{t_{sym}} \int_0^{t_{sym}} u(t)i(t)dt = \frac{1}{t_{sym}} V_{dd} \int_0^{t_{sym}} i(t)dt = \frac{1}{t_{sym}} V_{dd} \int_0^{t_{sym}} \frac{dQ}{dt} dt = \frac{1}{t_{sym}} V_{dd} \int_0^{t_{sym}} dQ = \frac{1}{t_{sym}} V_{dd} Q(t_{sym}) \quad (4.4)$$

gdzie: $Q(t_{sym})$ to całkowity ładunek elektryczny jaki przepłynął ze źródła zasilania do układu w czasie t_{sym} . Jeśli za ładunek podstawimy pojemność ekwiwalentną układu C_{equ} oraz za czas t_{sym} liczbę wszystkich zmian wektorów według (4.2), to w efekcie otrzymamy równanie:

$$P_{sr} = \frac{1}{2^{2N} T} V_{dd}^2 C_{equ} \quad (4.5)$$

lub zastępując okres T przez częstotliwość f :

$$P_{sr} = \frac{1}{2^{2N}} f V_{dd}^2 C_{equ} \quad (4.6)$$

Powyższe równanie opisuje średni całkowity pobór mocy dynamicznej przez N -wejściowy układ kombinacyjny dla rozkładu równomiernego. Nasuwa się pytanie, jaki w tej sytuacji będzie udział k -wejściowej bramki pierwszego poziomu w całkowitej konsumpcji mocy, przy czym $k \leq N$.

W przypadku dowolnego rozkładu prawdopodobieństwa zmian wektorów wejściowych do układu udział bramki w mocy całkowitej jest określony przez jej pojemność ekwiwalentną w tym układzie. Aby obliczyć jej wartość, można skorzystać z równania (1.17), wykorzystując wprowadzone w rozdziale trzecim pojęcie całkowitej pojemności ekwiwalentnej bramki C_{Lall} . Zatem dla k -wejściowej bramki pierwszego poziomu, można zapisać równanie określające jej całkowitą pojemność ekwiwalentną C_{br_t} jako:

$$C_{br_t} = \sum_{m=1}^{2^{2k}} c_{Lall}(m) p(dw_m) \quad (4.7)$$

Z tym, że mnożenie i sumowanie następuje dla wszystkich 2^{2k} sposobów sterowania bramki.

Jeśli liczba wejść bramki jest równa liczbie wejść do układu, to prawdopodobieństwo każdego sposobu sterowania bramki będzie równe prawdopodobieństwu zmian wektorów układu. Zatem w przypadku równomiernego rozkładu zmian wektorów przyjmuje wartość:

$$p_i = \frac{1}{2^{2N}} \quad (4.8)$$

Natomiast pojemność ekwiwalentna bramki będzie dana wzorem:

$$C_{br_t} = \sum_{m=1}^{2^{2k}} c_{Lall}(m) p_i = \frac{1}{2^{2N}} \sum_{m=1}^{2^{2k}} c_{Lall}(m) = \frac{1}{2^{2N}} C_{Lall} \quad (4.9)$$

gdzie: C_{Lall} – suma całkowitej pojemności ekwiwalentnej dla bramki. W przypadku, gdy liczba

wejść do bramki jest mniejszej niż liczba wejść do układu ($k < N$) i rozpatrywany jest równomierny rozkład zmian wektorów układu, to dana zmiana wektora wejściowego bramki powtórzy się tyle razy, ile jest możliwych zmian na tych wejściach głównych układu, które nie są podłączone do bramki. Liczba tych niepodłączonych wejść wynosi $N-k$, daje to 2^{N-k} wektorów występujących na niepodłączonych wejściach i $2^{2(N-k)}$ zmian. Zatem w przypadku bramki pierwszego poziomu prawdopodobieństwo każdego sposobu jej sterowania będzie $2^{2(N-k)}$ razy większe niż p_i – prawdopodobieństwo zmiany na wejściach układu. Stąd na podstawie równań (4.7) i (4.9) całkowitą pojemność ekwiwalentną k -wejściowej bramki $C_{br_t}^k$ można zapisać jako:

$$C_{br_t}^k = \frac{2^{2(N-k)}}{2^{2N}} \sum_{m=1}^{2^{2k}} c_{Lall}(m) = \frac{2^{2(N-k)}}{2^{2N}} C_{Lall} \quad (4.10)$$

i ostatecznie:

$$C_{br_t}^k = 2^{-2k} C_{Lall} \quad (4.11)$$

Powyższe równanie (4.11) pokazuje, że udział bramki w całkowitej mocy średniej układu nie zależy od liczby wejść do układu. Jedynie zależy od jej liczby wejść i jej parametrów energetycznych – pojemności ekwiwalentnej.

W tabeli 4.1 obliczono całkowitą pojemność ekwiwalentną dla bramek pierwszego poziomu, które były poddane ocenie własności energetycznych (rozdz. 1.3) dla przypadku równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych do układu. Szczegółowe wyniki oceny parametrów bramek – wartości pojemności ekwiwalentnych – umieszczono w załączniku B.

Tabela 4.1. Całkowita pojemność ekwiwalentna dla podstawowych bramek CMOS na pierwszym poziomie układu w przypadku równomiernego rozkładu zmian wektorów

bramka	liczba we. k	wsp. 2^{-2k}	ΣC_{Lall} [fF]	całk. poj.ekw. $C_{br_t}^k$ [fF]
not	1	$\frac{1}{4}$	10,022	2,505
nand2	2	$\frac{1}{16}$	45,712	2,857
nand3	3	$\frac{1}{64}$	207,125	3,236
nand4	4	$\frac{1}{256}$	883,647	3,452
nor2	2	$\frac{1}{16}$	74,344	4,646
nor3	3	$\frac{1}{64}$	537,434	9,241
nor4	4	$\frac{1}{256}$	3228,536	12,611

Na podstawie wyników zebranych w powyższej tabeli można wyciągnąć wnioski, że dla syntezy układów dwupoziomowych o obniżonym poborze mocy należy stosować bramki o jak najmniejszej liczbie wejść. Są to bramki realizujące kostki o jak największym wymiarze – implikanty proste funkcji. Zatem jest to stwierdzenie zgodne z celem optymalizacji powierzchni.

Jednakże jest ono słuszne tylko w przypadku równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych do układu. W przypadku rozkładu nierównomiernego każdy sposób sterowania może mieć inną wartość prawdopodobieństwa i równanie (4.11) nie będzie już słuszne. W mocy jednak pozostaje ogólne równanie (4.7). Zatem dla dowolnego rozkładu zmian wektorów należy obliczyć pojemność ekwiwalentną bramek pierwszego poziomu układu, realizujących pewien zbiór kostek funkcji. Następnie należy rozwiązać problem minimalnego ważonego pokrycia funkcji, traktując uzyskane wartości pojemności jako wagi kostek. Przy czym nie wystarczy ograniczyć się tylko od implikantów prostych. Porównując wyniki całkowitej pojemności ekwiwalentnej, zebrane w tabeli 4.1, widać pewne różnice pomiędzy wartościami dla kolejnych bramek, chociaż większe dla bramek NOR. Można by przypuszczać, że dla dowolnego rozkładu bramki o mniejszej liczbie wejść mogą być lepsze w aspekcie minimalizacji poboru mocy. Dlatego algorytm optymalizacji mógłby rozpoczynać prace od analizy przydatności implikantów prostych. Następnie mógłby poszukiwać wśród kostek o wymiarze o jeden mniejszym, gdyż nie można zaniedbywać tych mniejszych kostek funkcji.

Następujący przykład ilustruje możliwości wykorzystania bramek o większej liczbie wejść zamiast tych o mniejszej.

Przykład 4.1.

Funkcja logiczna, zupełna, jest zdefiniowana za pomocą zbioru $F=\{4,5,6,7,13,15\}$. Na rysunku 4.2, wykorzystując tabelę Karnaugh, przedstawiono realizację o minimalnej powierzchni oraz jedno z pokryć rozłącznych. W tym drugim przypadku dwuwejściowa bramka realizująca drugą kostkę (-1-1) została zastąpiona przez trójwejściową bramkę – kostka trzecia (11-1). Kostka pierwsza (01--) występuje w obu realizacjach funkcji.

	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	1	1	0
10	0	0	0	0

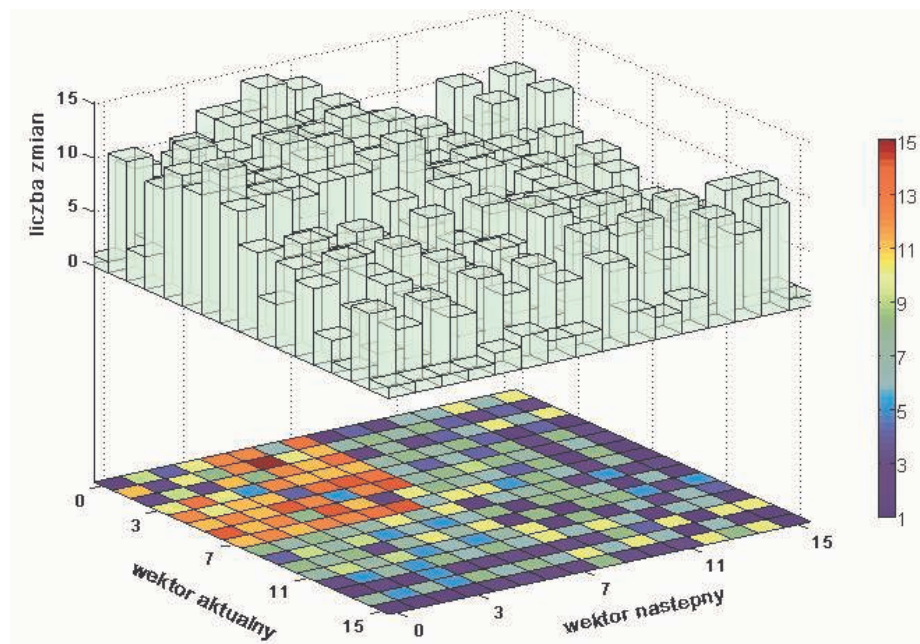
#1 (01--)

#2 (-1-1)

#3 (11-1)

Rys. 4.2. Tabela Karnaugh dla przykładowej funkcji z zaznaczonymi kostkami

Założono realizację funkcji z bramkami NAND. Następnie obliczono pojemności ekwiwalentne dla bramek realizujących podane kostki powyższej funkcji przy założeniu pewnego rozkładu prawdopodobieństwa zmian wektorów wejściowych, które zobrazowano na rysunku 4.3. Dla uproszczenia na osi Z podano całkowite liczby wystąpień poszczególnych



Rys. 4.3. Rozkład prawdopodobieństwa zmian wektorów wejściowych zastosowany w przykładzie 4.1 zmian wektorów. Jeśli wartości te zostaną podzielone przez sumę wszystkich możliwych zmian wektorów to zgodnie z definicją 1.1 oraz równaniem (1.16) otrzymamy prawdopodobieństwo sterowania układu. Wykorzystano również wcześniej wyznaczone parametry energetyczne bramek – pojemność ekwiwalentną C_{Lall} . Uzyskane wyniki obliczeń zebrano w tabeli 4.2. Oprócz pojemności ekwiwalentnej bramek, realizujących poszczególne kostki, podano również wartości pojemności ekwiwalentnej dla obu pokryć funkcji, ale nie uwzględniono tu jeszcze bramki wyjściowej. Pokrycie drugie, składające się z kostki pierwszej i trzeciej, jest o 3% lepsze pod względem redukcji mocy niż realizacja o minimalnej powierzchni.

Tabela 4.2. Pojemność ekwiwalentna kostek i pokryć dla funkcji z przykładu 4.1 (rozkład z rys. 4.3)

kostka	pojemność ekwiwalentna [fF]
#1 (01--)	2,8945
#2 (-1-1)	3,0024
#3 (11-1)	2,8278
pokrycie	pojemność ekwiwalentna [fF]
1) #1 i #2	5,8969
2) #1 i #3	5,7223

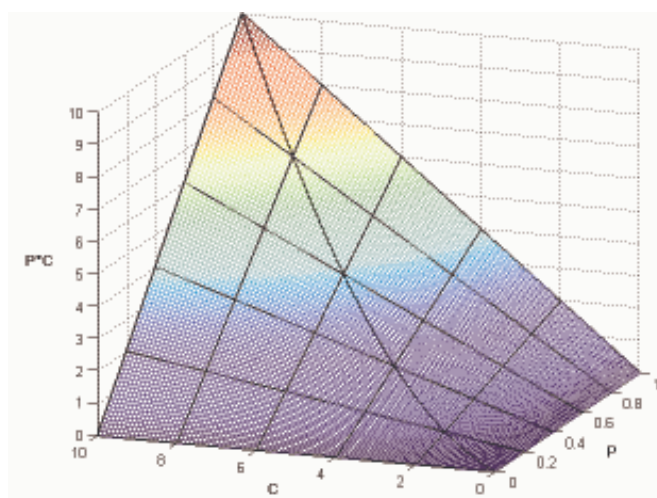
Wyniki zebrane w powyższej tabeli pokazują, że zastosowanie bramki o większej liczbie wejść powoduje redukcję mocy pobieranej przez układ. Jest to możliwe dla nierównomiernego rozkładu prawdopodobieństwa zmian wektorów, kiedy to równanie (4.11) oraz wartości pojemności ekwiwalentnej zebrane w tabeli 4.1, nie obowiązują. Potwierdza to wcześniej postawiony wniosek, że w ogólnym przypadku należy rozwiązać problem ważonego pokrycia funkcji, uwzględniając nie tylko implikanty proste.

Wyjaśnienie wyników otrzymanych w przykładzie 4.1 wymaga odwołania się do opisanego w rozdziale pierwszym nowego modelu mocy pobieranej przez statyczne układy CMOS oraz do sposobu obliczania całkowitej pojemności ekwiwalentnej bramki, który określa równanie (1.17). Całkowita pojemność ekwiwalentna to suma iloczynów prawdopodobieństwa poszczególnych sposobów sterowania bramki i odpowiednich częściowych pojemności ekwiwalentnych, które były wyznaczone w procesie oceny parametrów energetycznych bramki. Dla zrealizowania kostki drugiej z powyższego przykładu potrzeba bramki dwuwejściowej, a dla trzeciej trójwejściowej. Bramka dwuwejściowa, z racji liczby wejść, ma 16 sposobów sterowania, a trójwejściowa – 64. Wymiary tabel z prawdopodobieństwem sterowania, jak również z wartościami pojemności ekwiwalentnych, nie są więc równe dla tych bramek. Zatem nie ma możliwości bezpośredniego ich porównania. W związku z tym proces obliczania całkowitej pojemności ekwiwalentnej bramki zostanie odwrócony. Uprzednio dla każdej bramki obliczano tabelę z wartościami prawdopodobieństwa jej sposobów sterowania o wymiarach zależnych od liczby wejść. Każdy sposób sterowania układu wywołuje jakiś określony sposób sterowania bramki, przy czym różne sposoby sterowania układu mogą powodować ten sam sposób sterowania bramki. Zatem wtedy wartości prawdopodobieństwa zsumują się. W przypadku bramek pierwszego poziomu można bez symulacji logicznych określić, które sposoby sterowania układu spowodują ten sam sposób sterowania bramki. Jednakże można dla każdej bramki obliczyć nową tabelę pojemności ekwiwalentnej o wymiarach tabeli prawdopodobieństwa sterowania układu. Zatem wszystkie bramki pierwszego poziomu będą miały te same tabele prawdopodobieństwa sterowania, czyli tabelę sterowania układu. Natomiast tabele pojemności ekwiwalentnych będą różne nawet dla bramek tego samego typu i o tej samej liczbie wejść.

Proces tworzenia nowych tabel pojemności ekwiwalentnych dla bramek polega na przepisaniu wartości ze starych tabel w odpowiednie miejsca. Przepisanie nastąpi $2^{2(N-k)}$ -razy, podobnie jak we wcześniej przedstawionym sposobie obliczania prawdopodobieństwa sterowania bramki dla rozkładu równomiernego (równanie (4.10)). Przez N oznaczono liczbę wejść do układu, a przez k liczbę wejść do bramki. Pozycja poszczególnych wartości pojemności ekwiwalentnych w nowych tabelach jest określona przez to, do których wejść układu podłączone są wejścia bramki oraz to, czy jest to połączenie bezpośrednie, czy przez negację. Inaczej mówiąc N -pozycyjna kostka zapisana jako ciąg złożony z $\{0, 1, -\}$ będzie wykorzystana do obliczania nowych współczynników, adresów, określających miejsce w nowej tabeli. Ponadto kolejność wejść do samej bramki też ma wpływ na te współczynniki, a zatem rozkład wartości pojemności w tabeli. Jest to istotne, ponieważ całkowita wypadkowa pojemność ekwiwalentna bramki jest sumą iloczynów prawdopodobieństwa i odpowiednich wartości pojemności częściowych.

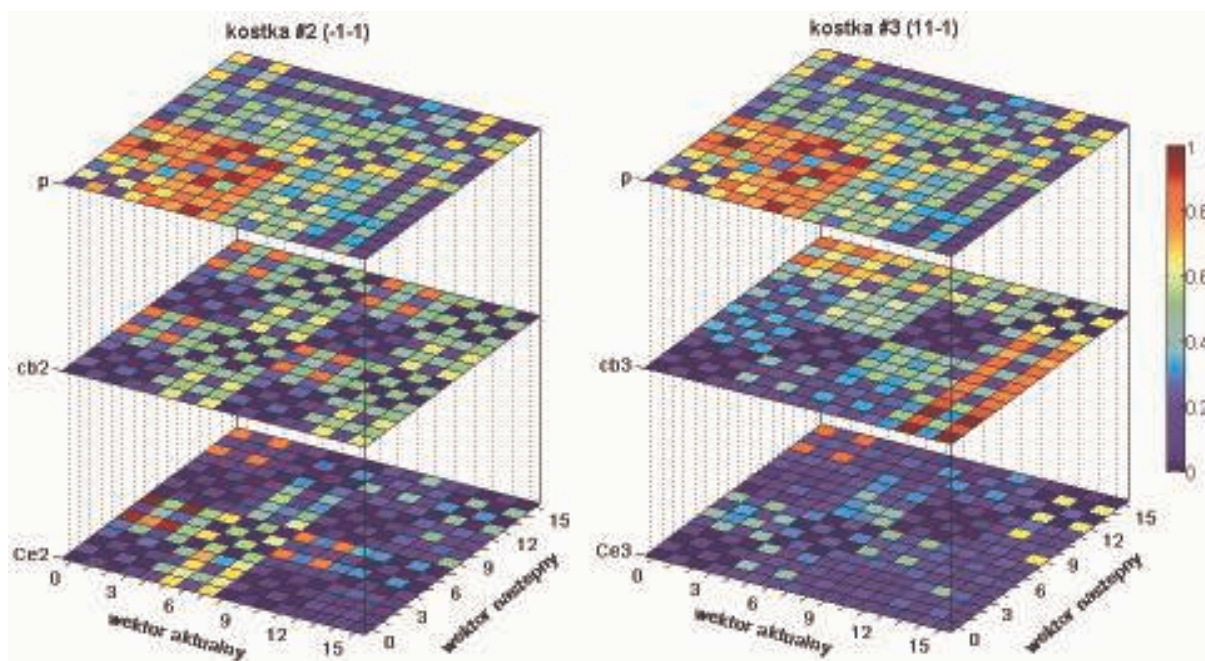
Aby otrzymać jak najmniejszą wartość pojemności całkowitej bramki, dobrze by było unikać mnożenia relatywnie dużych wartości, które mają potem znaczący wpływ na sumę. Znając wartości i rozkład prawdopodobieństwa sposobów sterowania układu można tak wybrać kostki funkcji i kolejność wejść do bramek, by suma iloczynów, czyli całkowita pojemność ekwiwalentna, była jak najmniejsza.

Na rysunku 4.4 przedstawiono iloczyn dwóch liczb P i C o wartościach odpowiednio $0÷1$ i $0÷10$ w celu zobrazowania wyniku iloczynu pojemności ekwiwalentnej i prawdopodobieństwa sterowania bramki. Jak widać, korzystnie jest, aby mnożone były wartości małe między sobą lub duża z małą. Jednakże należy mieć świadomość, że nie zawsze możliwy jest taki wybór kostek, aby spełnić te założenia.



Rys. 4.4. Iloczyn dwóch liczb o wartościach $0÷1$ i od $0÷10$ jako ilustracja iloczynu pojemności ekwiwalentnej i prawdopodobieństwa

Skoro opisano już sposób porównywania bramek realizujących różne kostki w układzie dwupoziomym i wyjaśniono przyczyny otrzymywanych wartości pojemności ekwiwalentnych, można wrócić do przykładu 4.1. W obu pokryciach funkcji występuje kostka pierwsza, zatem nie będzie ona przedmiotem naszego zainteresowania. Kostka trzecia (11-1) zawiera się w kostce drugiej (-1-1) a mimo to całkowita pojemność ekwiwalentna bramki realizującej ją jest mniejsza. Na rysunku 4.5 zobrazowano tabelę prawdopodobieństwa sterowania układu (p) oraz wartości pojemności ekwiwalentnych bramek realizujących te kostki ($cb2$ i $cb3$) jako kolorowe mapy o takich samych wymiarach zgodnie z uprzednim opisem. Dodatkowo przedstawiono również wartości iloczynów cząstkowych ($Ce2$ i $Ce3$). W celu zastosowania jednej skali i ułatwienia porównania unormowano wszystkie wartości w tabelach. Prawdopodobieństwo zmian wektorów wejściowych układu podzielono przez największą z wartości, tj. 0,0088 (należy pamiętać, że suma wszystkich wartości prawdopodobieństwa wynosi 1). Natomiast pojemność ekwiwalentną C_{Lall} bramek NAND2 i NAND3 podzielono przez największą z ich wartości, tj. 9,06fF

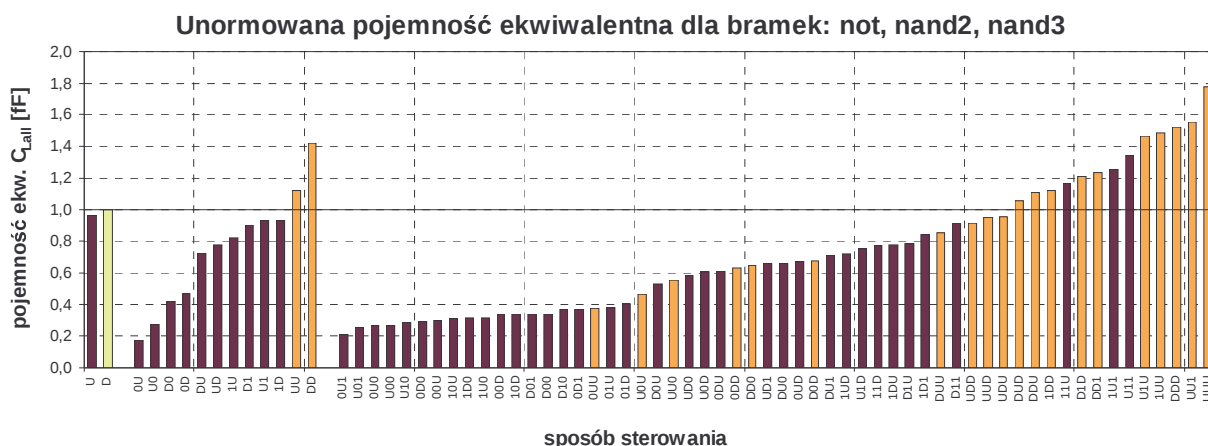


Rys. 4.5. Całkowita pojemność ekwiwalentna (C_e) bramek realizujących kostki z przykładu 4.1 (patrz załącznik B) oraz iloczyny pojemności i prawdopodobieństwa przez największą spośród wartości dla obu kostek, tj. 0,0551fF (największa wartość wystąpiła dla kostki #2).

W przypadku kostki trzeciej wartości znaczącej większości iloczynów ($Ce3$) mają małe wartości dzięki takiemu rozkładowi wartości pojemności bramki NAND3 w tabeli $cb3$, że praktycznie nie ma iloczynów dużych liczb. Dzięki temu całkowita, wypadkowa pojemność bramki realizującej kostkę trzecią jest mniejsza niż dla kostki drugiej, pomimo tego, że wartości pojemności ekwiwalentnej 3-wejściowej bramki NAND są większe niż 2-wejściowej (porównaj $cb2$, $cb3$, oraz tabela 4.1).

Przedstawione powyżej rozważania pozwalają na wybór odpowiednich kostek do pokrycia funkcji logicznej, z punktu widzenia redukcji poboru mocy. Jest to realizacja pierwszego poziomu układu. Pozostaje jeszcze zbadanie wpływu pokrycia, czyli wyboru kostek, na wielkość pobieranej mocy przez bramkę wyjściową.

Analizując wyniki oceny parametrów energetycznych bramek, uzyskane w rozdziale pierwszym (punkt 1.3.2), można zauważyć, że w przypadkach jednoczesnej i zgodnej zmiany sygnałów na kilku albo wszystkich wejściach bramki, następuje najczęściej znaczący pobór mocy. Na rysunku 4.6 przedstawiono wykresy pojemności ekwiwalentnej dla inwertera oraz dwu- i trójwejściowej bramki NAND. Wartości pojemności zostały unormowane do wartości pojemności ekwiwalentnej inwertera sterowanego opadającym zboczem oraz posortowane od najmniejszej do największej dla danej bramki. Ponadto pomarańczowym kolorem zaznaczono przypadki, gdy na kilku wejściach bramki występują jednoczesne i zgodne zbocza sygnałów.



Mając na uwadze wnioski z przeprowadzonych rozważań, można zaproponować praktyczny algorytm syntezy dwupoziomowych układów o obniżonym poborze mocy. Dla funkcji f określonej przez przynajmniej dwa ze zbiorów: F , R , D oraz dla rozkładu prawdopodobieństwa zmian wektorów wejściowych p_{we} i określonej biblioteki bramek z wyznaczonymi parametrami energetycznymi (pojemność ekwiwalentna) algorytm może być następujący:

Algorytm 4.1.

1. Dla danej funkcji f wygenerować zbiór wszystkich kostek K ,
2. Dla danego p_{we} obliczyć pojemność ekwiwalentną kostek K ,
3. Znaleźć pokrycia funkcji: minimalne ważone P_{min} i minimalne ważone, rozłączne PR_{min} ,
4. Obliczyć poj. ekwiwalentną dla bramki wyjściowej realizującej pokrycia P_{min} i PR_{min} ,
5. Z pośród PR_{min} i P_{min} wybrać lepsze pokrycie funkcji.

W algorytmie wzięto pod uwagę oba pokrycia funkcji: minimalne ważone oraz minimalne ważone, rozłączne, stosownie do wniosków poprzednio opisywanych.

Stosując metody symulacyjne, obliczanie pojemności ekwiwalentnej dla wszystkich kostek funkcji jest bardzo czasochłonne, dlatego w następnym punkcie niniejszego rozdziału zaproponowano efektywne metody obliczania aktywności bramek realizujących kostki funkcji. Zaproponowane algorytmy pozwolą na szybkie obliczenie pojemności w punkcie drugim powyżej zaproponowanego algorytmu syntezy.

4.2.2. Obliczanie prawdopodobieństwa sposobów sterowania bramek w układzie

Każda metoda estymacji, a także optymalizacji mocy pobieranej przez cyfrowe układy CMOS wymaga określenia aktywności każdej bramki w układzie. Informacja o aktywności układu jest bardzo ważna, gdyż to właśnie w czasie przełączania bramek następuje największa konsumpcja energii z zasilania, co jest ważną cechą technologii CMOS. W literaturze spotyka się metody symulacyjne oraz wykorzystujące probabilistyczne parametry sygnałów.

Zaproponowany w niniejszej rozprawie nowy model poboru mocy w kombinacyjnych układach CMOS wymaga obliczenia odpowiednich wartości dla nowej miary aktywności układu, prawdopodobieństwa sposobów sterowania bramek. W tym celu początkowo stosowano metodę symulacyjną. Wyniki estymacji i optymalizacji układów przedstawione w rozdziale drugim i trzecim były uzyskane właśnie z wykorzystaniem tej metody. Algorytm obliczania aktywności przedstawiono poniżej (Algorytm 4.2), a listing programu w załączniku A.

Algorytm 4.2.

1. *Dana jest netlista NL układu o n -wej. i prawdopodobieństwo zmian wektorów wej. p_{we} ,*
2. $V_a = 0$; *aktualny wektor wejściowy,*
3. *Dla V_a obliczyć stan logiczny wszystkich węzłów w układzie: $S_a = NL(V_a)$,*
4. $V_n = 0$; *następny wektor wejściowy,*
5. *Dla V_n obliczyć stan logiczny wszystkich węzłów w układzie: $S_n = NL(V_n)$,*
6. *Dla wszystkich węzłów w układzie obliczyć sposób sterowania:*
 - $v_a(j) = S_a(j)$; *obliczyć aktualny wektor wej. bramki sterującej węzeł j ,*
 - $v_n(j) = S_n(j)$; *obliczyć następny wektor wej. bramki sterującej węzeł j ,*
 - $p_j(v_a(j), v_n(j)) = p_j(v_a(j), v_n(j)) + p_{we}(V_a, V_n)$; *prawd. sterowania węzła j ,*
7. $V_n = V_n + 1$,
8. *Jeśli $V_n < 2^n$ idź do 5 w przeciwnym razie do 9,*
9. $V_a = V_a + 1$,
10. *Jeśli $V_a < 2^n$ idź do 3 w przeciwnym razie do 11,*
11. *Koniec obliczeń.*

Wynikiem obliczeń jest zestaw tabel z wartościami prawdopodobieństwa sposobów sterowania bramek dla wszystkich węzłów w układzie. Są to kwadratowe tabele o wymiarach zależnych od liczby wejść do bramek. Aktualny i następny wektor wejściowy jednoznacznie wskazuje miejsce w tabeli i jest sposobem sterowania bramki. Zatem aby wyznaczyć sposób sterowania bramki, należy na podstawie stanów wszystkich węzłów w układzie określić aktualny i następny wektor wejściowy danej bramki. Następnie prawdopodobieństwo określonej zmiany wektorów wejściowych układu $p_{we}(V_a, V_n)$ należy dodać do wartości wyznaczonej przez wektory bramki $(v_a(j), v_n(j))$ w jej tabeli sterowania, gdyż różne zmiany wektorów wejściowych mogą powodować taki sam sposób sterowania danej bramki.

Główną wadą tego algorytmu obliczania aktywności układu jest potrzeba 2^{2n} -krotnego obliczania stanu wszystkich węzłów w układzie. Obliczenia mogą zająć dużo czasu dla układów o dużej liczbie węzłów i wejść. Zaletą są niezbyt duże wymagania co do pamięci operacyjnej komputera.

Powyższy algorytm można zmodyfikować tak, aby najpierw obliczyć stany logiczne węzłów w układzie dla wszystkich 2^n wektorów wejściowych i zapamiętać je. Następnie, korzystając z tych wyników, można obliczać sposoby sterowania bramek w ten sam sposób jak w algorytmie 4.2. Wtedy stan węzłów jest obliczany 2^n -razy zamiast 2^{2n} , ale wymaga to zaangażowania większej ilości pamięci operacyjnej do zapamiętania stanów węzłów.

Zaproponowany w niniejszej pracy model poboru mocy wykorzystuje pojęcie sposobu sterowania bramki (punkt 1.2.3). Sposób sterowania może być zapisany jako k -elementowy wektor o składowych $\{0, 1, \uparrow, \downarrow\}$, przy czym k to liczba wejść do bramki. Podając parę binarnych wektorów, aktualny i następny, również można opisać sposób sterowania bramki, tak jak to robiono w punkcie 6 algorytmu 4.2. Jednakże dla obliczenia wartości tych wektorów trzeba było znać aktualne i następne stany węzłów w układzie.

Ponieważ model poboru mocy uwzględnia cztery wartości sygnałów wejściowych do bramki $\{0, 1, \uparrow, \downarrow\}$, to zamiast klasyczną tabelą prawdy bramkę należy opisać nową czterowartościową tabelą. Ponadto, taki sposób opisu zachowania się wyjścia bramki zawiera informację o stanie przed i po zmianie wektora wejściowego. Zatem informacja o sposobie sterowania następnej bramki jest podana bezpośrednio. Jeśli te cztery stany zostaną następująco zakodowane: 0 – 00, 1 – 11, $\uparrow$ – 01, $\downarrow$ – 10, to dla kilkuwejściowej bramki pierwsze bity z tych par określają aktualny wektor wejściowy a drugie następny. Te wektory wyznaczają jednoznacznie miejsce w tabeli aktywności bramki, podobnie jak w poprzednim algorytmie. Na rysunku 4.7 przedstawiono czterowartościową tabelę dla bramki AND wykorzystując symbole $\{0, 1, \uparrow, \downarrow\}$, oraz tabelę zakodowaną dwubitowo.

a)					b)				
a \ b	0	1	$\uparrow$	$\downarrow$	a \ b	00	11	01	10
0	0	0	0	0	00	00	00	00	00
1	0	1	$\uparrow$	$\downarrow$	11	00	11	01	10
$\uparrow$	0	$\uparrow$	$\uparrow$	0	01	00	01	01	00
$\downarrow$	0	$\downarrow$	0	$\downarrow$	10	00	10	00	10

Rys. 4.7. Czterowartościowa tabela prawdy dla bramki AND (a), zakodowana dwubitowo (b)

Analizując układ, począwszy od wejść głównych, należy dla każdej bramki obliczyć czterowartościową tabelę prawdy o rozmiarach równych rozmiarom tabeli prawdopodobieństwa zmian wektorów wejściowych układu. W ten sposób ze stanem wyjścia dowolnej bramki w układzie związane jest prawdopodobieństwo jego wystąpienia a zatem sterowania następnej bramki. Na podstawie netlisty i tabel czterowartościowych bramek można obliczyć tabele aktywności dla wszystkich bramek w układzie podobnie, jak w punkcie 6 algorytmu 4.2.

Algorytm obliczania prawdopodobieństwa sterowania bramek, wykorzystujący powyżej opisany pomysł tabel czterowartościowych, pozwala na uniknięcie czasochłonnych symulacji logicznych układu. Obliczenia rozpoczyna się począwszy od głównych wejść układu podążając do wyjść. Jeśli węzły w netliście są właściwie posortowane, to można jednocześnie obliczyć

tabelę czterowartościową i tabelę prawdopodobieństwa sterowania dla danego węzła. Na początku, oczywiście, potrzebne są tabele czterowartościowe dla zmiennych wejściowych. Są to ich wartości aktualne i następne, czyli odpowiednie bity przepisane z wektorów wejściowych. Na rysunku 4.8 przedstawiono przykładową tabelę dla zmiennej x_1 w układzie dwuwejściowym. Stan wyjścia bramki można określić na podstawie stosownych czterowartościowych tabel prawdy dla bramek (na rys. 4.7.a podano tabelę dla AND). Jednakże można wykorzystać zaproponowane dwubitowe kodowanie. Wtedy obliczanie stanu wyjścia bramki będzie polegało na wykonaniu operacji logicznej stosownej dla danej bramki (AND, OR, NAND, itd.) na wszystkich pierwszych bitach stanów wejściowych dla określenia pierwszego bitu stanu wyjściowego i na drugich dla drugiego bitu.

a)

$x_1x_2 \backslash x_1$	00	01	10	11
00	0	0	↑	↑
01	0	0	↑	↑
10	↓	↓	1	1
11	↓	↓	1	1

b)

$x_1x_2 \backslash x_1$	00	01	10	11
00	00	00	01	01
01	00	00	01	01
10	10	10	11	11
11	10	10	11	11

Rys. 4.8. Czterowartościowa tabela prawdy dla zmiennej x_1 (a), zakodowana dwubitowo (b)

Ponieważ rozpatrywane są stany wyjść bramek w odniesieniu do wszystkich możliwych zmian wektorów wejściowych, więc proponowana metoda obliczania aktywności układu jest niewrażliwa na występowanie korelacji przestrzennych między sygnałami w postaci rezbieżnych ścieżek w układzie. Algorytm 4.3 jest zapisem proponowanej metody. Kod źródłowy programu napisanego w języku Matlab zamieszczono w załączniku.

Algorytm 4.3.

1. Dana jest netlista NL układu o n -wej. i prawdopodobieństwo zmian wektorów wej. p_{we} ,
2. Obliczyć tabele czterowartościowe dla zmiennych wejściowych $T4(n)$,
3. Zaczynając od głównych wejść układu dla każdego węzła j :
 - obliczyć tabelę czterowartościową $T4(j)$,
 - obliczyć tabelę prawdopodobieństwa sposobów sterowania bramki $t_b(j)$,
4. Jeśli wszystkie węzły to koniec obliczeń.

Powyższa metoda pozwala na szybsze niż poprzednia (algorytm 4.2) obliczenie prawdopodobieństwa sposobów sterowania węzłów w układzie. Wadą jest znaczna zajętość pamięci operacyjnej komputera, chociaż mniejsza niż w przypadku zmodyfikowanego algorytmu 4.2.

Z każdym węzłem związana jest czterowartościowa tabela jego stanów o wymiarach $2^N \times 2^N$, gdzie N to liczba wejść do układu. Są to także wymiary tabeli prawdopodobieństwa zmian wektorów wejściowych układu. Można by oczywiście kasować w celu zaoszczędzenia pamięci, te tabele stanów węzłów, które już nie będą potrzebne. Jeśli obliczono stany wszystkich węzłów, dla których dany węzeł jest węzłem wejściowym (*ang. fan-in node*), to jego stan już nie jest potrzebny. Jednakże takie sprawdzanie netlisty mogłoby dodatkowo skomplikować program i spowolnić jego pracę.

Obserwacje i doświadczenia związane z poszukiwaniem różnych metod szybkiego obliczania nowej miary aktywności układu doprowadziły autora do opracowania jeszcze jednej metody. Pozwala ona jednak tylko na obliczanie aktywności bramek pierwszego poziomu w układzie. W obliczu algorytmu syntezy układów dwupoziomowych o obniżonym poborze mocy (alg. 4.1) znaczenie tej metody jest duże, gdyż zaproponowana metoda syntezy wymaga obliczania pojemności ekwiwalentnej dla wielu bramek realizujących kostki funkcji.

Jeśli bramka pierwszego poziomu ma tyle wejść, ile układ, to jej aktywność opisuje tabela prawdopodobieństwa zmian wektorów układu. Jednakże w większości przypadków bramki mają mniej wejść niż układ i ich tabele prawdopodobieństwa sposobów sterowania będą mniejsze niż tabela układu. Wtedy różne zmiany wektorów układu spowodują takie samo sterowanie bramki a wartości prawdopodobieństwa należy zsumować, podobnie jak to już opisywano w poprzednich algorytmach. Wykorzystując zapis pozycyjny kostek realizowanych przez bramki, można obliczyć nowe indeksy, nowe miejsca, w tabelach aktywności bramek dla poszczególnych wartości z tabeli układu. Na rysunku 4.9 przedstawiono sposób obliczania aktywności dla bramki realizującej przykładową kostkę k (10-) w układzie trójwejściowym. Pierwsza tabela (rys. 4.9.a) to tabela sterowania układu. Wektory binarne opisujące tę tabelę posłużą do obliczenia indeksów dla tabeli bramki. Ponieważ na trzeciej pozycji kostki k jest kreska, zatem istotne są tylko dwa pierwsze bity wektorów. Ponadto na drugiej pozycji jest zero, więc drugi bit trzeba zanegować. W ten sposób powstają nowe wektory, które są wektorami wejściowymi bramki i korespondują z poszczególnymi wektorami układu. Rysunek 4.9.b przedstawia tabelę układu opisaną nowymi wektorami, które zostały obliczone dla kostki k . Ostatni etap to przepisanie i zsumowanie wartości prawdopodobieństwa zmian układu do tabeli aktywności bramki stosownie do wartości nowych wektorów. Dziesiętna wartość wektorów to indeksy w tabeli dla bramki. Trzecia tabela (rys. 4.9.c) przedstawia zasadę sumowania wartości prawdopodobieństwa dla danego przykładu.

a)

$V_n \backslash V_a$	000	001	010	011	100	101	110	111
000	p ₀₀	p ₀₁	p ₀₂	p ₀₃	p ₀₄	p ₀₅	p ₀₆	p ₀₇
001	p ₁₀	p ₁₁	p ₁₂	p ₁₃	p ₁₄	p ₁₅	p ₁₆	p ₁₇
010	p ₂₀	p ₂₁	p ₂₂	p ₂₃	p ₂₄	p ₂₅	p ₂₆	p ₂₇
011	p ₃₀	p ₃₁	p ₃₂	p ₃₃	p ₃₄	p ₃₅	p ₃₆	p ₃₇
100	p ₄₀	p ₄₁	p ₄₂	p ₄₃	p ₄₄	p ₄₅	p ₄₆	p ₄₇
101	p ₅₀	p ₅₁	p ₅₂	p ₅₃	p ₅₄	p ₅₅	p ₅₆	p ₅₇
110	p ₆₀	p ₆₁	p ₆₂	p ₆₃	p ₆₄	p ₆₅	p ₆₆	p ₆₇
111	p ₇₀	p ₇₁	p ₇₂	p ₇₃	p ₇₄	p ₇₅	p ₇₆	p ₇₇

b)

$V_n \backslash V_a$	01	01	00	00	11	11	10	10
01	p ₀₀	p ₀₁	p ₀₂	p ₀₃	p ₀₄	p ₀₅	p ₀₆	p ₀₇
01	p ₁₀	p ₁₁	p ₁₂	p ₁₃	p ₁₄	p ₁₅	p ₁₆	p ₁₇
00	p ₂₀	p ₂₁	p ₂₂	p ₂₃	p ₂₄	p ₂₅	p ₂₆	p ₂₇
00	p ₃₀	p ₃₁	p ₃₂	p ₃₃	p ₃₄	p ₃₅	p ₃₆	p ₃₇
11	p ₄₀	p ₄₁	p ₄₂	p ₄₃	p ₄₄	p ₄₅	p ₄₆	p ₄₇
11	p ₅₀	p ₅₁	p ₅₂	p ₅₃	p ₅₄	p ₅₅	p ₅₆	p ₅₇
10	p ₆₀	p ₆₁	p ₆₂	p ₆₃	p ₆₄	p ₆₅	p ₆₆	p ₆₇
10	p ₇₀	p ₇₁	p ₇₂	p ₇₃	p ₇₄	p ₇₅	p ₇₆	p ₇₇

c)

$V_n \backslash V_a$	00	01	10	11
00	p ₂₂ +p ₂₃ + p ₃₂ +p ₃₃	p ₂₀ +p ₂₁ + p ₃₀ +p ₃₁	p ₂₆ +p ₂₇ + p ₃₆ +p ₃₇	p ₂₄ +p ₂₅ + p ₃₄ +p ₃₅
01	p ₀₂ +p ₀₃ + p ₁₂ +p ₁₃	p ₀₀ +p ₀₁ + p ₁₀ +p ₁₁	p ₀₆ +p ₀₇ + p ₁₆ +p ₁₇	p ₀₄ +p ₀₅ + p ₁₄ +p ₁₅
10	p ₆₂ +p ₆₃ + p ₇₂ +p ₇₃	p ₆₀ +p ₆₁ + p ₇₀ +p ₇₁	p ₆₆ +p ₆₇ + p ₇₆ +p ₇₇	p ₆₄ +p ₆₅ + p ₇₄ +p ₇₅
11	p ₄₂ +p ₄₃ + p ₅₂ +p ₅₃	p ₄₀ +p ₄₁ + p ₅₀ +p ₅₁	p ₄₆ +p ₄₇ + p ₅₆ +p ₅₇	p ₄₄ +p ₄₅ + p ₅₄ +p ₅₅

Rys. 4.9. Obliczanie tabeli aktywności dla bramki pierwszego poziomu,
(a) tabela prawdopodobieństwa zmian wektorów układu, (b) nowe współczynniki tabeli układu
obliczone dla kostki k (10-), (c) tabela aktywności bramki

Przedstawiona metoda pozwala na bardzo szybkie obliczanie aktywności bramek pierwszego poziomu na podstawie tabeli sterowania układu oraz kostek funkcji. Algorytm 4.4 jest ogólnym zapisem metody, a kod programu zamieszczono w załączniku A.

Algorytm 4.4.

1. Dany jest zbiór kostek K funkcji n -zmiennych wej. i prawdopodobieństwo zmian wektorów wej. p_{we} ,
2. Dla każdej kostki $k \in K$:
 - obliczyć wartości indeksów dla tabeli aktywności bramki na podstawie znaków w kostce: „-” – pominąć bit, „0” – zanegować, „1” – bez zmian,
 - obliczyć tabelę aktywności bramki wykorzystując obliczone indeksy,
3. Koniec obliczeń

Powyższy algorytm można w punkcie drugim uzupełnić o obliczanie czterowartościowych tabel stanów dla wyjść bramek. Wtedy można zastosować go jako wstępny etap w algorytmie 4.3. Natomiast w proponowanej metodzie syntezy dwupoziomowych układów o małym poborze mocy obydwa algorytmy znajdą zastosowanie.

W celu oceny efektywności zaproponowanych metod obliczania aktywności układów porównano czas obliczeń i ilość zajętej pamięci przez programy napisane według algorytmów 4.2 i 4.3. Obliczono prawdopodobieństwo sposobów sterowania bramek w układach testowych, które były przedmiotem eksperymentów w rozdziałach 2 i 3. Wyniki zebrano w tabeli 4.3.

Tabela 4.3. Wyniki porównania zaproponowanych metod obliczania aktywności układu

układ	Algorytm 4.2		Algorytm 4.3		Alg_4.2 / Alg_4.3	
	czas [s]	pamięć [kB]	czas [s]	pamięć [kB]	czas	pamięć
bbtas_alg	1,5649	32,259	0,1111	117,146	14,09	0,28
bbtas_rug	1,3886	30,658	0,1093	108,770	12,70	0,28
dk14_alg	18,1599	114,452	0,5883	952,803	30,87	0,12
dk14_rug	14,3313	101,726	0,5611	783,057	25,54	0,13
dk15_alg	2,7737	59,381	0,1281	187,828	21,65	0,32
dk15_rug	2,6042	49,013	0,1233	171,541	21,12	0,29
dk17_alg	2,4693	50,257	0,1256	167,406	19,66	0,30
dk17_rug	2,2202	44,168	0,1211	153,531	18,33	0,29
dk27_alg	0,2833	19,627	0,0290	34,180	9,77	0,57
dk27_rug	0,3058	20,574	0,0301	35,662	10,16	0,58
dk512_alg	3,0149	60,065	0,1322	198,117	22,81	0,30
dk512_rug	2,8733	57,699	0,1270	189,412	22,62	0,30
ex7_alg	11,0548	81,778	0,5297	628,850	20,87	0,13
ex7_rug	11,0697	76,639	0,5255	623,297	21,07	0,12
lion_alg	0,2431	15,475	0,0285	28,844	8,53	0,54
lion_rug	0,2126	13,768	0,0258	26,094	8,24	0,53
lion9_alg	11,4135	81,848	0,5244	644,650	21,76	0,13
lion9_rug	9,2599	73,338	0,5167	557,307	17,92	0,13
mod12_alg	1,8136	39,414	0,1148	133,936	15,80	0,29
mod12_rug	1,8436	38,262	0,1164	134,412	15,84	0,28
shift_alg	0,3803	20,288	0,0362	35,701	10,51	0,57
shift_rug	0,3203	20,559	0,0291	36,361	11,01	0,57
tav_alg	4,6491	49,164	0,4759	336,393	9,77	0,15
tav_rug	4,7290	49,176	0,4712	336,400	10,04	0,15
tra04_alg	0,2507	15,917	0,0272	29,189	9,22	0,55
tra04_rug	0,2030	12,808	0,0263	24,652	7,72	0,52
tra11_alg	13,4800	97,680	0,5459	749,260	24,69	0,13
tra11_rug	11,5868	84,817	0,5338	648,334	21,71	0,13
średnia:					16,57	0,31

Ostatnia kolumna w powyższej tabeli zawiera porównanie czasu i ilości zajętej pamięci. Okazuje się, że program napisany według algorytmu 4.3, jest średnio 16 razy szybszy niż według algorytmu 4.2, w którym każdorazowo obliczane są stany węzłów w układzie. Z drugiej strony, zastosowanie tabel czterowartościowych w algorytmie 4.3 wymaga zaangażowania, średnio, ponad 3 razy więcej pamięci operacyjnej. W skrajnym przypadku dla układu *dk14_alg*, który ma 6 wejść i 92 węzły, najwięcej spośród testowanych tu układów, program według algorytmu 4.3 był 30 razy szybszy, ale wykorzystał ponad 8 razy więcej pamięci.

4.2.3. Realizowalność rozkładu prawdopodobieństwa zmian wektorów wejściowych

Rozkład prawdopodobieństwa zmian wektorów wejściowych, do tej pory przedstawiany jako tabela kwadratowa o wymiarach $2^N \times 2^N$, gdzie N to liczba wejść do układu, wymaga nieco uwagi. Wartości prawdopodobieństwa w tej tabeli nie mogą być przypadkowe, aby możliwa była fizyczna realizacja rozkładu zmian wektorów wejściowych.

Rozpatrzmy tabelę, do której wpisano liczby zmian wektorów zamiast prawdopodobieństwa. Aby możliwe było fizyczne wygenerowanie tak zadanych zmian wektorów wejściowych, to dla wektora o danej wartości liczba zmian ze wszystkich innych wartości na daną musi być równa sumie zmian z tej wartości na wszystkie pozostałe. Niech $L_{IN}(v_i)$ oznacza sumę wszystkich zmian z dowolnej wartości wektora wejściowego na wartość v_i , a $L_{OUT}(v_i)$ sumę wszystkich zmian z wartości v_i na pozostałe. Dla wszystkich $V=2^N$ możliwych wektorów N -wejściowego układu, warunek realizowalności rozkładu zmian można zapisać jako:

$$\forall_{v_i \in V} L_{IN}(v_i) = L_{OUT}(v_i) \quad (4.12)$$

Używając prawdopodobieństwa i przyjmując analogiczne oznaczenia powyższy warunek można podobnie zapisać:

$$\forall_{v_i \in V} p_{IN}(v_i) = p_{OUT}(v_i) \quad (4.13)$$

W praktyce spełnienie warunku realizowalności rozkładu oznacza, że zarówno dla liczby zmian wektorów jak i dla prawdopodobieństwa musi zachodzić warunek, że suma wszystkich wartości w i -tym wierszu tabeli musi być równa sumie w i -tej kolumnie. Rozkład zmian wektorów zastosowany w przykładzie 4.1, a przedstawiony na rysunku 4.3 spełnia podany tu warunek realizowalności fizycznej. Natomiast rozkład równomierny niejako z natury jest zawsze realizowalny.

Zaniedbanie powyższego warunku może doprowadzić do błędnych wyników estymacji mocy strat oraz może spowodować wyciągnięcie niewłaściwych wniosków podczas optymalizacji układów logicznych.

4.2.4. Wyniki syntezy układów dwupoziomowych

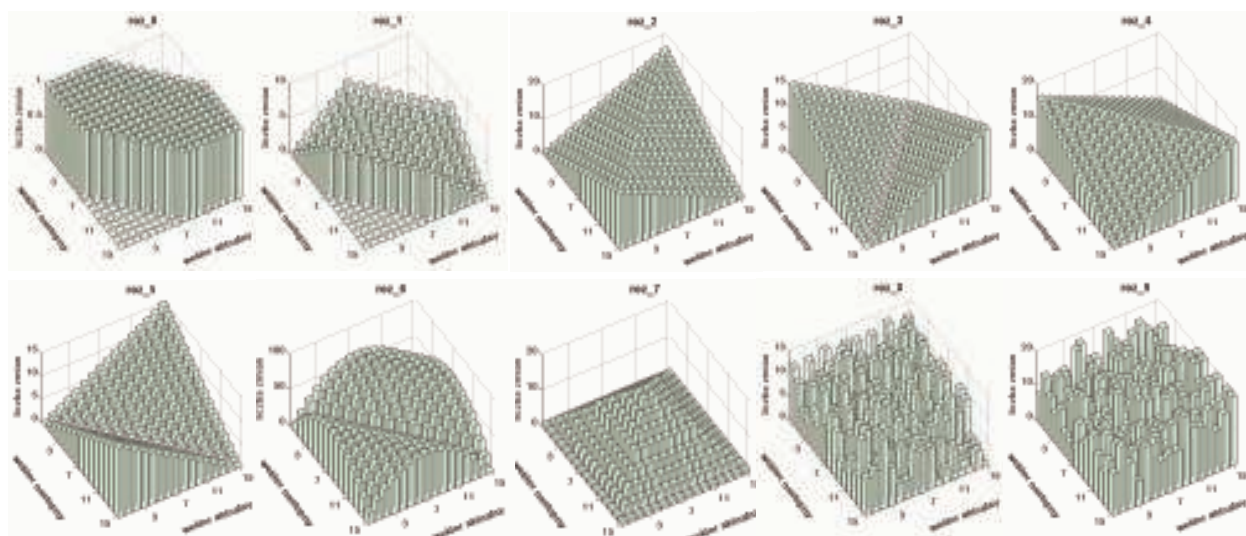
W celu zobrazowania i sprawdzenia efektywności zaproponowanego w punkcie 4.2.1 praktycznego algorytmu syntezy dwupoziomowych układów o zredukowanym poborze mocy, poniżej przedstawiono wyniki syntezy kilkunastu funkcji logicznych. W tabeli 4.4 podano podstawowe informacje o rozważanych funkcjach. Są to funkcje zupełne, których zbiory F wygenerowane zostały losowo.

Tabela 4.4. Funkcje logiczne, dla których zaprojektowano układy dwupoziomowe o obniżonym poborze mocy

nazwa	liczba wej.	zbiór F wektorów „jedynkowych”	współczynnik ON-set
f_1	3	[0, 4, 5]	0,375
f_2	3	[0, 2, 3, 4, 5]	0,625
f_3	3	[1, 2, 3, 4, 5]	0,625
f_4	3	[0, 1, 2, 3, 5, 6]	0,750
f_5	3	[1, 2, 3, 5, 7]	0,625
f_6	3	[1, 2, 3, 4, 5]	0,625
f_7	4	[0, 1, 7, 9, 14]	0,313
f_8	4	[0, 1, 3, 6, 11, 15]	0,375
f_9	4	[3, 4, 6, 10, 11, 12, 13]	0,438
f_10	4	[0, 2, 4, 6, 8, 13, 14]	0,438
f_11	4	[2, 4, 5, 6, 8, 9, 10, 15]	0,500
f_12	4	[3, 5, 6, 8, 11, 12, 14, 15]	0,500
f_13	4	[0, 1, 5, 6, 8, 9, 11, 12, 14, 15]	0,625
f_14	4	[3, 5, 7, 9, 10, 11, 13, 14, 15]	0,563

W powyższej tabeli obok liczby zmiennych wejściowych i numerów wektorów zbiorów F zawarto także wartości współczynnika *ON-set*, tj. ilorazu mocy zbioru F przez liczbę wszystkich wektorów funkcji.

Zgodnie z uprzednio opisanym modelem poboru mocy, aby zaprojektować układ o zredukowanym poborze energii, potrzebna jest informacja o jego aktywności. Dlatego dla powyższych funkcji przygotowano 10 rozkładów zmian wektorów wejściowych. Na rysunku 4.10 przedstawiono szkice tych rozkładów dla układów czterowejściowych. Dla uproszczenia na osi z podano liczby wystąpień zmian wektorów zamiast prawdopodobieństwa. Na ich podstawie, podczas syntezy, obliczone są odpowiednie wartości prawdopodobieństwa. Wszystkie rozkłady spełniają warunki realizowalności opisane w punkcie 4.2.3.

**Rys. 4.10. Rozkłady zmian wektorów wej. zastosowane do syntezy funkcji z Tabeli 4.4**

Wyniki syntezy układów dla funkcji z tabeli 4.4 zawarto w tabelach 4.5a i 4.5b. Wszystkie funkcje były zrealizowane z wykorzystaniem dwu-, trzy-, cztero- lub pięciowejściowych bramek NAND, których parametry energetyczne obliczono wcześniej i zawarto w załączniku B.

Tabela 4.5.a. Wyniki syntezy układów dla funkcji z tabeli 4.4 (część 1)

rozkład:		roz_0			roz_1			roz_2			roz_3			roz_4		
fun.	realiz.	poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]		
		1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma
f_1	min_P	5,70	3,00	8,70	7,45	4,03	11,48	6,22	2,78	9,00	5,73	3,00	8,73	5,23	2,57	7,80
	pok_R	5,28	2,47	7,75	7,01	3,32	10,33	6,55	2,46	9,01	5,88	2,76	8,64	5,28	2,21	7,49
				10,9%						-0,1%			1,1%			3,9%
f_2	m_P	7,38	3,31	10,69	10,98	4,71	15,69	9,47	3,71	13,17	7,60	4,12	11,72	7,48	3,33	10,81
	p_R	7,86	3,12	10,99	10,37	4,05	14,42	9,80	3,48	13,28	8,19	3,86	12,05	7,82	3,10	10,92
				-2,8%			8,1%			-0,8%			-2,8%			-1,0%
f_3	m_P	7,62	3,35	10,97	10,18	4,50	14,69	9,47	3,63	13,09	7,60	4,31	11,91	7,61	3,38	10,99
	p_R	8,00	3,08	11,07	10,45	4,03	14,48	9,80	3,41	13,21	8,16	4,05	12,21	7,92	3,13	11,05
				-1,0%			1,4%			-0,9%			-2,5%			-0,6%
f_4	m_P	6,23	2,98	9,21	7,53	3,54	11,08	5,94	3,77	9,71	5,24	2,81	8,05	5,52	3,10	8,61
	p_R	5,84	2,37	8,21	7,37	2,86	10,22	7,12	3,35	10,46	5,03	2,29	7,32	5,78	2,62	8,40
				10,9%			7,7%			-7,8%			9,1%			2,4%
f_5	m_P	2,58	3,11	5,69	3,53	3,91	7,43	3,25	2,97	6,22	2,02	2,70	4,72	2,53	2,81	5,34
	p_R	3,00	2,78	5,78	3,79	3,35	7,14	3,58	2,66	6,24	2,48	2,46	4,93	2,93	2,54	5,47
				-1,5%			3,9%			-0,4%			-4,6%			-2,3%
f_6	m_P	7,62	3,35	10,97	10,18	4,50	14,69	9,47	3,63	13,09	6,15	3,49	9,65	7,61	3,38	10,99
	p_R	8,00	3,08	11,07	10,45	4,03	14,48	9,80	3,41	13,21	6,61	3,28	9,89	7,92	3,13	11,05
				-1,0%			1,4%			-0,9%			-2,5%			-0,6%
f_7	m_P	12,51	2,65	15,16	14,58	2,92	17,49	14,15	2,79	16,94	12,88	3,24	16,12	12,24	2,66	14,90
	p_R	12,43	2,55	14,98	14,67	2,79	17,46	14,32	2,61	16,93	12,61	3,02	15,63	12,32	2,54	14,85
				1,2%			0,2%			0,1%			3,1%			0,3%
f_8	m_P	11,61	2,77	14,38	13,74	3,11	16,85	13,96	3,14	17,10	12,17	3,50	15,67	11,74	2,89	14,63
	p_R	11,98	2,67	14,65	14,05	2,96	17,01	14,13	2,95	17,08	12,42	3,34	15,76	12,05	2,76	14,82
				-1,9%			-0,9%			0,2%			-0,6%			-1,3%
f_9	m_P	12,20	3,41	15,61	14,27	3,61	17,88	13,97	3,58	17,55	11,81	3,58	15,39	11,96	3,43	15,39
	p_R	12,00	3,25	15,25	14,23	3,42	17,65	14,36	3,42	17,77	11,62	3,43	15,05	11,96	3,26	15,22
				2,3%			1,3%			-1,3%			2,2%			1,1%
f_10	m_P	12,22	3,33	15,55	14,02	3,63	17,65	13,46	3,52	16,99	12,45	3,72	16,18	11,87	3,31	15,19
	p_R	11,90	2,98	14,88	13,95	3,24	17,19	14,25	3,22	17,48	12,07	3,35	15,41	11,95	3,01	14,96
				4,4%			2,6%			-2,9%			4,7%			1,5%
f_11	m_P	15,21	4,12	19,33	18,11	4,53	22,64	17,57	4,15	21,71	14,75	4,39	19,13	15,00	4,04	19,04
	p_R	15,39	3,96	19,35	18,14	4,44	22,59	17,98	4,03	22,01	15,04	4,27	19,32	15,21	3,90	19,11
				-0,1%			0,2%			-1,4%			-1,0%			-0,4%
f_12	m_P	15,75	3,98	19,73	18,00	4,33	22,33	17,33	4,22	21,55	15,87	4,19	20,07	15,30	3,91	19,21
	p_R	15,67	3,96	19,63	18,14	4,32	22,46	17,50	4,09	21,59	15,60	4,06	19,66	15,38	3,84	19,22
				0,5%			-0,6%			-0,2%			2,0%			-0,1%
f_13	m_P	14,84	4,15	18,99	17,12	4,48	21,60	16,68	4,39	21,07	15,30	4,53	19,82	14,56	4,13	18,70
	p_R	15,15	4,01	19,17	17,68	4,34	22,02	17,16	4,17	21,34	15,15	4,24	19,39	14,89	3,96	18,85
				-0,9%			-2,0%			-1,3%			2,2%			-0,8%
f_14	m_P	10,55	3,87	14,41	12,20	4,34	16,55	12,19	4,61	16,80	10,81	4,09	14,90	10,48	3,95	14,43
	p_R	11,46	3,24	14,70	13,39	3,57	16,96	13,43	3,77	17,20	11,32	3,36	14,68	11,58	3,33	14,91
				-2,0%			-2,5%			-2,4%			1,4%			-3,3%

Tabela 4.5.b. Wyniki syntezy układów dla funkcji z tabeli 4.4 (część 2)

rozkład:		roz_5			roz_6			roz_7			roz_8			roz_9		
funk.	realiz.	poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]		
		1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma
f_1	min_P	6,66	2,86	9,52	6,51	3,04	9,55	5,79	2,69	8,48	6,06	2,72	8,78	5,46	2,67	8,12
	pok_R	6,94	2,54	9,48	6,88	2,67	9,55	5,78	2,18	7,96	6,43	2,41	8,85	5,63	2,33	7,96
				0,5%			0,0%			6,2%			-0,8%			2,0%
f_2	m_P	10,15	4,11	14,26	9,84	4,21	14,05	8,75	3,28	12,03	9,15	3,68	12,82	7,88	3,49	11,38
	p_R	10,43	3,83	14,26	10,21	3,91	14,12	8,80	2,95	11,76	9,52	3,44	12,96	8,29	3,30	11,58
				0,0%			-0,5%			2,3%			-1,0%			-1,8%
f_3	m_P	10,02	4,10	14,12	9,84	4,28	14,12	8,94	3,15	12,09	9,10	3,75	12,85	7,95	3,39	11,35
	p_R	10,25	3,79	14,04	10,20	3,98	14,19	9,32	3,03	12,35	9,42	3,50	12,92	8,26	3,16	11,42
				0,6%			-0,5%			-2,1%			-0,5%			-0,6%
f_4	m_P	6,10	4,18	10,28	6,36	3,95	10,31	5,78	3,37	9,15	5,95	3,60	9,55	5,54	3,30	8,84
	p_R	7,67	3,79	11,46	7,40	3,47	10,86	6,46	2,92	9,38	6,95	3,15	10,10	6,03	2,85	8,88
				-11,5%			-5,3%			-2,5%			-5,8%			-0,5%
f_5	m_P	3,49	3,02	6,51	3,33	3,25	6,58	3,02	2,92	5,94	3,09	2,99	6,08	2,65	2,85	5,51
	p_R	3,77	2,71	6,48	3,70	2,90	6,60	3,34	2,61	5,96	3,53	2,74	6,27	2,96	2,54	5,49
				0,5%			-0,2%			-0,2%			-3,2%			0,3%
f_6	m_P	10,02	4,10	14,12	9,84	4,28	14,12	8,94	3,16	12,09	9,10	3,75	12,85	7,95	3,39	11,35
	p_R	10,25	3,79	14,04	10,20	3,98	14,19	9,17	2,95	12,11	9,42	3,50	12,92	8,26	3,16	11,42
				0,6%			-0,5%			-0,2%			-0,5%			-0,6%
f_7	m_P	15,38	3,19	18,57	14,41	3,01	17,42	12,76	2,28	15,04	14,16	2,78	16,94	13,20	3,02	16,22
	p_R	15,39	2,98	18,36	14,54	2,83	17,37	13,07	2,19	15,26	14,29	2,60	16,89	13,32	2,83	16,15
				1,1%			0,3%			-1,5%			0,3%			0,5%
f_8	m_P	15,16	3,58	18,74	14,20	3,38	17,58	12,39	2,52	14,90	14,01	3,35	17,36	12,82	3,19	16,01
	p_R	15,42	3,45	18,87	14,56	3,24	17,80	12,67	2,35	15,03	14,28	3,19	17,47	12,98	3,05	16,03
				-0,7%			-1,3%			-0,8%			-0,7%			-0,1%
f_9	m_P	14,61	3,54	18,16	14,17	3,75	17,92	13,15	3,58	16,72	13,91	3,52	17,43	12,36	3,27	15,62
	p_R	15,19	3,42	18,60	14,56	3,59	18,15	13,26	3,39	16,64	14,25	3,37	17,61	12,52	3,11	15,63
				-2,4%			-1,3%			0,5%			-1,1%			0,0%
f_10	m_P	14,40	3,79	18,19	13,64	3,66	17,30	12,43	3,22	15,65	13,38	3,51	16,90	12,33	3,36	15,69
	p_R	15,41	3,45	18,86	14,40	3,35	17,75	12,90	2,94	15,84	13,84	3,16	17,00	12,80	3,08	15,88
				-3,7%			-2,6%			-1,2%			-0,6%			-1,2%
f_11	m_P	18,55	4,23	22,78	17,81	4,44	22,25	16,42	3,99	20,41	17,43	4,12	21,55	15,84	4,05	19,88
	p_R	18,71	4,11	22,82	18,03	4,33	22,35	16,72	3,89	20,61	17,66	4,02	21,68	16,08	3,95	20,03
				-0,2%			-0,4%			-1,0%			-0,6%			-0,7%
f_12	m_P	18,11	4,34	22,45	17,47	4,27	21,74	16,15	3,96	20,11	17,00	4,10	21,09	16,08	4,10	20,19
	p_R	18,37	4,26	22,62	17,68	4,14	21,83	16,47	3,89	20,36	17,27	3,98	21,25	16,21	3,99	20,20
				-0,8%			-0,4%			-1,2%			-0,8%			-0,1%
f_13	m_P	18,03	4,59	22,62	16,97	4,54	21,51	15,22	4,15	19,36	16,41	4,34	20,75	15,71	4,26	19,97
	p_R	18,18	4,29	22,47	17,35	4,28	21,63	15,97	4,03	20,00	16,86	4,10	20,96	15,90	3,99	19,90
				0,7%			-0,6%			-3,3%			-1,0%			0,4%
f_14	m_P	13,24	5,02	18,26	12,31	4,59	16,90	11,17	4,18	15,35	11,73	4,29	16,02	11,21	4,25	15,46
	p_R	13,90	3,91	17,81	13,52	3,75	17,26	12,79	3,57	16,36	12,97	3,46	16,43	12,29	3,43	15,72
				2,5%			-2,1%			-6,6%			-2,5%			-1,7%

Dla układów, realizujących rozważane funkcje, które zaprojektowano dla każdego rozkładu z rysunku 4.10 obliczono pojemność ekwiwalentną. Rozważono dwie realizacje: układ o minimalnej powierzchni oraz realizację rozłącznego pokrycia funkcji, oznaczone odpowiednio

przez min_P i pok_R w tabelach 4.5.a i 4.5.b. Ponadto wyszczególniono wartości pojemności ekwiwalentnych dla bramek realizujących pierwszy poziom, dla bramki wyjściowej oraz dla całego układu (tj. sumę). Podano również procentową różnicę między pierwszą a drugą realizacją funkcji.

Analizując wyniki w powyższych tabelach, można zaobserwować, że nie zawsze realizacja funkcji z rozłącznym pokryciem daje lepsze rozwiązanie energetyczne. Potwierdza to wnioski wyciągnięte w punkcie 4.2.1. W takich przypadkach pojemność ekwiwalentna bramek realizujących pierwszy poziom układu jest mniejsza dla realizacji o minimalnej powierzchni. Wzrasta wtedy jednak pojemność ekwiwalentna bramki wyjściowej. Rozłączne pokrycie funkcji daje układ, w którym pojemność ekwiwalentna bramki wyjściowej jest mniejsza, ale w większości przypadków zysk ten nie rekompensuje większej pojemności bramek pierwszego poziomu. Zaproponowany algorytm wybiera lepsze spośród pokryć: minimalnego ważonego oraz ważonego rozłącznego. Dzięki temu układy o minimalnej powierzchni też są uwzględniane jako kandydaci podczas poszukiwania najlepszego układu o zredukowanym poborze mocy.

Mogą się zdarzyć przypadki funkcji mających kilka rozwiązań o minimalnej powierzchni. Zaproponowany algorytm, poszukując minimalnego ważonego pokrycia, wybiera najlepsze z nich pod względem poboru mocy. Przykładowo w tabeli 4.6 przedstawiono wyniki obliczeń pojemności ekwiwalentnej dla trzech równoważnych realizacji funkcji f_{11} o minimalnej powierzchni.

Tabela 4.6. Trzy równoważne realizacje funkcji f_{11} o minimalnej powierzchni

rozkład:	roz_0			roz_1			roz_2			roz_3			roz_4		
realizacja	poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]			poj.ekw.[fF]		
min_P	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma	1-st.	wy.	suma
1	15,50	4,08	19,58	18,11	4,53	22,64	17,57	4,14	21,71	15,29	4,41	19,70	15,16	4,02	19,18
2	15,21	4,12	19,33	18,09	4,65	22,73	17,80	4,16	21,96	14,75	4,39	19,13	15,00	4,04	19,04
3	15,56	4,12	19,67	18,29	4,61	22,90	17,63	4,14	21,77	15,15	4,42	19,57	15,15	4,04	19,19
	roz_5			roz_6			roz_7			roz_8			roz_9		
1	18,55	4,25	22,80	17,81	4,44	22,25	16,42	3,99	20,41	17,47	4,19	21,67	15,89	4,05	19,93
2	18,84	4,23	23,07	18,04	4,46	22,50	16,60	4,07	20,67	17,43	4,12	21,55	15,94	4,09	20,03
3	18,55	4,23	22,78	17,88	4,45	22,33	16,54	4,05	20,59	17,60	4,20	21,80	15,84	4,05	19,88

Wyniki przedstawione w tabelach 4.5.a i 4.5.b uwzględniają powyższą uwagę, tak, więc spośród realizacji o minimalnej powierzchni wybrano najlepszą pod względem poboru mocy.

Otrzymane wyniki syntezy układów o zredukowanych stratach mocy pokazują, że możliwa jest redukcja poboru mocy ponad 10%, dzięki zastosowaniu zaproponowanego w niniejszej pracy modelu strat energii i opracowanego algorytmu syntezy układów dwupoziomowych.

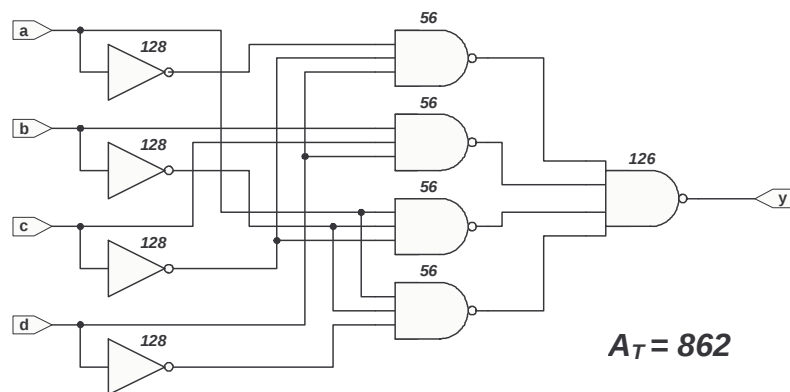
4.3. Redukcja poboru mocy w układach dwupoziomowych przez przekształcenie do układów trójpoziomowych

W rozważaniach dotyczących układów dwupoziomowych, przedstawionych powyżej a także spotykanych w literaturze, zakładano, że dostępne są negacje zmiennych wejściowych dla projektowanego układu. Rzeczywiście jest to dość słuszne założenie, gdyż układy kombinacyjne, jako część układów sekwencyjnych, zwykle współpracują z przerzutnikami, a te najczęściej posiadają dwa wyjścia komplementarne, tzn. wyjście Q i wyjście zanegowane $\bar{Q}$. Zatem dostępne są negacje zmiennych wejściowych. Jednakże mogą się zdarzyć przypadki, gdy trzeba zastosować inwertery na wejściach układu w celu zanegowania zmiennych. Taka sytuacja może wystąpić, na przykład, na głównych wejściach systemu, jeszcze przed przerzutnikami, lub gdy doprowadzenie sygnału zanegowanego będzie się wiązało z prowadzeniem bardzo długich połączeń w układzie scalonym. W związku z powyższym, w tym miejscu, będzie przeanalizowany wpływ inwerterów wejściowych na całkowitą moc strat układu dwupoziomowego.

4.3.1. Redukcja liczby przełączeń – zmniejszanie aktywności układu

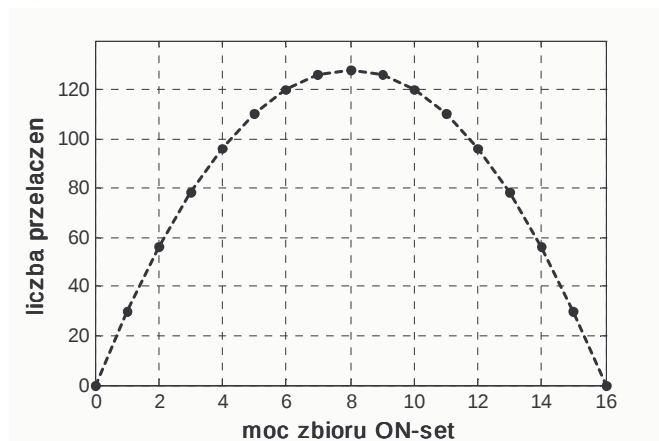
Problem usuwania inwerterów z wejść układów dwupoziomowych, w celu redukcji mocy, zostanie przeanalizowany w oparciu o tradycyjny model poboru mocy opisany w rozdziale pierwszym (punkt 1.1.1). Model ten wykorzystuje klasyczną miarę aktywności układu – aktywność przełączeniową. Będzie wykorzystana całkowita liczba przełączeń węzłów w układzie opisana w [Brz99a], [Brz99b], która po unormowaniu do sumy wszystkich możliwych zmian wektorów wejściowych określa aktywność przełączeniową.

Następująca funkcja zupełna zdefiniowana zbiorem ON-set: $F=\{1, 5, 7, 8, 9, 10, 15\}$ [Brz99a] posłuży do zobrazowania znaczenia inwerterów na wejściu układu. Rysunek 4.11 przedstawia jedną z trzech realizacji o minimalnej powierzchni. Na schemacie zaznaczono liczby przełączeń bramek dla równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych.



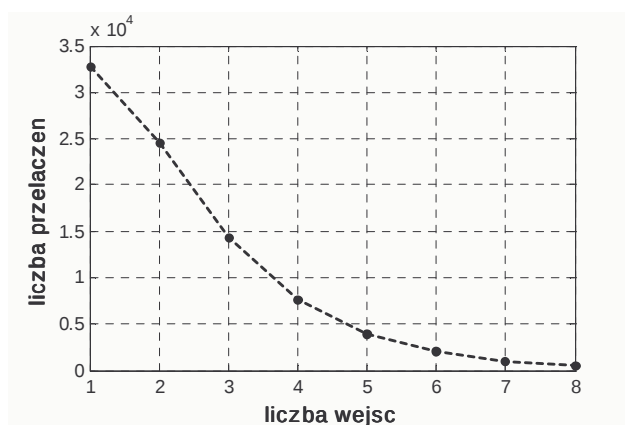
Rys. 4.11. Realizacja (min. pow.) przykładowej funkcji z zaznaczoną liczbą przełączeń bramek

Jak można zauważyć, aktywność inwerterów wejściowych jest największa, gdyż dla rozkładu równomiernego, liczby „zer” i „jedynek” na każdym wejściu układu są równe. Zatem liczba przełączeń opisana równaniem (1.8) jest największa. Równanie (1.8) jest równaniem kwadratowym względem mocy zbioru ON-set. Dla przypadku bramki czterowejściowej ($n = 4$) obliczono aktywności dla wszystkich możliwych wartości mocy zbioru jedynek i przedstawiono na rysunku 4.12. Największa aktywność występuje w przypadku równej mocy zbiorów ON-set i OFF-set. Podobne wnioski były zastosowane w pracy [She92], gdzie wykorzystywano prawdopodobieństwo sygnału.



Rys. 4.12. Liczba przełączeń w funkcji mocy zbioru ON-set dla bramki 4-wejściowej

Równanie (1.8) może być stosowane dla bramek dowolnego poziomu, pod warunkiem znajomości mocy zbiorów ON-set lub OFF-set, o czym pisano w rozdziale pierwszym. Jednak w przypadku inwerterów na wejściu układu mamy do czynienia z sytuacją opisaną równaniem (1.9). Można potraktować inwerter jako bramkę realizującą $(n-1)$ -wymiarową kostkę, której aktywność dla rozkładu równomiernego opisuje właśnie równanie (1.9). Dla przykładu obliczono liczby przełączeń dla bramek o liczbie wejść od 1 do 8 będących bramkami pierwszego poziomu w układzie ośmiowejściowym. Największa wartość występuje dla bramki jednowejściowej, czyli inwertera (rys. 4.13).



Rys. 4.13. Liczba przełączeń bramki pierwszego poziomu w funkcji jej wejść k w 8-wej. układzie

W przypadku inwertera na wejściu i równomiernego rozkładu zmian wektorów wejściowych zarówno równanie (1.8) jak i (1.9) opisuje tę samą sytuację, gdyż wtedy moce zbiorów ON-set i OFF-set są równe. Jest to przypadek największej liczby przełączeń, którą można zapisać jako:

$$A_{i_{max}} = 2^{2n-1} \quad (4.14)$$

Podobnie najmniejsza wartość liczby przełączeń, która ma miejsce w przypadku mocy zbioru ON-set równej 1 lub $2^n - 1$, wynosi:

$$A_{i_{min}} = 2^{n+1} - 2 \quad (4.15)$$

Powyższe rozważania, wykorzystujące liczbę przełączeń, można uogólnić stosując prawdopodobieństwo sygnału i aktywność przełączeniową. Wystarczy podzielić liczbę przełączeń przez liczbę wszystkich zmian wektorów wejściowych. Dla układu 4-wejściowego i równomiernego rozkładu liczba wszystkich możliwych zmian wektorów wej. wynosi 2^{2n} , czyli 256, ponieważ równomierny rozkład wymaga, aby każda zmiana wektorów wystąpiła przynajmniej jeden raz.

Z powyższych rozważań wynikają następujące wnioski. Dla redukcji aktywności układu a więc konsumpcji mocy, dobrze by było, aby moc zbioru ON-set dla dowolnej bramki w układzie była możliwie najmniejsza lub największa. Wniosek ten dotyczy dowolnego rozkładu prawdopodobieństwa. Jednakże z punktu widzenia inwerterów na wejściach układu ważniejszy jest drugi wniosek. Otóż, ze wzoru (1.9) oraz rysunku 4.13 wynika, że im więcej wejść do bramki, tym mniejsza liczba przełączeń, przynajmniej dla równomiernego rozkładu zmian wektorów wejściowych. Zatem wniosek dla syntezy układów o obniżonym poborze mocy jest taki, aby usuwać z wejść układu inwertery, ogólnie bramki o małej liczbie wejść.

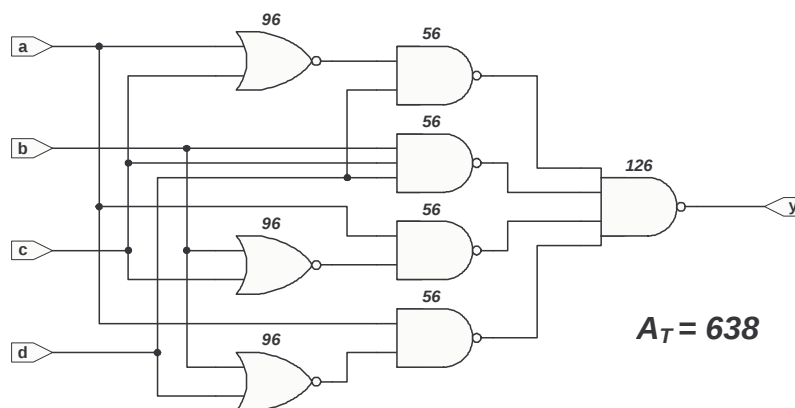
Wracając do przykładowej funkcji, której schemat realizacji z bramkami NAND przedstawiono na rysunku 4.11, poniżej zapisano równanie opisujące tę realizację.

$$y = \overline{\overline{acd}} \cdot \overline{\overline{bcd}} \cdot \overline{\overline{abc}} \cdot \overline{\overline{abd}} \quad (4.16)$$

Stosując prawo de Morgana względem zanegowanych zmiennych wejściowych, można wyeliminować inwertery wejściowe, otrzymując następujące równanie:

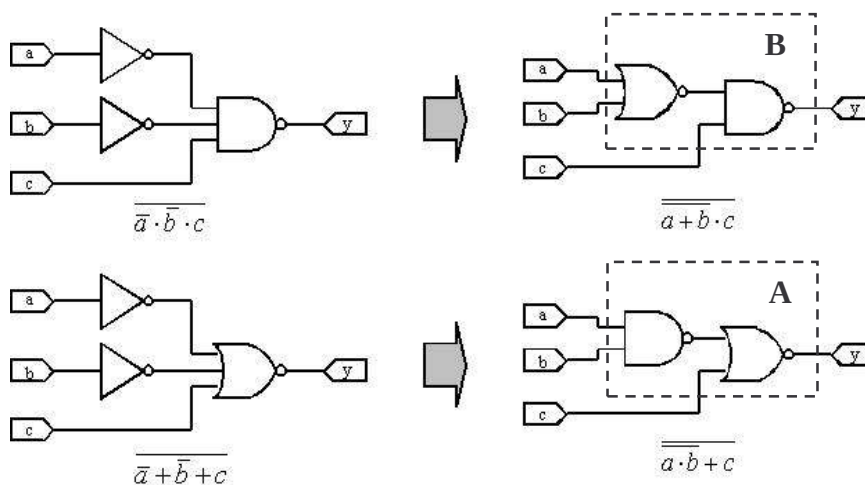
$$y = \overline{\overline{a + c}} \cdot \overline{\overline{d + bcd}} \cdot \overline{\overline{a + b + c}} \cdot \overline{\overline{a + b + d}} \quad (4.17)$$

W realizacji funkcji opisanej powyższym równaniem inwertery wejściowe zastąpiono bramkami NOR, otrzymując układ trójpoziomowy. W ten sposób wyeliminowano bramki o największej aktywności. Na rysunku 4.14 przedstawiono schemat układu oraz zaznaczono liczby przełączeń poszczególnych bramek. Całkowita liczba przełączeń A_T rozwiązania o minimalnej powierzchni (rys 4.11) wynosi 862, a po zamianie inwerterów (rys. 4.14) 638. W efekcie eliminacji inwerterów uzyskano redukcję aktywności układu o 26%.



Rys. 4.14. Realizacja przykładowej funkcji – zredukowana aktywność

Uogólniając spostrzeżenia z powyżej przedstawionego przykładu, można powiedzieć, że proponowana metoda redukcji pobieranej mocy na drodze redukcji aktywności układu polega na zastępowaniu inwerterów wejściowych bramkami NOR w przypadku realizacji funkcji opisanej za pomocą kanonicznej postaci dysjunkcyjnej lub bramkami NAND dla postaci koniunkcyjnej. Wykorzystuje się przy tym prawo de Morgana. Na rysunku 4.15 przedstawiono graficznie opisywaną zasadę syntezy. Połączenie bramek NOR i NAND nazwano elementem *B*, a połączenie odwrotne elementem *A* [Brz00b], [Brz00c].



Rys. 4.15. Ogólna zasada projektowania układów o zmniejszonej aktywności

Ponieważ, w ogólnym przypadku, k inwerterów jest zastępowanych przez k -wejściową bramkę pierwszego poziomu, to zakładając równomierny rozkład zmian wektorów wejściowych i korzystając z równania (1.9), można obliczyć zysk redukcji aktywności układu po takiej zamianie. Równanie (1.9) bezpośrednio opisuje liczbę przełączeń k -wejściowej bramki, a aktywność k inwerterów będzie wyrażać się następująco:

$$A_i(k_NOT) = k(2^{2n-1+1} - 2^{2n-2-1+1}) = k 2^{2n-1} \quad (4.18)$$

Zatem zysk aktywności układu można zdefiniować jako stosunek liczby przełączeń k -wejściowej bramki do liczby przełączeń k inwerterów danej powyższym równaniem, co można zapisać jako:

$$\frac{A_i(k, n)}{A_i(k_NOT)} = \frac{k(2^{2n-k+1} - 2^{2n-2k+1})}{k 2^{2n-1}} = \frac{4}{k}(2^{-k} - 2^{-2k}) \quad (4.19)$$

Wynika z tego, że im większa liczba k , czyli większa liczba inwerterów została zastąpiona jedną bramką, to mniejszy jest stosunek dany powyższym równaniem i mniejsza liczba przełączeń bramki zastępującej inwertery, a więc większy zysk na takiej zamianie.

Jednakże proponowana metoda zamiany inwerterów przez bramki NOR lub NAND ma pewne ograniczenia. Po pierwsze, w klasycznym układzie pojedyncze inwertery mogą być wielokrotnie wykorzystane dla różnych bramek realizujących implikanty funkcji, a bramka zastępująca kilka inwerterów staje się mniej uniwersalna. Może się zdarzyć, że dla funkcji o większej liczbie implikantów liczba bramek zastępujących inwertery będzie większa niż liczba inwerterów tak, że pomimo zmniejszenia aktywności bramek całkowita liczba zmian tych bramek będzie większa niż inwerterów.

Rozwiązaniem tego problemu może być dekompozycja bramki zastępującej inwertery tak, aby bramki o mniejszej liczbie wejść mogły być wykorzystane dla innych implikantów. Zatem komplikuje się nieco problem poszukiwania pokrycia funkcji. Trzeba, bowiem wybrać do realizacji funkcji takie implikanty, które będą miały jak najwięcej wspólnych zmiennych zanegowanych. Poniższe równania obrazują proces dekompozycji bramek zastępujących inwertery wejściowe:

$$\begin{aligned} \overline{abcde} &= a \overline{b+c} \overline{d+e} \\ \overline{abcd} &= \overline{abcc} \overline{d} = a \overline{b+c} \overline{c+d} \end{aligned} \quad (4.20)$$

W pierwszym przypadku bramkę czterowejściową NOR zastępujemy dwoma dwuwejściowymi, które mogą być wykorzystane dla innych implikantów funkcji. W drugim równaniu bramkę trójwejściową również zastępujemy dwoma dwuwejściowymi. Bramka NOR ($\overline{b+c}$) będzie teraz wspólna dla obydwu równań.

Drugim problemem, jaki może wystąpić podczas przekształcania układów dwupoziomowych na trójpoziomowe według proponowanej metody, jest obecność pojedynczych zmiennych zanegowanych w iloczynach opisujących implikanty funkcji (por. (4.16)). Wtedy nie można zastosować bezpośrednio prawa de Morgana tak jak w (4.17), ponieważ jest tylko jedna zmienna zanegowana. Owszem, nadal możliwe jest zastosowanie prawa de Morgana, ale nieco inaczej i powoduje to komplikacje układu. Co więcej, zwiększa się aktywność układu, chociaż najbardziej aktywny inwerter z wejścia będzie usunięty.

Dla przykładu, niech jeden z implikantów pewnej funkcji będzie opisany iloczynem:

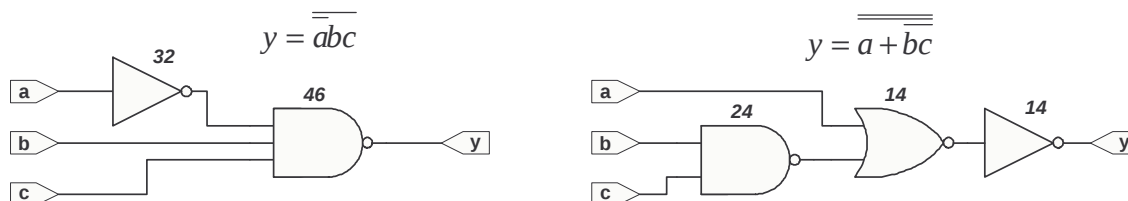
$$\overline{a}bc \quad (4.21)$$

Do jego realizacji potrzebny jest jeden inwerter i jedna trójwejściowa bramka NAND. Grupując zmienne niezanegowane i teraz stosując prawo de Morgana, iloczyn można zapisać jako sumę:

$$a + \overline{b}\overline{c} \quad (4.22)$$

Obecnie realizacja wymaga jednej bramki NAND i jednej OR. Jednakże w technologii CMOS bramka OR to połączenie bramki NOR i inwertera. W efekcie takiego postępowania nastąpiło przesunięcie inwertera z wejścia bramki realizującej dany implikant na jej wyjście. Pozostaje jeszcze pytanie o całkowitą liczbę przełączeń bramek w takim układzie.

Analizując aktywność poszczególnych bramek realizujących formułę (4.22) w technologii CMOS, można zauważyć, że nie ma już bramki o największej liczbie przełączeń, czyli inwertera, na wejściu. Jednakże całkowita liczba przełączeń takiego układu będzie większa niż dla układu według równania (4.21). Na rysunku 4.16 przedstawiono realizację obu rozwiązań oraz zaznaczono liczby przełączeń bramek zakładając układ trójwejściowy. W pierwszym wypadku całkowita liczba przełączeń wynosi 46 a w drugim 52. Zatem nastąpił wzrost aktywności układu.

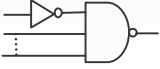
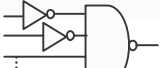
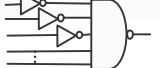


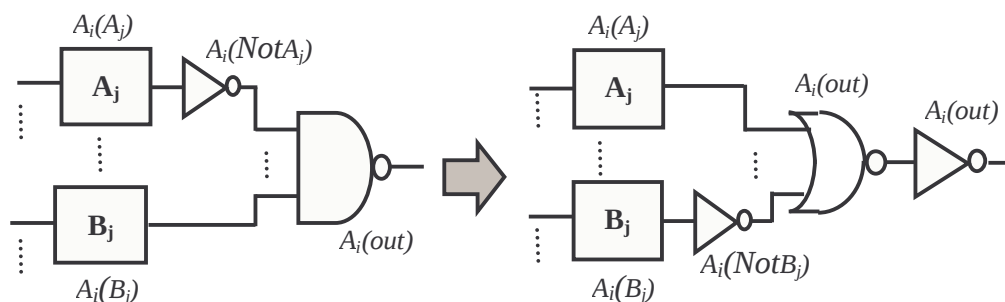
Rys. 4.16. Przesuwanie inwertera z wejścia w głąb układu

W tabeli 4.7 zamieszczono porównanie klasycznej realizacji o minimalnej powierzchni oraz nowej o zredukowanej aktywności dla implikantów dowolnej funkcji logicznej [Brz99b]. Proponowany sposób przesuwania inwerterów przynosi zysk już w przypadku funkcji czterech zmiennych, co pokazują wyniki dla pierwszego przypadku w tabeli.

W wyniku przesuwania inwerterów z wejść układu może wystąpić sytuacja, gdy projektowany układ będzie zbudowany z dużej liczby elementów A lub B (z rys. 4.15), na wyjściach których będą występować inwertery. Wtedy może okazać się celowe dalsze przesuwanie inwerterów polegające na zamianie bramki wyjściowej na dualną (NAND $\rightarrow$ NOR, NOR $\rightarrow$ NAND) wyposażoną w inwerter na wyjściu oraz na każdym z jej wejść. Teraz pomiędzy elementami A i B z inwerterami na wyjściach a bramką wyjściową występują po dwa inwertery, które można usunąć. W efekcie pozostają nieliczne inwertery wewnątrz układu oraz inwerter na wyjściu układu. Zasadę przesuwania inwerterów z wyjść elementów A lub B na wyjście układu przedstawiono na rysunku 4.17 [Brz00b], [Brz00c], [Brz01].

Tabela 4.7. Porównanie klasycznej realizacji implikantów funkcji z proponowaną o zredukowanej liczbie przełączeń wykorzystującą prawa de Morgana

Lp.	realizacja klasyczna			realizacja o zredukowanej aktywności		
	implikant	schemat	liczba przełączeń A_i	formuła	realizacja	liczba przełączeń A_i
1	$\overline{x_1}x_2 \cdots x_n$		$2^{2n-1} + 2^{n+1} - 2$ $A_i(n=2) = 14$ $A_i(n=3) = 46$ $A_i(n=4) = 158$ $A_i(n=5) = 574$	$\overline{x_1} + x_2 \cdots x_n$		$2^{n+3} - 12$ $A_i(n=2) = 20$ $A_i(n=3) = 52$ $A_i(n=4) = 116$ $A_i(n=5) = 244$
2	$\overline{x_1}\overline{x_2}x_3 \cdots x_n$		$2^{2n} + 2^{n+1} - 2$ $A_i(n=2) = 22$ $A_i(n=3) = 78$ $A_i(n=4) = 286$	$\overline{x_1} + \overline{x_2} + x_3 \cdots x_n$		$2^{2n-1} - 2^{2n-3} + 2^{n+1} - 2$ $A_i(n=2) = 12$ $A_i(n=3) = 38$ $A_i(n=4) = 126$
3	$\overline{x_1}\overline{x_2}\overline{x_3}x_4 \cdots x_n$		$3 \cdot 2^{2n-1} + 2^{n+1} - 2$ $A_i(n=3) = 110$ $A_i(n=4) = 414$	$\overline{x_1} + \overline{x_2} + \overline{x_3} + x_4 \cdots x_n$		$2^{2n-2} - 2^{2n-5} + 2^{n+1} - 2$ $A_i(n=3) = 28$ $A_i(n=4) = 86$
...						
m	$\overline{x_1} \cdots \overline{x_m} x_{m+1} \cdots x_n$		$m \cdot 2^{2n-1} + 2^{2n-k+1} - 2^{2n-2k+1}$	$\overline{x_1} + \cdots + \overline{x_m} + x_{m+1} \cdots x_n$		$2^{2n-m+1} - 2^{2n-2m+1} + 2^{n+1} - 2$
...						
n-1	$\overline{x_1} \cdots \overline{x_{n-1}} x_n$		$(n-1)2^{2n-1} + 2^{n+1} - 2$ $A_i(n=2) = 14$ $A_i(n=3) = 78$ $A_i(n=4) = 414$	$\overline{x_1} + \cdots + \overline{x_{n-1}} + x_n$		$3 \cdot 2^{n+1} - 10$ $A_i(n=2) = 14$ $A_i(n=3) = 38$ $A_i(n=4) = 86$



Rys. 4.17. Przesuwanie inwerterów na wyjście układu

Zastosowanie takiego przesuwania inwerterów będzie opłacalne, jeśli w jego efekcie aktywność całego układu się zmniejszy. Sytuację tę opisuje równanie:

$$A_{profit} = \sum_{j=1}^{j_A} A_i(NotA_j) - \sum_{j=1}^{j_B} A_i(NotB_j) - A_i(out) \quad (4.23)$$

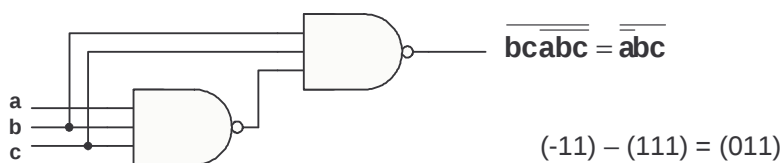
Oczywiście tak określony zysk w aktywności powinien być większy od zera, aby było uzasadnione stosowanie przesuwania inwerterów na wyjście układu.

Zaproponowana metoda przesuwania inwerterów z wejść układu w głąb i ostatecznie na wyjście nie jest zbyt efektywną i w wielu przypadkach może nie przynieść zysku. Lepsze wyniki w redukcji aktywności układu może przynieść zastosowanie metody TANT.

W latach 60-tych i 70-tych szeroko zajmowano się układami logicznymi zbudowanymi tylko z bramek NAND lub NOR. Świadczą o tym początkowe prace np.: Hellermana [Hel63] lub Elisa [Eli65]. Natomiast pełny opis metody TANT można znaleźć w [Gim67], [Cha70]. Nazwa

metody pochodzi od pierwszych liter nazwy angielskiej: *Three-Level AND-NOT Network with True* (trójpoziomowy układ z bramkami NAND bez zmiennych zanegowanych). Można również znaleźć polskojęzyczne prace na ten temat [Tra76], [Tra86], gdzie rozważano realizację układów zarówno z bramkami NAND jak i NOR. Zatem, mając na uwadze przedstawione wcześniej wnioski o potrzebie eliminacji zmiennych zanegowanych, i w efekcie inwerterów z wejścia układu można przypuszczać, że zastosowanie metody TANT do syntezy układów o obniżonym poborze mocy przyniesie korzyści.

Chcąc scharakteryzować metodę TANT jednym zdaniem, trzeba powiedzieć, że polega ona na odejmowaniu zbiorów. Wprowadza się tu pojęcie odejmowania kostek, które reprezentują zbiory mintermów lub makstermów funkcji, mamy wtedy odpowiednio realizację układu tylko z bramkami NAND lub NOR. Dla przykładu, aby odjąć jeden z wektorów od kostki (-11), która składa się z dwóch wektorów: (011) oraz (111), należy zastosować układ taki jak na rysunku 4.18. Operację realizowaną przez ten układ, potraktowany jako element pierwszego poziomu w układzie, można zapisać jako: $(-11) - (111) = (011)$.



Rys. 4.18. Układ realizujący odejmowanie wektora (111) od kostki (-11)

Działanie odejmowania realizowane przez powyższy układ można przedstawić w tabeli Karnough'a jak na rysunku 4.19. Ponieważ metoda TANT polega na odejmowaniu elementów zbiorów, to, jak można zauważyć, aby otrzymać wektor (011) od kostki (-11), czyli zbioru, w którym elementami są wektory (011) i (111), wystarczy odjąć jeden z wektorów zawierający niepotrzebny element, czyli (111). Jak widać z rysunku 4.19 są cztery takie zbiory, które można wyrazić za pomocą następujących kostek: (111), (11-), (1-1) i (1--). Oczywiście interesujące są tylko zbiory, które są kostkami, bo te można łatwo zrealizować za pomocą bramek logicznych.

a \ bc	00	01	11	10
	0	0	1	0
0	0	0	1	0
1	0	0	0	0

(-11)

(111)

Rys. 4.19. Tabela Karnough'a reprezentująca odejmowanie kostek w układzie z rys. 4.18

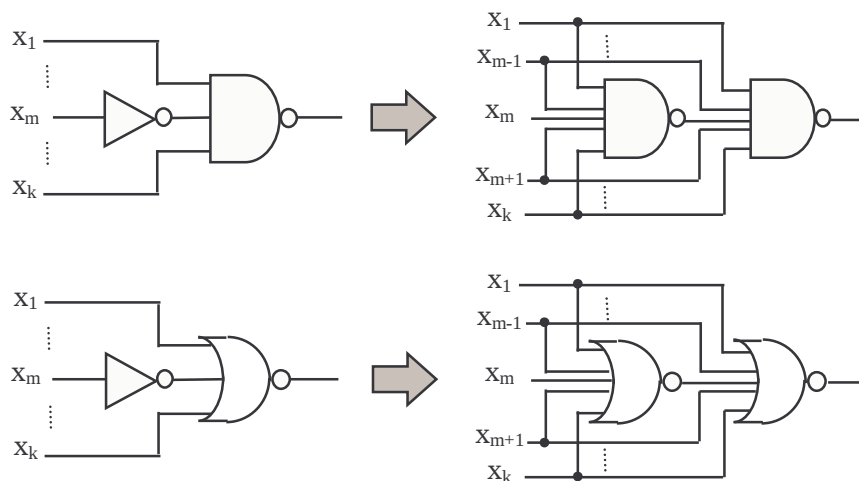
Z punktu widzenia eliminacji zmiennych zanegowanych powyżej opisany proces odejmowania zostanie odwrócony. Chodzi o to, aby za pomocą odejmowania kostek przedstawić kostkę, której realizacja wymaga negacji – „0” w kostce dla realizacji z bramkami NAND i „1” dla NOR.

Na początku dana kostka wejściowa k funkcji jest dzielona na dwie części. Następnie należy wyznaczyć kostki odjemną i odjemnik. W związku z tym wprowadza się pewne oznaczenia wykorzystane w ogólnym równaniu opisującym odejmowanie kostek. Symbol $\hat{A}$ rozumiemy jako wszystkie kostki, w których zera albo jedynki zastąpiono kreskami. Na przykład zapis $(\hat{a}\hat{b})$ oznacza kostki: (ab) , $(-b)$, $(a-)$, $(--)$. Człon A tak dobieramy, aby składał się tylko z „1” i „-” dla realizacji z bramkami NAND lub tylko z „0” i „-” dla realizacji z NOR. Natomiast pozostałe elementy kostki wejściowej tj. „0” dla NAND i „1” dla NOR oznaczamy przez α . Człon A oraz α są fragmentami kostek i oznaczają niepełne iloczyny (lub sumy dla NOR), zatem dana kostka $k = \alpha A$. Potrzebne jest jeszcze dopełnienie, w sensie zbiorów, członu α . Wyznaczenie dopełnienia sprowadza się do obliczenia negacji członu α , i tak też będzie oznaczane, tj. $\bar{\alpha}$. Zatem, można zapisać ogólne równanie metody TANT:

$$(\alpha A) = (-A) - (\bar{\alpha} \hat{A}) \quad (4.24)$$

Wracając do przykładowego odejmowania z rysunku 4.18, można teraz zapisać kompletne odejmowanie jako: $(011) = (-11) - (1\hat{1}\hat{1})$. Podstawiając za symbol „1” jedynki lub kreski można otrzymać wszystkie wspomniane wcześniej, możliwe przypadki odejmowania kostek dające wynik (011) , również przypadek z rysunku 4.18.

Na rysunku 4.20 przedstawiono zasadę eliminacji inwerterów z wejść układu wykorzystując metodę TANT dla realizacji funkcji z bramkami NAND i NOR.



Rys. 4.20. Eliminacja inwerterów wejściowych z zastosowaniem metody TANT

Z punktu widzenia wcześniejszych rozważań wydaje się, że dla syntezy układów dwupoziomowych o obniżonym poborze mocy najkorzystniejsze jest zastosowanie równania (4.24) zastępując symbol „1” jedynkami. Wtedy bramka pierwszego poziomu, realizująca odjemnik, ma największą z możliwych liczbę wejść. Należy przypomnieć, że im większa liczba wejść do

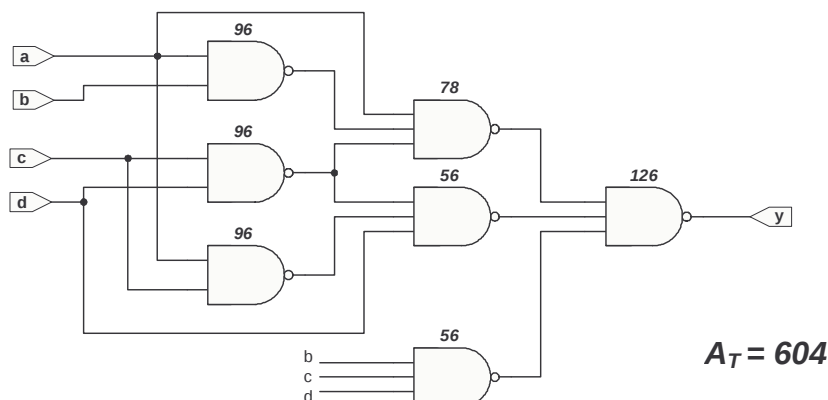
bramki, tym mniejsza aktywność (rys. 4.13). Z drugiej strony zmniejszenie liczby bramek również może spowodować redukcję aktywności układu. Aby zastosować kilkakrotnie tę samą bramkę, może się okazać, że trzeba zmniejszyć liczbę jej wejść, zastępując część symboli „1” kreskami, czyniąc bramkę bardziej uniwersalną, ale o większej aktywności.

W celu zobrazowania efektywności opisywanej metody dla przykładowej funkcji z początku niniejszego podpunktu, której realizacja o minimalnej powierzchni została przedstawiona na rysunku 4.11, zastosowano metodę TANT i wyniki przedstawiono poniżej.

Realizację danej funkcji można opisać następującymi kostkami: $\{(-111), (10-0), (100-), (0-01)\}$. Kostka pierwsza nie wymaga inwerterów do realizacji, więc pozostanie bez zmian. Kostki druga i trzecia mają jedynkę na tej samej pozycji, więc człon A będzie dla nich taki sam: $A_{12} = (1---)$, a dla kostki trzeciej: $A_3 = (---1)$. Człony $\alpha_{12} = \{(-0-0), (-00-)\}$ i $\alpha_3 = (0-0-)$. Korzystając z prawa de Morgana można łatwo obliczyć dopełnienie członów α , otrzymując: $\overline{\alpha_{12}} = \{(-1--), (--11)\}$ i $\overline{\alpha_3} = \{(1---), (--1-)\}$. Teraz dla rozważanej funkcji można zapisać równania według (4.24):

$$\begin{aligned} \begin{pmatrix} 10-0 \\ 100- \end{pmatrix} &= (1---) - \begin{pmatrix} \hat{1}1-- \\ \hat{1}-11 \end{pmatrix} \\ (0-01) &= (---1) - \begin{pmatrix} 1--\hat{1} \\ --1\hat{1} \end{pmatrix} \end{aligned} \quad (4.25)$$

Zgodnie z wcześniejszymi wnioskami należy wszystkie „1” w członach $\hat{A}$ zamienić na „1”, ale jeśli w drugim wierszu pierwszego równania element „1” zamienimy kreską, to powstała kostka będzie identyczna jak dla drugiego równania ($--11$). Ta kostka będzie więc wykorzystana dwukrotnie. Na podstawie (4.25) i uwzględniając powyższą uwagę można zapisać równanie opisujące realizację zadanej funkcji tylko z bramkami NAND: $y = \overline{\overline{a} \overline{ab} \overline{cd} \overline{d} \overline{ac} \overline{cd} \overline{bcd}}$. Na rysunku 4.21 przedstawiono schemat układu. Podobnie jak poprzednio, i tym razem zaznaczono liczby przełączeń poszczególnych bramek dla równomiernego rozkładu zmian wektorów



Rys. 4.21. Realizacja przykładowej funkcji metodą TANT

wejściowych. Całkowita liczba przełączeń w tym układzie wynosi 604 i jest 30% mniejsza niż w układzie o minimalnej powierzchni (rys. 4.11). Jednakże, gdyby zastosować metodę TANT niezależnie dla każdej kostki, pomijając części wspólne, otrzymany układ będzie nieco większy.

Liczba przełączeń wyniesie 736, a opisuje go równanie: $y = a \overline{ab} \overline{ad} a \overline{ab} \overline{ac} d \overline{ad} \overline{cd} \overline{bcd}$.

Jak pokazuje powyższy przykład, zastosowanie metody TANT umożliwia nie tylko usunięcie pojedynczych inwerterów z wejść układu, ale także eliminację wszystkich inwerterów wejściowych. W tym konkretnym przypadku udało się zmniejszyć także liczbę bramek drugiego poziomu, co ma dodatkowy wpływ na zmniejszenia aktywności układu. Jednak w większości przypadków tak nie musi być i uzyskana redukcja aktywności układu będzie porównywalna z wynikami metody usuwania inwerterów z zastosowaniem praw de Morgana (rys. 4.15).

Dotychczasowe rozważania na temat potrzeby usuwania inwerterów z wejść układu były przeprowadzone przy założeniu równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych. W klasycznym modelu strat mocy wyraża się to prawdopodobieństwem sygnału równym $\frac{1}{2}$. Nasuwa się pytanie, czy powyższe rozumowanie będzie słuszne dla dowolnych wartości prawdopodobieństwa sygnału.

W związku z tym należy przeanalizować aktywność przełączeniową węzła, zdefiniowaną równaniem (1.6), zamiast liczby przełączeń. Korzystając z prawdopodobieństwa sygnału oraz równań (1.3) i (1.6), można obliczyć aktywność przełączeniową dwóch inwerterów s_{2Not} jako:

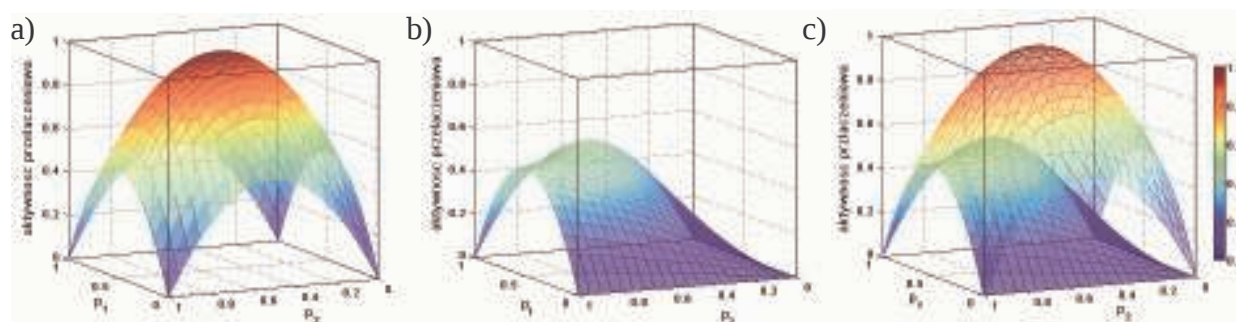
$$s_{2Not} = 2s_{Not} = 2p_1 - 2p_1^2 + 2p_2 - 2p_2^2 \quad (4.26)$$

gdzie: p_1 – prawdopodobieństwo sygnału na wejściu pierwszego inwertera, a p_2 – drugiego. Wartości p_1 i p_2 mogą się zmieniać od 0 do 1. Proponowana metoda eliminacji inwerterów polega na zastąpieniu ich bramkami NAND lub NOR, zatem aby porównać efektywność takiej zamiany należy obliczyć aktywność tych bramek, korzystając z równań (1.3) i (1.6):

$$s_{Nand2} = 2p_1p_2 - 2p_1^2p_2^2 \quad (4.27)$$

$$s_{Nor2} = 2(1-p_1)(1-p_2) - 2(1-p_1)^2(1-p_2)^2 \quad (4.28)$$

Analizując powyższe równania można zauważyć, że największa wartość aktywności dwóch inwerterów wynosi 1 a bramek dwuwejściowych 0,5. W celu porównania wartości aktywności przełączeniowej bramek narysowano wykresy $s = f(p_1, p_2)$. Na rysunku 4.22 przedstawiono kolejno, aktywność przełączeniową dla dwóch inwerterów (a), dla bramki NAND (b), oraz dla lepszego porównania oba wykresy razem (c). Jak można zaobserwować na tym rysunku, dla dowolnych wartości prawdopodobieństwa sygnałów p_1 i p_2 aktywność dwóch inwerterów jest większa niż bramki NAND. Powierzchnia obrazująca aktywność bramki NAND znajduje się



Rys. 4.22. Aktywność przełączeniowa w funkcji prawdopodobieństwa sygnału dla: dwóch inwerterów (a), dwuwejściowej bramki NAND (b) i porównanie obu przypadków (c)

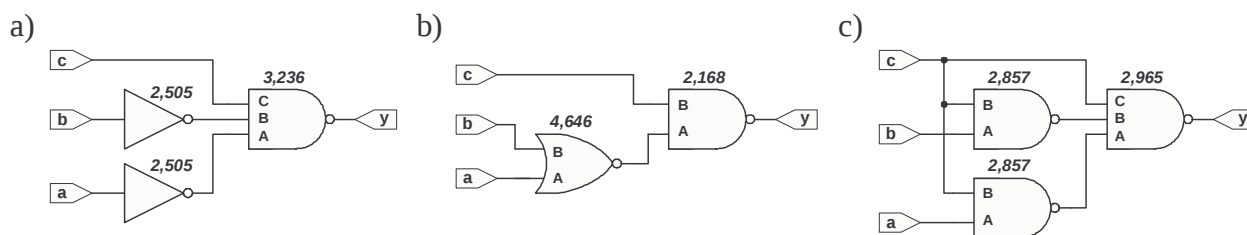
„pod” powierzchnią dla dwóch inwerterów (rys. 4.22.c). Podobna sytuacja ma miejsce dla bramki NOR. Dzięki dualizmowi bramek NAND i NOR wykres bramki NOR można otrzymać z wykresu NAND przez symetryczne odbicie względem przekątnej.

Podsumowując należy stwierdzić, że usuwanie inwerterów z wejść układów logicznych jest bardzo istotne z punktu widzenia poboru mocy. Potwierdza to przeprowadzona analiza dla równomiernego rozkładu zmian wektorów wejściowych do układu. Podobne wnioski można wyciągnąć z analizy układu dla dowolnych wartości prawdopodobieństwa sygnałów dokonanej na końcu niniejszego paragrafu. Ponadto, taki inverter o dużej aktywności może być jednym z elementów rozpraszających najwięcej energii w układzie. Lokalne maksima temperaturowe spowodowane przez elementy pobierające najwięcej mocy są przyczyną naprężeń termicznych w układzie scalonym, które mogą prowadzić do uszkodzeń.

4.3.2. Usuwanie inwerterów wejściowych w kontekście nowego modelu strat energii

Analiza wpływu inwerterów wejściowych na pobór mocy przedstawiona w poprzednim paragrafie opierała się na modelu tradycyjnym i skupiała się tylko na redukcji aktywności układu. W kontekście poboru mocy bardzo ważne są parametry energetyczne bramek. Zwykle są one wyrażane przez pojemność węzłową, a w nowym modelu poboru mocy przez pojemność ekwiwalentną bramek. Ponieważ nowy model dokładniej opisuje wpływ zależności między sygnałami wejściowymi bramek na pobór mocy, co jest odzwierciedlone przez wartości pojemności ekwiwalentnych, należy zweryfikować wyniki i wnioski z poprzedniej analizy właśnie z nowym modelem.

Zaproponowana metoda redukcji aktywności układu polega na eliminacji inwerterów z wejść układu przez zastąpienie ich bramkami NAND lub NOR wykorzystując prawa de Morgana albo przez zastosowanie metody TANT. Na rysunku 4.23 przedstawiono trzy układy realizujące przykładowy iloczyn $\overline{a}bc$, który może być fragmentem większej funkcji. Pierwszy układ to klasyczna realizacja dwupoziomowa o minimalnej powierzchni (rys. 4.23.a), drugi – to układ



Rys. 4.23. Realizacja kostki (001) w układach: klasycznym (a), o zredukowanej aktywności (b) i po zastosowaniu metody TANT (c)

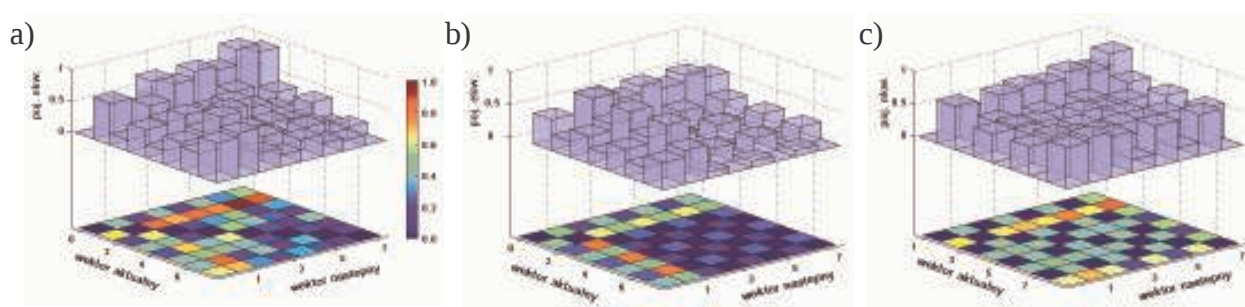
ze zredukowaną aktywnością po zastąpieniu inwerterów bramką NOR (b) i trzeci – to układ otrzymany po zastosowaniu metody TANT (c). Następnie dla tych układów oszacowano pobór energii bazując na nowym modelu strat mocy dla równomiernego rozkładu prawdopodobieństwa zmian wektorów wejściowych. Otrzymano następujące wartości pojemności ekwiwalentnej po zastosowaniu, opisanej w rozdziale trzecim, optymalizacji kolejności wejść do bramek: 8,246fF, 6,814fF i 8,679fF. Dodatkowo na rysunku 4.23 wpisano obliczone wartości pojemności ekwiwalentnych dla poszczególnych bramek. W drugim układzie pojemność ekwiwalentna zmniejszyła się o ponad 17%. Zatem powyższe wyniki potwierdzają celowość usuwania inwerterów z wejść układu.

Przyczyna takich wyników tkwi nie tylko w zmniejszeniu aktywności układu, ale mają w tym udział również parametry energetyczne bramek. Na początku niniejszego rozdziału obliczono wartości pojemności ekwiwalentnej dla bramek pierwszego poziomu przy założeniu równomiernego rozkładu zmian wektorów, a wyniki zebrano w tabeli 4.1. Analizując układy z rysunku 4.23 można zauważyć, że w pierwszym układzie są dwa inwertery o łącznej pojemności ekwiwalentnej równej 5,010fF (por. tab. 4.1). Zastąpienie tych inwerterów jedną dwuwejściową bramką NOR o pojemności ekwiwalentnej 4,646fF jest atrakcyjne i daje zysk prawie 8%. Ponadto w drugim układzie zastosowano dwuwejściową bramkę NAND zamiast trójwejściowej o gorszych parametrach energetycznych. W tym przypadku pojemność ekwiwalentna tej bramki zmniejszyła się o 30%, chociaż w tabeli 4.1 różnica jest mniejsza.

W układzie drugim i trzecim bramki NAND drugiego poziomu mają mniejsze wartości pojemności ekwiwalentnej niż by to wynikało z tabeli 4.1. Przyczyną jest fakt, że bramki pierwszego poziomu mają wpływ na sposób sterowania następnych bramek, jakkolwiek w przypadku inwerterów na wejściach i równomiernego rozkładu wynik dla bramki drugiego poziomu pozostaje bez zmian – pierwszy układ.

W celu przeanalizowania wpływu bramek pierwszego poziomu na sposób sterowania następnych bramek i w efekcie na ich pojemność ekwiwalentną wartości składowe pojemności ekwiwalentnej zostaną przedstawione w funkcji sposobów sterowania całego układu. Sposób

zobrazowania pojemności ekwiwalentnych bramek drugiego poziomu będzie analogiczny do opisanego w punkcie 4.2.1 i wykorzystanego na rysunku 4.5. Niezależnie do liczby wejść do bramek wszystkie tabele pojemności będą miały wymiary 8×8 , gdyż rozpatrywany układ ma trzy wejścia. Do tabel wpisano te wartości pojemności C_{Lall} bramek dwu- i trójwejściowej NAND, na które wskazują wartości binarne obliczone dla każdego wektora wejściowego do układu zgodnie ze strukturą układu. Np. dla układu trzeciego (rys. 4.23.c) wektor wejściowy $abc=0\hat{1}1$ spowoduje sposób sterowania wyjściowej bramki NAND: $ABC=1\hat{1}1$. Rysunek 4.24 przedstawia pojemność ekwiwalentną bramek drugiego poziomu w funkcji sposobów sterowania układu. Wartości pojemności unormowano do największej dla bramki NAND trójwejściowej, tj. 9,06fF w celu łatwiejszego porównania.



Rys. 4.24 Unormowana pojemność ekwiwalentna dla bramek drugiego poziomu z rys. 4.22

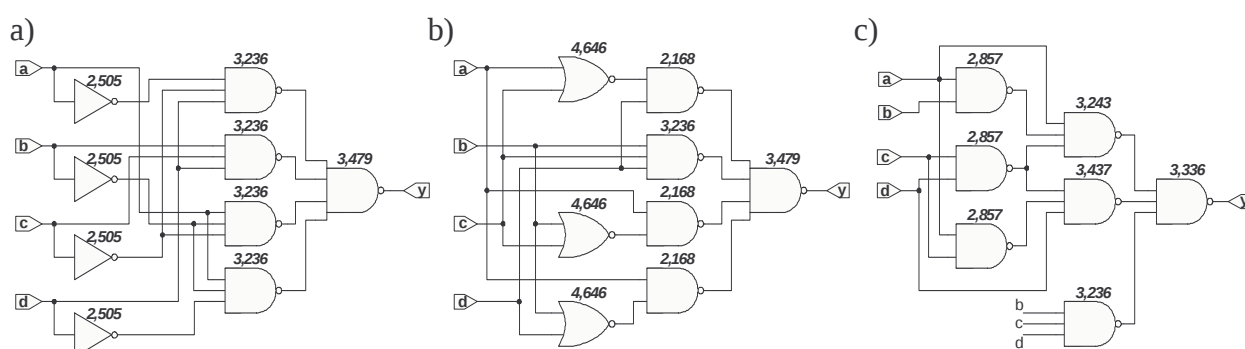
Tabela dla pierwszego, klasycznego układu (rys. 4.23.a) zawiera wszystkie wartości pojemności ekwiwalentnej bramki NAND3 z oryginalnej tabeli C_{Lall} . Jedynie zmieniło się ich położenie, gdyż negacje zmiennych spowodowały konieczność przeliczenia indeksów i wyznaczenie nowych miejsc w tabeli na rysunku 4.24.a. W trzecim przypadku nastąpiło wyeliminowanie niektórych wartości pojemności ekwiwalentnej, natomiast inne powtórzyły się kilkakrotnie. Spowodowane jest to tym, że różne sposoby sterowania układu powodują taki sam sposób sterowania bramki. W efekcie suma pojemności dla bramki NAND3 w tym układzie jest mniejsza niż w układzie pierwszym. Dla rozkładu równomiernego możemy zsumować wartości pojemności, gdyż prawdopodobieństwo zmian wektorów wejściowych jest jednakowe. W przypadku drugiego układu, gdzie bramka dwuwejściowa ma cztery razy mniej sposobów sterowania niż trójwejściowa, wartości pojemności z oryginalnej tabeli C_{Lall} dla NAND2 nie powtórzyły się czterokrotnie. To oznacza, że tu również pewne sposoby sterowania zostały wyeliminowane. Ponadto wartości pojemności ekwiwalentnej dla dwuwejściowej bramki NAND są mniejsze niż dla trójwejściowej (por. rys. 4.6). W efekcie bramka wyjściowa w drugim układzie ma najmniejszą wartość całkowitej pojemności ekwiwalentnej.

Przedstawioną powyżej analizę własności energetycznych trzech logicznie równoważnych układów z rysunku 4.23 można rozszerzyć dla przypadku dowolnego rozkładu zmian wektorów

wejściowych do układu. Pomocne będą przy tym tabele przedstawione graficznie na rysunku 4.24. Wartości z tych tabel należy pomnożyć przez wartości prawdopodobieństwa zmian wektorów wejściowych do układu i wtedy będą to składowe całkowitej pojemności ekwiwalentnej bramek.

Dla dowolnego rozkładu zmian wektorów wejściowych należy także, zgodnie z nowym modelem, obliczyć pojemności ekwiwalentne dla bramek pierwszego poziomu (NOT, NOR2 i NAND2 z rys. 4.23). Wtedy bazując na otrzymanych wynikach można podjąć właściwą decyzję co do zamiany inwerterów wejściowych.

Dla oceny efektywności i sprawdzenia celowości usuwania inwerterów, oszacowano pobór mocy układów realizujących przykładową funkcję z poprzedniego paragrafu (rysunki: 4.11, 4.14 i 4.21) z zastosowaniem nowego modelu strat energii i przy założeniu równomiernego rozkładu zmian wektorów wejściowych. Dzięki temu zostaną teraz uwzględnione parametry energetyczne bramek, czyli pojemność ekwiwalentna, gdyż poprzednio analizowano tylko aktywność układu. Pierwszy układ to realizacja o minimalnej powierzchni, drugi to układ o aktywności zredukowanej przez zastąpienie inwerterów bramkami NOR a trzeci to układ zaprojektowany z zastosowaniem metody TANT. Otrzymano następujące wartości całkowitej pojemności ekwiwalentnej dla poszczególnych układów: 26,443fF, 27,157fF i 21,823fF, po optymalizacji kolejności wejść do bramek stosownie do opisu w rozdziale trzecim. Na rysunku 4.25 przedstawiono schematy układów z wartościami pojemności ekwiwalentnych dla bramek.



Rys. 4.25. Pojemności ekwiwalentne [fF] dla układów realizujących przykładową funkcję o: minimalnej powierzchni (a), zredukowanej aktywności (b) i po zastosowaniu metody TANT (c)

Okazuje się, że drugi układ (rys. 4.25.b), w którym aktywność wyrażona całkowitą liczbą przełączeń, poprzednio była zredukowana o 26%, jest gorszym rozwiązaniem niż układ o minimalnej powierzchni (a). Całkowita pojemność ekwiwalentna jest większa o 2,7%. Natomiast trzeci układ (c) jest najlepszym rozwiązaniem. Pojemność ekwiwalentna jest mniejsza o ponad 17% w porównaniu z pierwszym układem. Tak dobry wynik uzyskano dzięki zmniejszeniu liczby bramek NAND na drugim poziomie układu. Stosując metodę TANT niezależnie dla

każdej kostki otrzymany układ będzie większy i jego pojemność ekwiwalentna wyniesie 27,037fF. Ten wynik jest zbliżony do wyniku dla układu drugiego, gdyż cztery inwertery zostały zastąpione czterema bramkami NAND. Podobna sytuacja miała miejsce w poprzednim paragrafie, gdy analizowano aktywność układów.

Przyczyny takich wyników należy upatrywać w tym, że w drugim rozwiązaniu zastosowano bramki NOR o gorszych parametrach energetycznych, niż bramki NAND. Zastąpienie dwóch inwerterów jedną dwuwejściową bramką NOR jest atrakcyjne, o czym już pisano. Jednakże w tym przykładzie cztery inwertery zastąpiono trzema bramkami NOR i wtedy już bilans pojemności ekwiwalentnej wypada niekorzystnie dla NOR. Podobne wnioski wyciągnięto w poprzednim paragrafie odnośnie wzrostu liczby przełączeń spowodowanej nadmierną liczbą bramek NOR w układzie. Z drugiej jednak strony, zastosowanie jednej bramki zamiast dwóch inwerterów ma korzystny wpływ na następne bramki.

Kilka z funkcji, które były przedmiotem testowania algorytmu syntezy przedstawionego w punkcie 4.2, przeanalizowano pod kątem problemu inwerterów wejściowych. Opis funkcji przedstawiono w tabeli 4.4. Realizacje o minimalnej powierzchni przedstawione w tabeli 4.5 uzupełniono o inwertery wejściowe. Następnie zgodnie z przedyskutowanymi metodami eliminacji inwerterów przeprojektowano układy uzyskując trzy nowe realizacje. Po pierwsze, wykorzystując prawo de Morgana zastąpiono inwertery bramkami NOR. Następne to układy zaprojektowane z zastosowaniem metody TANT. Trzecim rozwiązaniem są układy mieszane, które powstały przez usunięcie pojedynczych inwerterów z układów uzyskanych na podstawie prawa de Morgana wykorzystując metodę TANT. Dla wszystkich układów oszacowano aktywność przełączeniową oraz całkowitą pojemność ekwiwalentną dla równomiernego rozkładu zmian wektorów wejściowych oraz dla kilku z rysunku 4.10. Wyniki obliczeń zebrano w tabeli 4.8.

W układach o minimalnej powierzchni ($m_{pow.}$) dla funkcji f_8 i f_{11} nie występują pojedyncze negacje w iloczynach, zatem nie ma realizacji mieszanej (*mix*). W przypadku funkcji f_7 , f_8 i f_{10} w niektórych iloczynach wszystkie zmienne są zanegowane, wtedy zastosowanie metody TANT opisaną równaniem (4.24) staje się niemożliwe. Należy, korzystając z praw de Morgana, zastosować bramkę NOR z inwerterem – przesuwanie inwertera na wyjście bramki. W związku z tym, dla funkcji f_{10} układ otrzymany za pomocą metody TANT jest identyczny jak układ mieszany.

Analizując wyniki zawarte w tabeli 4.8 można zauważyć, że zaproponowane metody eliminacji inwerterów zawsze dają znaczne zmniejszenie aktywności układów. W niektórych przypadkach nawet ponad 50%. Najlepszym rozwiązaniem najczęściej jest układ mieszany.

Tabela 4.8. Aktywność przełączeniowa i pojemność ekwiwalentna układów zrealizowanych klasycznie i po usunięciu inwerterów wejściowych

nazwa fun.	aktywność przełączeniowa				pojemność ekw. ukl. [fF]			
	m_pow.	deMorg.	TANT	mix	m_pow.	deMorg.	TANT	mix
	równ.				równ.			
f_7	3,10	2,37	2,19	1,81	26,25	32,57	31,32	29,46
f_8	3,24	2,21	2,53	–	26,35	30,81	33,94	–
f_9	3,37	3,52	2,55	2,18	26,45	32,59	27,24	25,37
f_10	3,42	2,95	2,35	2,35	26,29	34,06	32,90	32,90
f_11	3,74	3,24	2,86	–	30,52	34,81	27,54	–
f_12	3,74	3,68	2,93	2,71	30,52	36,27	31,03	30,38
f_13	3,86	3,80	2,98	3,05	30,13	35,75	28,90	33,09
	roz_0				roz_0			
f_7	2,89	2,16	2,02	1,63	24,53	29,34	28,54	26,59
f_8	2,93	1,75	2,13	–	23,75	25,20	29,66	–
f_9	3,23	3,22	2,36	2,03	24,98	29,60	25,32	23,13
f_10	3,22	2,75	2,18	2,18	24,92	32,52	31,37	31,37
f_11	3,60	3,10	2,70	–	28,95	32,53	25,37	–
f_12	3,61	3,65	2,71	2,64	29,24	35,23	28,77	29,09
f_13	3,65	3,49	2,87	2,84	28,36	32,96	27,60	30,76
	roz_1				roz_1			
f_7	3,35	2,52	2,29	1,88	28,56	35,03	33,47	31,60
f_8	3,41	2,05	2,46	–	27,91	30,87	35,24	–
f_9	3,66	3,58	2,69	2,25	28,94	34,18	29,62	26,90
f_10	3,70	3,11	2,47	2,47	28,72	37,05	35,94	35,94
f_11	4,09	3,47	3,14	–	33,70	37,95	30,05	–
f_12	4,05	3,98	2,97	2,87	33,39	39,78	33,06	32,97
f_13	4,14	3,97	3,17	3,18	32,66	38,29	31,44	35,71
	roz_3				roz_3			
f_7	3,09	2,71	2,51	2,10	25,30	32,70	31,69	29,54
f_8	3,23	2,36	2,83	–	24,84	28,57	33,52	–
f_9	3,22	3,48	2,65	2,25	24,56	30,60	26,38	24,15
f_10	3,34	3,13	2,52	2,52	25,35	34,56	32,94	32,94
f_11	3,66	3,31	2,93	–	28,87	33,10	26,29	–
f_12	3,62	3,81	2,99	2,79	29,26	36,06	29,93	29,95
f_13	3,78	3,90	3,10	3,20	29,00	35,17	28,76	32,39
	roz_7				roz_7			
f_7	2,88	1,79	1,71	1,33	25,06	28,16	27,54	25,57
f_8	2,96	1,62	1,84	–	24,93	26,64	29,41	–
f_9	3,45	3,42	2,31	2,06	26,74	31,85	25,95	24,49
f_10	3,31	2,59	2,05	2,05	25,67	31,72	30,99	30,99
f_11	3,69	3,07	2,66	–	30,43	33,99	26,33	–
f_12	3,76	3,60	2,72	2,63	30,51	35,74	29,98	29,79
f_13	3,74	3,36	2,80	2,69	29,39	33,13	27,80	31,16
	roz_8				roz_8			
f_7	3,25	2,31	2,17	1,81	27,72	33,58	32,62	30,74
f_8	3,50	2,49	2,55	–	28,46	34,05	35,83	–
f_9	3,57	3,61	2,53	2,23	28,20	34,18	28,21	26,62
f_10	3,60	2,94	2,38	2,38	27,67	34,80	34,17	34,17
f_11	3,96	3,44	2,85	–	32,44	37,17	28,42	–
f_12	3,90	3,71	2,84	2,73	31,87	37,31	31,52	31,24
f_13	4,02	3,80	2,96	3,04	31,52	36,65	29,42	34,15

Po uwzględnieniu własności energetycznych bramek usuwanie inwerterów nie zawsze spowodowało redukcję poboru mocy. W przypadku analizowanych funkcji, zastosowanie prawa de Morgana skutkowało zaprojektowaniem układów, które ani raz nie były najlepszym rozwiązaniem. Stało się tak za sprawą bramek NOR, które mają gorsze parametry energetyczne niż bramki NAND. Dlatego też w przypadku funkcji f_7 , f_8 i f_{10} , których realizacje bez inwerterów wejściowych zawierają bramki NOR z przesuniętymi inwerterami, użycie proponowanych metod nie dało redukcji pobieranej mocy. Zastosowanie bramek o gorszych parametrach energetycznych oraz zwiększenie całkowitej liczby bramek spowodowało pogorszenie własności energetycznych całego układu.

Inaczej sprawa wygląda w przypadku funkcji f_9 , dla której najlepszą realizacją jest układ mieszany. Składa się on z jednej bramki NOR, zastępującej dwa inwertery wejściowe. Ponadto, trzykrotnie zastąpiono pojedyncze inwertery bazując na metodzie TANT. Jedną bramkę NAND wykorzystano dwukrotnie. To spowodowało, że ekwiwalentna pojemność całego układu jest najmniejsza dla rozważanych rozkładów zmian wektorów wejściowych.

Najlepsze wyniki uzyskano dla funkcji f_{11} zrealizowanej metodą TANT sięgające ponad 13% dla rozkładu *roz_7*. Układ jest zbudowany tylko z bramek NAND.

Uogólniając, zaproponowane metody eliminacji inwerterów wejściowych przynoszą korzyść w postaci zmniejszenia poboru mocy, jeśli w układzie będzie możliwie jak najmniej bramek NOR oraz bramki zastępujące inwertery będą wielokrotnie wykorzystywane.

Drugim wnioskiem wypływającym z powyższych wyników jest to, że do oceny własności energetycznych układów logicznych nie wystarczy tylko analiza aktywności przełączeniowej układu. Bardzo ważne są parametry energetyczne poszczególnych bramek użytych do budowy układu cyfrowego. Dopiero połączenie tych dwóch parametrów daje bliższy rzeczywistemu obrazowi właściwości energetycznych układu.

ROZDZIAŁ 5.

Podsumowanie

Pomimo bardzo dużej liczby publikacji na temat redukcji strat energii w układach cyfrowych, można zauważyć, że właściwie wszystkie metody bazują na bardzo zbliżonym sposobie modelowania strat mocy. Niektóre prace uwzględniają, co prawda, wzajemne zależności sygnałów, bądź to wejściowych, bądź wewnątrz układu, lecz tylko w sensie korelacji niesionej informacji, co ma wpływ na aktywność przełączeniową. Na domiar tego, sygnały wejściowe bramek są rozpatrywane niezależnie. Zaniedbuje się jednoczesny wpływ wszystkich sygnałów wejściowych na sposób przełączania bramki, a zatem i na wartość poboru mocy. Zakłada się, że każde przełączenie bramki jest równoważne co do ilości rozpraszanej energii. Nieuwzględnianie tych zjawisk skłoniło autora do podjęcia tematyki związanej z estymacją i optymalizacją poboru mocy w układach cyfrowych zbudowanych w technologii CMOS.

W pracy przedstawiono nowe możliwości redukcji energii pobieranej przez układ cyfrowy CMOS dzięki uwzględnieniu pewnych zaniedbywanych dotychczas zjawisk występujących w bramkach. Przedstawiona analiza zachowania się bramki CMOS podczas przełączania, w kontekście poboru mocy, pozwoliła na uwzględnienie wpływu sposobu sterowania bramki – zmiany wektora wejściowego – na jej parametry energetyczne. To w efekcie zaowocowało opracowaniem stosownego modelu energetycznego bramek.

Dla nowego, zaproponowanego modelu strat mocy opracowano metody szacowania jego parametrów. Opisano sposób wyznaczania pojemności ekwiwalentnej – jako parametru charakteryzującego własności energetyczne bramek – oraz przedyskutowano otrzymane wyniki.

Nowy model wymaga nowej miary aktywności układu, uwzględniającej sposób sterowania bramek. Informacja o aktywności układu musi pozwalać na wzięcie pod uwagę subtelnych zależności pomiędzy pobieraną mocą a sygnałami wejściowymi bramki. W związku z tym zaproponowano taką miarę – prawdopodobieństwo sposobu sterowania bramki. Opracowano kilka metod obliczania nowej miary aktywności układu i porównano ich efektywność.

Uwzględnienie zaniedbywanych zjawisk, czyli zwiększenie dokładności modelu zawsze powinno owocować lepszymi wynikami oceny parametrów obiektu, wykonanej w oparciu o taki model. Rzeczywiście, tak też się stało w przypadku zastosowania nowego modelu do estymacji

poboru mocy układów cyfrowych. Uzyskane w *Rozdziale 2* wyniki dla kilkudziesięciu układów testowych są o kilka procent lepsze niż w przypadku zastosowania modelu tradycyjnego.

Analiza otrzymanych parametrów energetycznych kilku bramek, zaprojektowanych w celach testowych, sugerowała możliwości uzyskania zysku w konsumpcji energii przez układy cyfrowe na drodze optymalizacji kolejności wejść do bramek. Opracowano stosowną procedurę wybierającą najlepszą kolejność podłączenia wejść do bramek. Uzyskane wyniki testów, przedstawione w *Rozdziale 3*, również pokazują możliwość kilkuprocentowej oszczędności mocy pobieranej przez układ.

Zastosowana metoda optymalizacji układu przez dobór kolejności wejść do bramek nie w pełni wykorzystuje możliwości nowego modelu. Optymalizacja ma miejsce na ostatnim etapie projektowania, już po syntezie logicznej i mapowaniu układu. Dlatego podjęto próbę wykorzystania informacji zawartych w nowym modelu już na początkowym etapie syntezy logicznej. W efekcie opracowano algorytm syntezy układów dwupoziomowych o obniżonym poborze mocy. Ponadto, przedyskutowano problem wpływu inwerterów na wejściach takiego układu na jego straty mocy. Zaproponowano stosowne metody eliminujące inwertery z wejść wykorzystując tradycyjny i nowy model opisujący konsumpcję energii.

Otrzymane wyniki syntezy kilkudziesięciu funkcji dla różnych rozkładów prawdopodobieństwa zmian wektorów wejściowych pokazują możliwość zmniejszenia mocy pobieranej przez układ. Wyniki zebrane w punkcie 4.2.4 wskazują na ponad 10% zysk w porównaniu z układami o minimalnej powierzchni. Również eliminacja inwerterów może przynieść efekty w postaci redukcji strat mocy. Wyniki testów dla kilku układów pokazują, że usunięcie inwerterów, może zmniejszyć aktywność przełączeniową nawet o ponad 50%. Zysk w redukcji poboru mocy jest mniejszy, ale przekracza 13% dla niektórych przypadków.

Podsumowując wyniki tej rozprawy, można stwierdzić, że pomimo bardzo dużej liczby prac poświęconych tematyce estymacji i redukcji poboru mocy w cyfrowych układach CMOS, możliwe było opracowanie nowych, efektywnych metod redukcji poboru mocy. Niniejszym można stwierdzić, że postawiona na wstępie teza mówiąca, że:

Istnieje możliwość minimalizacji strat energii w układach logicznych CMOS, jeśli zostanie wykorzystana informacja o prawdopodobieństwie zmian wszystkich sygnałów, traktowanych jako wektor, na wejściach układu i poszczególnych bramek

została udowodniona, co pokazano analitycznie oraz z wykorzystaniem licznych benchmarków.

Zaproponowane w pracy metody redukcji strat mocy w cyfrowych układach kombinacyjnych wykonanych w technologii CMOS dają dobre efekty pod warunkiem znajomości prawdopodobieństwa zmian wektorów wejściowych. W praktyce, bez tych informacji jest prawie niemożliwe zaprojektowanie układu o bardzo małym poborze mocy. Nie można zaprojektować układu uniwersalnego. Być może, założenie równomiernego rozkładu zmian wektorów wejściowych mogłoby być pewnym przybliżeniem.

Autor niniejszej rozprawy skoncentrował się na rozważaniach wokół dynamicznych pojemnościowych strat energii powstałych podczas przełączania bramki CMOS. Uwzględniając zanedbywane dotychczas zależności pomiędzy sygnałami wejściowymi, które zamodelowano w odpowiedni sposób, wskazano na możliwości zmniejszenia strat mocy oraz opracowano stosowne metody redukcji energii pobieranej przez układ cyfrowy.

Niniejszą pracę należy traktować jako próbę zwrócenia uwagi na nowe możliwości redukcji strat mocy w cyfrowych układach CMOS. Jest ona również uzupełnieniem powszechnie stosowanego modelu strat mocy i zwiększeniem jego dokładności. Ale nie tylko, gdyż w pracy zaproponowano metody projektowania układów prowadzące do redukcji poboru mocy. Pomimo bardzo szybkiego rozwoju technologii mikroelektronicznych w kierunku nanotechnologii przedstawione metody i opracowane algorytmy pozostają ważne, a zmiany technologiczne powodują jedynie zmiany ilościowe w otrzymanych wynikach. Ponadto, przedstawione metody i algorytmy mogą stanowić dobrą podstawę do zbudowania większego systemu, środowiska do projektowania kombinacyjnych układów cyfrowych o obniżonym poborze mocy w technologii CMOS.

Załączniki

Załącznik A

Kody źródłowe wybranych programów powstałych podczas prac badawczych zapisane w języku Matlab.

1. Obliczanie pojemności ekwiwalentnej na podstawie wyników symulacji

```
%*****
% gece.m - Gate Equivalent Capacitance Extraction:
%   za pomocą fun. "read_csv" wczytuje dane z CSV i przygotowuje plik „matlabowy” z danymi
%   następnie oblicza średni prąd i poj. ekwiwalentną, wyniki zapisuje do pliku
% ZMIENNE:
% okres   - okres przebiegu wejściowego
% pli_csv - nazwa pliku CSV z wynikami symulacji bramki
% DZ      - struk. - zawiera: nagłówek i dane wczytane z pliku CSV, te same dane podzie-
%           lone na okresy, liczbę wydzielonych okresów i liczbę kolumn
% CEK     - struk. - zawiera: pole z wektorami dziesiętnymi dla określonej kolejności oraz
%           pola z obliczoną poj. ekw., nazwy pól takie jak nazwy prądów wczytane ze SPICE
% Cekw    - struk. z posortowanymi poj. ekw., nazwy pól takie jak nazwy prądów ze SPICE
% wek_dz  - macierz kolumnowa z wektorami dziesiętnymi dla kolejnych okresów
% li_wej  - liczba wejść do bramki/układu
%-----
function Cekw=gece
%----- OPERACJE WSTĘPNE -----
sw=input('Czy wczytać wyniki symulacji i "obrobić" pliki CSV (t/n)? ','s');
if sw=='t' | sw=='T'
%----- WCZYTANIE DANYCH z plików CSV -----
fprintf('\n\tWCZYTYWANIE wyników symulacji z plików CSV\n')
[DZ,k_pli_DZ]=read_csv; % "k_pli_DZ" pusta, jeśli NIE zapisano w pliku celi "DZ"
elseif sw=='n' | sw=='N'
fprintf('Podaj nazwę pliku z "obrobionymi" wynikami z symulacji ("DZ")\n')
k_pli_DZ=[input(' z ewentualnym podkatalogiem: ','s'),'_DZ'];
load(k_pli_DZ);
else
fprintf('\n Błędna nazwa pliku/katalogu!\n')
return
end
%----- OBLICZANIE PRĄDÓW ŚREDNICH i POJ. EKWIWALENTNEJ -----
fprintf('\n\tObliczanie ŚREDNICH PRĄDÓW i POJ. EKWIWALENTNEJ\n')
[CEK,li_wej]=prad_sr2(k_pli_DZ,DZ); % wywołanie funkcji obliczającej średni prąd
%----- SORTOWANIE obliczonej pojemności ekwiwalentnej
pola=fieldnames(CEK); % nazwy prądów dla których obliczono poj.ekw.
ind=find((strcmp('jednostka',pola) | strcmp('wek_dz',pola))==0); % indeksy nazw pól oprócz
% "jednostka" i "wek_dz", czyli tylko pola z pojemnością
li_wek=size(CEK.wek_dz,1); % liczba wektorów (równa liczbie okresów symulacji)
CEK.wek_dz=CEK.wek_dz+1; % bo indeksy od 1 a wektory są od 0 do 2^li_wej
for p=1:size(ind,1) % po liczbie prądów, "ind" zaw. indeksy nazw pól, które są nazwami pr.
C_t=zeros(2^li_wej); % tymczasowa macierz na poj. ekw.
for x1=2:li_wek
C_t(CEK.wek_dz(x1-1),CEK.wek_dz(x1))=getfield(CEK,{1},pola{ind(p)},{x1,1});
end
for x2=1:(2^li_wej) % zerowanie przekątnej,
C_t(x2,x2)=0; % od 1 do rozmiar macierzy C_t
end
eval(['Cekw.',pola{ind(p)}, '=C_t;'])
end
%*****
% read_csv.m - czyta dane z pliku CSV i przygotowuje plik „matlabowy”, wykorzyst. w "gece"
%-----
function [DZ,gdzie]=read_csv % funkcja zwraca wczytane i podzielone dane w str. DZ oraz
% miejsce gdzie je zapisano
pli_csv=input('Podaj nazwę pliku z wynikami symulacji bramki (z podkat.): ','s'),'.csv';
okres=input('Podaj okres zmian wektorów wej. [ns]: '); % okres przebiegu w [ns]
DZ.okres=okres; % okres w [ns]
ix=find(pli_csv=='/' | pli_csv=='\'); % nazwa wczytywanego pliku bez katalogu
```

```

DZ.plik=pli_csv(ix(size(ix,2))+1:size(pli_csv,2));
%----- CZYTANIE -----
[DZ.dane,DZ.nag,DZ.licz_kol]=czytanie(pli_csv); % funkcja "czytanie" wczytuje dane z pliku
% CSV i zamienia je na cele, zmienia jedn. czasu na [ns] i zwraca 3 elementy struktury DZ
%----- PODZIAŁ DANYCH -----
[DZ.dane_zb,DZ.li_ok] = podzial2(DZ.dane,DZ.licz_kol,okres); % za pomocą fun. "podzial2",
% która dzieli dane na zbocza w/g okresu zmian wekt., zwraca cele z podz. danymi i li. zb.
%----- ZAPIS pośrednich wyników do pliku -----
fprintf('Podaj nazwę pliku w którym będą zapisane dane \n');
nazwa_pli_z=input(' ("1" - taki sam jak CSV, "0" - pomiń): ','s');
switch nazwa_pli_z
    case ('0')
        gdzie=[]; % NIE zapisano
        return
    case ('1')
        nazwa_pli_z=[DZ.plik(1:size(DZ.plik,2)-4),'_DZ'];
        % nazwa z końcówką "DZ" oznacza dane podzielone
end
kat_wyn_obl=[input('oraz nazwę podkatalogu z wynikami obliczeń: ','s'),''];
gdzie=[kat_wyn_obl,nazwa_pli_z]; % miejsce gdzie zapisano plik ze str. DZ
save(gdzie,'DZ');
%*****

%*****
% prad_sr2.m - oblicza średni prąd z uwzględnieniem prądu statycznego (dla "gece.m")
%           i poj. ekw. oraz określa SPOSOBY sterowania bramką (układem),
%           zwraca: obliczoną poj. ekw (CEK)
% ZMIENNNE
% DZ - struktura z danymi z symulacji po podziale na okresy
% PRS - struk. ze średnim prądem [uA] i numerami dzies. wektorów ster. w poszcz. okresach
% CEK - struktura podobna do PRS, ale z poj. ekw. obliczoną z PRS (pojemność w [fF])
% li_wej - to liczba wejść do bramki/układu
%-----
function [CEK,li_wej]=prad_sr2(co_z_wyn,DZ)
%----- OBLICZENIA - ŚREDNI PRĄD na zbocze -----
Vzas=input('Podaj napięcie zasilania w [V]: ');
PRS.jednostka='uA'; % prądy obliczono w [uA]
fprintf('Przetwarzany plik: "%s" ',DZ.plik)
fprintf('zawiera kolumny z danymi o następujących nazwach:\n')
wydruk(DZ.nag) % sformatowany wydruk nagłówka tj. nazw kolumn
fprintf('Podaj numery kolumn z prądem do obliczeń oddzielone przecinkami:\n')
kol_pr_s=input(' (k1, .. ,kn): ','s'); % nr. kol. jako string znakowy
[kol_pr,li_k_pr]=przer_str(kol_pr_s); % zwraca numery kolumn (text) oraz ich liczbę
for k=1:li_k_pr % po liczbie kolumn danych z prądem
    nr_k_pr=str2num(kol_pr{k});
    pole=strrep(DZ.nag{nr_k_pr}, '(' , '_'); % zamiana nawiasów na kreski "_"
    pole=pole(1:size(pole,2)-1); % obcięcie drugiej kreski
    eval(['PRS(1,1).',pole,'(1,1)=0;']) % żeby zdefiniować strukturę
    eval(['CEK(1,1).',pole,'(1,1)=0;']) % żeby zdefiniować strukturę
%----- obliczanie PRĄDU ŚREDNIEGO -----
for n=1:DZ.li_ok % po liczbie okresów (zboczy)
    DZ.dane_zb{n,nr_k_pr}=DZ.dane_zb{n,nr_k_pr}.*1e6; % PRĄD w [uA]
    x2=1; % chwilowy indeks do zliczania okresów z zerowym prądem stat.
    prad_sred=trapz(DZ.dane_zb{n,1},DZ.dane_zb{n,nr_k_pr})./DZ.okres;
    if isempty(prad_sred) | prad_sred==0 % dla pustych macierzy lub zer
        prad_sred=0;
        stat_pr_sr=0;
    else
        % Wybór liczby punktów do obliczenia średniego prądu statycznego: weźmiemy te
        % punkty dla których obl. krok czasowy był max. więc zaczniemy od końca półokresu
        ldpr=size(DZ.dane_zb{n,nr_k_pr},1); % Liczba Danych z PRądem - jest to liczba
        % punktów, w których Spice wykonał obliczenia
        dt=round((DZ.dane_zb{n,1}(ldpr,1)-DZ.dane_zb{n,1}(ldpr-1,1))*1000);
        % ostatni krok czasowy, razy 1000 i zaokrąglamy do liczby całkowitej, bo są
        % różnice na dalekich miejscach
        chdt=dt; % chwilowa delta czasu, będzie porównywana z ostatnią "dt"
        lps=0; % liczba punktów do obl. prądu statycznego
        while chdt >= dt & lps<ldpr-2 % żeby nie wyjść poza macierz
            lps=lps+1;
            chdt=round((DZ.dane_zb{n,1}(ldpr-lps,1)-DZ.dane_zb{n,1}(ldpr-lps-1,1))*1000);
        end
        lps=lps-2; % na wszelki wypadek nie bierzemy średniej z dwóch punktów, bo
        if lps <= 1 % prąd może nieco odstawać od statycznego
            stat_pr_sr=0;
            stat_zero{x2}=num2str(n);
            x2=x2+1; %indeks
        else

```

```
%----- Obliczanie STATYCZNEGO prądu śr. - średnia "lps" ostatnich wart. chwilowych w okr.
      stat_pr_sr=mean(DZ.dane_zb{n,nr_k_pr}((ldpr-lps):ldpr));
      end
      end
      prad_sred=prad_sred-stat_pr_sr;
      PRS=setfield(PRS,{1},pole,{n,1},prad_sred);
%----- Obliczanie POJEMNOŚCI EKWIWALENTNEJ (w [fF] bo okres w [ns] a prąd w [uA]) -----
      poj_ek=prad_sred.*DZ.okres./Vzas;
      CEK=setfield(CEK,{1},pole,{n,1},poj_ek);
      end
      if exist('stat_zero')
          fprintf('\n\tPrzyjęto śr_pr_stat=0 dla prądu: "%s" w okresach o nr.: \n\t',pole)
          wydruk(stat_zero);
          clear stat_zero
      end
      end
% ----- OKREŚLENIE SPOSOBÓW STEROWANIA w kolejnych zboczach (na podstawie napięć wej) ----
fprintf('\n\tOkreślanie SPOSOBÓW STEROWANIA bramką/układem\n')
fprintf('Przetwarzany plik: "%s" ',DZ.plik)
fprintf('zawiera kolumny z danymi o następujących nazwach:\n')
wydruk(DZ.nag) % sformatowany wydruk nagłówka tj. nazw kolumn
fprintf('\nPodaj numery kolumn z NAPIĘCIEM WEJ kolejno od najstarszego oddzielone')
fprintf(' przecinkami:\n')
kol_nap_s=input(' (k1, .. ,kn): ','s'); % nr. kol. jako string znakowy
[kol_nap,li_wej]=przer_str(kol_nap_s); % zwraca numery kolumn (text) i ich liczbę (li. wej)
% Opis: w każdym z wydzielonych okresów począwszy od połowy do jego końca będzie obliczana
% średnia wartość z nap., zaokrąglana do liczby całkowitej i podzielona przez nap. zasil.
% W ten sposób powstaną wektory binarne opisujące każdy okres. Następnie będą zamienione
% na liczby dzies. i umieszczone w str. PRS oraz zwrócone jako "wek_dz" i "li_wej"
for n=1:DZ.li_ok % po liczbie okresów
    ind=find(DZ.dane_zb{n,1}>=(n-0.5)*DZ.okres); % indeksy czasu od 1/2 do końca okresu
    for m=1:li_wej % po kolumnach z napięciami (rozmiar wektora, liczba wjść)
        nr_k_nap=str2num(kol_nap{m});
        wek_bin(n,m)=round(mean(DZ.dane_zb{n,nr_k_nap}(ind(1):ind(size(ind,1))))./Vzas);
    end
    PRS.wek_dz(n,1)=bin2dec(wek_bin(n,:)); % zamiana wart. binarnych na dziesiętne i
    CEK.wek_dz=PRS.wek_dz; % dopisanie do str. PRS i CEK
    clear ind
end
wek_dz=PRS.wek_dz;
% ----- ZAPIS WYNIKÓW (lub NIE) -----
if exist('co_z_wyn') & (isequal(co_z_wyn,'nosave') | isempty(co_z_wyn)); % NIE zapisujemy
else
    plik_prs=[co_z_wyn(1:size(co_z_wyn,2)-3),'_PRS']; % nazwa pliku z obl. PRADEM ŚREDNIM
    save(plik_prs,'PRS')
end
%*****
```

2. Obliczanie pojemności dla tradycyjnego modelu strat mocy

```
%*****
% cswa - pojemność ekw. dla tradycyjnego modelowania strat mocy (ze switching activity)
%-----
function Cekw=cswa(C) % C - struk. z poj. ekw. obliczoną dla nowego modelu (driving ways)
%----- sprawdzenie macierzy poj .ekw. -----
d=size(C.CLint);
if d(1) ~= d(2)
    fprintf(' Macierz CLint NIE JEST kwadratowa!\n')
    return
end
%----- Pojemność wyjściowa CL -----
bra=input('Podaj nazwę bramki (nand, nor): ','s');
d1=d(1)-1;
switch bra
    case {'nand','Nand','NAND'}
        cln=mean(C.CLint(d(1),1:d1)); % zbocze NAR na wy.
        clo=mean(C.CLint(1:d1,d(1))); % zbocze OPA na wy.
    case {'nor','Nor','NOR'}
        cln=mean(C.CLint(2:d(1),1)); % zbocze NAR na wy.
        clo=mean(C.CLint(1,2:d(1))); % zbocze OPA na wy.
    otherwise
        fprintf(' "%s" to NIE jest poprawna nazwa bramki\n',bra)
end
cl=cln+clo;
fprintf('Pojemność wyjściowa: CL=%.4f fF\n\n',cl);
Cekw={'CLint',cl}; % cela z obliczonymi poj. ekw., pierwsza CL a potem wejściowe
```

```

dalej=input('Czy obliczyć pojemności wejściowe (t,n): ','s');
if (dalej~='t') | (dalej~='tak')
    fprintf(' KONIEC\n')
    return
end
fprintf('\n')
%----- Pojemność wejściowa -----
inn='A'; % nazwa wejścia, dla którego będziemy obl. poj. we., zaczynamy od "A"
nw=1; % numer wejścia, dla "A" jest to "1"
Cin=C.CinA; % bo zaczynamy od A, Cin - tymczasowa macierz z wej. poj. ekw.
lw=log2(d(1)); % liczba wejść do bramki
x1=1;
while nw <= lw
    ind=1;
    for x=1:d(1) % po wierszach w macierzy poj.ekw.
        xb=dec2binm(lw,x-1); % x binarnie i od zera
        for y=1:d(1) % po kolumnach
            yb=dec2binm(lw,y-1); % y binarnie i od zera.
            % poszukiwanie zboczy narastających, x - wektor aktualny, y - wektor następny
            rb=yb-xb; % "-1" - zbocze opadające, "1" zbocze narastające
            rb=rb./2; % podział przez 2 i następnie zaokrąglamy do góry, żeby pozbyć
            rb=ceil(rb); % się ujemnych wartości, "1" oznacza zbocze NAR. "0" pozostałe
            if rb(nw)==1 % gdy dla we o num. nw zbocze jest NAR., to sprawdzamy czy wy się
            % zmienia (OPA) i Clint dla odwrotnych spos. ster. są równe co do modułu
                switch bra
                    case {'nand','Nand','NAND'}
                        if y==d(1) % ostatnia kolumna w tab. zm. wek. - zbocze OPA w NAND
                            se=0;
                        else
                            se=1;
                        end
                    case {'nor','Nor','NOR'}
                        if x==1 % pierwszy wiersz w tab. zm. wek. - zbocze OPA w NOR
                            se=0;
                        else
                            se=1;
                        end
                end
            if se==1 & abs((C.CLint(x,y)+C.CLint(y,x))./C.CLint(x,y))>0.2 % gdy nie ma
            % zbocza OPA na wy i Clint NIE są prawie równe co do modułu, to powinno
            % być ZERO, ale może być nieco więcej, dlatego margines 20% wart. bezw.
            ct(ind)=Cin(x,y)+C.CLint(x,y)./sum(rb); % suma jedynek w "rb"
            else
                ct(ind)=Cin(x,y);
            end
            ind=ind+1;
        end
    end
    end
    Cins=mean(ct);
    x1=x1+1;
    Cekw{x1,1}=['Cin',inn]; % zapis do celi z wynikami, 1-kolumna to nazwa poj.
    Cekw{x1,2}=Cins; % 2-kolumna - wart. pojemności
    inn=input('Podaj nazwę wejścia (A,B,C...,k-koniec): ','s'); % nazwa we w/g moich zasad
    nw=string(inn)-64; % numer wejścia (kod ascii A=65, zakł. duże litery)
    if nw==43 | nw==11
        fprintf(' *** KONIEC ***\n');
        return
    elseif nw<1 | nw>lw
        fprintf(' ZŁA nazwa wejścia!\n');
        return
    end
    Cin=eval(['C.Cin',inn, ';']);
end
%*****

```

3. Obliczanie prawdopodobieństwa sterowania bramek (węzłów) w układzie (algorytm 4.2)

```

%*****
% nam_pdw.m - New Activity Measure - Probability of Driving Way: oblicza DW - inf. o prawd
% sposobów ster. bramek na podstawie informacji o strukturze układu "fun" dla
% danego rozkładu prwad. zmian wektorów wejściowych "roz"
%----- ZMIENNE -----
% fun - nazwa pliku z opisem badanego układu
% pin - Primary INputs - liczba wejść do układu
% roz - nazwa ROZkładu prawdopodobieństwa zmian wektorów we.

```

```

% nn - Number of Nodes - liczba węzłów w układzie odczytane z celi 'stan0'
% wP - wektor Początkowy
% wN - wektor Następny
% BwP - Binarny wektor Początkowy o długości pin
% BwN - Binarny wektor Następny o długości pin
% DW - Driving Ways - cell'a (nn,3): nazwy węzłów i bramek, macierze z prawdopod.
% sposobów ster. węzłami i kolejnością połączeń wejść bramek
% stanP - cela (nn,3) stan układu dla wektora początkowego wP
% stanN - cela (nn,3) stan układu dla wektora następnego wN
% nch - Number of CHanges wejściowych wektorów odczytany z tab. rozkładu zmian wekt. we
% nac - Number of All Changes of input vectors (sum(sum(trans)))
% nw - wektor wskaźników wierszy do tab. prawdop. przejść dla bramek ster. węzły
% nk - wektor wskaźników kolumn do tab. prawdop. przejść dla bramek ster. węzły
% kwb - Kolejność Wejść Bramek ster. węzły odczytujemy z pliku opisującego układ, cell'a
% i,j - chwilowe indeksy
% ofn - Output File Name gdzie jest struktura DW i inne dane
%-----
function ofn=nam_pdw(fun,roz,katalog)
%----- ODCZYT DANYCH WEJŚCIOWYCH: układ, rozkład prawd. zmian wektorów we. -----
% jeśli NIE podano ich w wywołaniu funkcji
if (exist('katalog') == 1) && (isempty(katalog) == 0)
    katalog=[katalog,'\'];
else
    katalog=[];
end
if exist('fun')==0
    fun=input('Podaj nazwę pliku opisującego układ: ','s'); % nazwa m-pliku opisu układu
end
piny=feval(fun,'ile'); % odczyt liczby we. do układu z powyższego pliku opisu układu
pin=piny(1,1); % liczba wejść
if exist('roz') == 0
    roz=input('Podaj nazwę rozkładu: ','s');
end
roz1=[roz,num2str(pin)]; % uzupełnienie liczby we. do układu
trans=load(['rozklady\',roz1,'.mat']); % ładowanie pliku
trans=getfield(trans,roz1); % trans - macierz rozk. zmian wektorów wej.
%----- DEKLARACJE i OBLICZENIA POCZĄTKOWE -----
x0=dec2bin(pin,[0]); % generowanie zerowego wektora binarnego o długości "pin"
stan0=feval(fun,x0); % oblicz. stanu bramek w układzie dla 0 na we, żeby poznać budowę
% układu i przygotować odpowiednią cell'ę na wyniki obl. prawdop.
nn=size(stan0,1); % ile jest wierszy w celi stan0 tyle jest WĘZŁÓW w układzie
DW=cell(nn,3); % pusta cela
for i=1:nn % przepisywanie nazw ze stan0, tworzenie macierzy dla liczb sposobów ster.
    rozm=pow2(str2num(stan0{i,2}(5))); % rozmiar tabeli opisującej przejścia
    DW(i,:)={stan0{i,1},stan0{i,2},zeros(rozm)}; % cela z nazwami i zerowymi tabelami
end
koniec=pow2(pin); % wielkość tabeli przejść wektorów wejściowych (ogran. pętli obl.)
wP=0; wN=0; % wartości startowe wektorów wej., będą one wskaźnikami w pętlach
nac=sum(sum(trans)); % liczba wszystkich zmian wekt. wej. w macierzy "trans"
trans=trans./nac; % teraz macierz "trans" zawiera prawdopodobieństwo zmian wektorów we
%----- OBLICZENIA GŁÓNE -----
fprintf('\nObliczanie AKTYWNOŚCI układu (nam_pdw) dla rozkadu: %s\n',roz)
gwiazdka=1; % zmienna pomocnicza do wyświetlania postępu obliczeń
while wP<koniec
    BwP=dec2bin(pin,wP);
    stanP=feval(fun,BwP); % stan początkowy (w/g wierszy)
    wP=wP+1; wN=0; % następny wP i zerowy wN
    for i=1:nn % obliczanie wskaźników wierszy do tabeli sposobów ster. węzłów
        nw(i)=bin2dec(stanP{i,3})+1; % na podstawie stanu wejść bramek, czyli
    end % zamiana liczb. bin. na dec. + 1 (żeby nie od zera)
    while wN<koniec
        BwN=dec2bin(pin,wN);
        stanN=feval(fun,BwN); % stan następny (w/g kolumn)
        wN=wN+1;
        nch=trans(wP,wN); % odczyt częst. zmian o wsp. (wP,wN) z tab. rozk. prawd. przejść
        for i=1:nn % obliczanie wskaźników kolumn do tab. sposobów ster. węzłów,
            nk(i)=bin2dec(stanN{i,3})+1; % które trzeba zwiększyć
        % zwiększenie prawd. sposobu ster. węzłem, na który wskazuje (nw(i),nk(i)), o prwad.
            DW{i,3}(nw(i),nk(i))=DW{i,3}(nw(i),nk(i))+nch; % przejść wektorów wejściowych
        end
    end
    fprintf('*') % wyświetlanie postępu obliczeń - jedna * po jednym wierszu obliczeń
    if wP == gwiazdka*64 % max po 64 znaki w linii
        fprintf('\n')
        gwiazdka=gwiazdka+1;
    end
end
end

```

```
%----- Odczyt opisu KOLEJNOŚCI podłączenia we bramki sterujących węzłami -----
kwb=feval(fun,['kol']); % odczyt nazw węzłów które są wej. do bamek w układzie, teraz
for i=1:nn % funkcja zwraca nazwy węzłów zamiast stanów we. bamek
    DW{i,4}=kwb{i,3};
end
%----- ZAPIS wyników -----
ofn=[katalog,fun,'_',roz,'_pdw']; % nazwa pliku z wynikami wraz ze ścieżką
save (ofn,'DW','fun','roz');
%*****
```

4. Estymacja pojemności ekwiwalentnej układu z wykorzystaniem nowego modelu

Pojemność ekwiwalentna układów otrzymana z zastosowaniem nowego modelu strat mocy jest uzyskiwana podczas optymalizacji układu za pomocą programu „opt_poj.m”. Wyniki zamieszczone w rozdziale drugim zostały uzyskane właśnie w ten sposób.

5. Estymacja pojemności ekwiwalentnej z tradycyjnym modelem strat mocy

```
%*****
% est_poj.m - Estymacja mocy z zast. tradycyjnego modelu (wykorzyst. "switching activity")
% wykorzystujemy info o strukturze układu oraz o aktywności węzłów ("sw") z "DW" i
% poj. zastępcze obl. za pom. "cswa.m" dla wy i we bramki, z pliku "poj_sw_tech.mat"
%----- ZMIENNE -----
% naf - Name of Activity File, nazwa pliku z obliczoną aktywnością "nam_pdw.m"
% tech - Technology: "AMIS07", "AMS035" - parametry energetyczne bamek dla danej tech.
% CG - Capacitance of Gates, struktura z poj. ekw. dla bamek niezależną od draving way
% DW - Driving Ways - cell'a (nn,5) wczytana z pliku "naf" zawiera: nazwy węzłów i
% bamek, macierze z prawdopod. sposobów ster. węzłami i kolejnością połączeń we
% bamek oraz w kol. 5 klasyczne Switching Activity
% nn - Number of Nodes w układzie, liczba wierszy w celli "DW"
% gn - Name of Gate sterującej dany węzeł (char [1x5])
% gin - Gate Inputs Number wzięta z nazwy bramki
% NC - Nodes Capacitance - 1 i 2 kol. to kopia z "DW", suma obl. poj. wyj. bramki i
% wejściowych, [wektor z poj. wy, oraz weA, weB ...]
% eec - Estimated Equivalent Capacitance of a circuit - suma poj. wszystkich węzłów "NC"
% pcr - Power of the CiRcuit, trzeba to liczyć ze wzoru: "1/2 C U^2" bo fun "sw" liczy
% wszystkie przełączenia wy bramki, a ekw.poj. jest obliczona jako stosowna
% średnia z "gdw" - zależnej poj.ekw. Zaś moc z zasilania pobierana jest tylko
% dla zbocza narastającego na wyjściu.
%-----
function stat=est_poj(naf,tech,kat)
global DW nn
%----- ODCZYT DANYCH WEJŚCIOWYCH: aktywność i struktura ukł., poj. br. -----
if exist('naf')==0 % jeśli wywołanie bez par. to pytamy o nazwę pliku
    naf=input('Podaj nazwę pliku z obliczoną aktywnością układu: ','s'); %nazwa pliku
end
load (naf); % WCZYTANIE z pliku zmiennych o nazwach: DW - aktywności węzłów inne info.
% (patrz "nam_pdw.m"), fun - nazwa pliku z opisem badanego układu, roz - nazwa
% ROZkładu prawd. zmian wekt. we, sw - Switching activity całego układu
if ~exist('tech') % jeśli wywołanie bez par.
    tech=input('Podaj technologię: "AMIS07", "AMS035": ','s');
end
CG=load(['poj_sw_',tech]); %ładowanie z pliku pojemności ekw. dla bamek
%----- OBLICZENIA -----
nn=size(DW,1); % liczba WĘZŁÓW w układzie
%----- PĘTLA GŁÓNA - obliczenia poj. ekw. w każdym węźle w UKŁADZIE -----
for j=1:nn % po wszystkich węzłach w układzie
    gn=DW{j,2}; % nazwa bramki sterującej ten węzeł, zawsze o wym. [1x5]
    NC{j,1}=DW{j,1}; % przepisanie kolumny 1 z DW do NC (nazwa węzła)
    NC{j,2}=DW{j,2}; % przepisanie kolumny 2 z DW do NC (nazwa bramki ster. węzeł)
    NC{j,4}(1)=CG.(gn)(1).*DW{j,5}; % obl .poj. wyj. br.: CLint*sw(j)
    gin=str2num(gn(1,5)); % liczba wejść do bramki
    for k=1:gin % po liczbie wejść do bramki ster. węzeł "j"
        gnn=DW{j,4}{k}; % nazwa węzła podł. do wejścia nr "k" bramki
        swi=szu_we(gnn); % szukaj "sw" dla węzła o nazwie "gnn" w DW
        NC{j,4}(k+1)=CG.(gn)(k+1).*swi;
    end
    NC{j,3}=sum(NC{j,4}); %suma poj. wy i we skojarzonych z węzłem "j"
end
eec=sum(cell2mat(NC(:,3))); % pojemność w [fF]
```

```
%----- Obliczanie mocy strat dla f=100MHz -----
f=0.100; % w [GHz]
switch tech
    case 'AMIS07'
        Vdd=5;
    case 'AMS035'
        Vdd=3.3;
end
pcr=0.5*eec.*f.*Vdd^2; % w [uW], UWAGA na 1/2 w tym wzorze
%----- zapis wyników obliczeń -----
nazwa_pli=naf(1:findstr(naf,roz)+size(roz,2)); % "naf" bez "cec"
nazwa_pli=[kat,'\ ',nazwa_pli,'csw']; % "csw" oznacza, że poj. zast. obl. z "swit. act."
save(nazwa_pli,'NC','eec','pcr','fun','roz')
%*****
% SUBFUNKCJE
%-----
function snw=szu_we(naw)
% naw - nazwa węzła, snw - "switch. act." dla węzła o nazwie "naw", którą zwraca funkcja
global DW nn
k=0; % indeks pomocniczy będzie określał liczbę znalezionych linii z nazwą "naw"
for m=1:nn
    if isequal(DW{m,1},naw)
        k=k+1;
        nw(k)=m;
    end
end
if k>1 % gdy więcej niż 1 jest szukanych węzłów to JAKIŚ BŁĄD
    fprintf('Znaleziono więcej niż jeden węzeł o nazwie %s w celi "DW"\n',naw);
    snw=DW{nw(1),5}; % "sw" węzła o nazwie "naw", bierzemy pierwszy znaleziony węzeł
elseif k==0 % szukany węzeł jest węzłem wejściowym do układu i można by go pominąć
    snw=0;
elseif k==1 % znaleziono dokładnie jeden węzeł (to dobrze)
    snw=DW{nw(1),5}; % "sw" węzła o nazwie "naw"
end
```

6. Optymalizacja układu cyfrowego przez dobór najlepszej kolejności wejść do bramek

```
%*****
%opt_poj.m - Optymalizacja kolejności połączeń do wejść bramek
%----- ZMIENNE -----
% naf - Name of Activity File, nazwa pliku z obliczoną aktywnością "nam_pdw.m"
% tech - Technology: "AMIS07", "AMS035" - parametry energetyczne bramek w tech.
% EC - Equivalent Capacitory, struktura z poj. ekw. dla bramek
% TNA - Table of New Addresses, struktura z tab. adresów dla różnych podł. we. bramek
% nn - Number of Nodes w układzie, liczba wierszy w celli "DW"
% gn - Name of Gate sterującej dany węzeł (char [1x5])
% gin - Gate Inputs Number wzięta z nazwy bramki
% pdw - Probability of Driving Way for the gate (macierz)
% pojew - poj. ekw. węzła Clall, czyli suma CLint i Cinx wszystkich wejść
% nec - Node Equivalent Cap.
% DW0 - "DW" po Optymalizacji
% cec - Circuit Equivalent Capacitance wart. min i max jako sumy odpo. wart. z "nec"
%      oraz poj. ekw. dla połączeń wg. netlisty i poj. WE UKŁADU
% inn - circuit Input Names
% CIN - Circuit Input Nodes, cła z nazwami węzłów, których br. sterujące są podł. do we
%      UKŁ. CIN{j x 4}={'node name' [node number in DW] 'gate name' 'gate input name'}
% j - liczba wejść wszystkich bramek podłączonych do wejść UKŁ.
% inc - Input Node equivalent Capacitance
%-----
function [cec,nec,cin]=opt_poj(naf,tech)
%----- ODCZYT DANYCH WEJŚCIOWYCH: układ, rozkład prawd. zmian wekt. we., poj., adresy -----
if exist('naf')==0 % jeśli wywołanie bez par. to pytamy o nazwę pliku
    naf=input('Podaj nazwę pliku z obliczoną aktywnością układu: ','s'); %nazwa pliku
end
load (naf); % WCZYTANIE z pliku zmiennych o nazwach: DW - aktywności węzłów inne info.
% (patrz "nam_pdw.m"), fun - nazwa pliku z opisem badanego układu, roz - nazwa
% ROZKŁADU prawd. zmian wekt. we, sw - Switching activity całego układu
if ~exist('tech') % jeśli wywołanie bez par.
    tech=[];
end
EC=load(['poj_ekw_br_',tech]); %ładowanie z pliku pojemności ekw. dla bramek
TNA=load('tab_new_addr'); %ład. z pliku tab. z adresami dla różnych komb. podł. we bramek
%----- OBLICZENIA -----
nn=size(DW,1); % liczba WĘZŁÓW w układzie
%----- PĘTLA GŁÓNA - obliczenia poj. ekw. UKŁADU -----
for j=1:nn % po wszystkich węzłach w układzie
```

```

gn=DW{j,2}; % nazwa bramki sterującej ten węzeł, zawsze o wym. [1x5]
gin=str2num(gn(1,5)); % liczba wejść do bramki
pdw=DW{j,3}; % macierz z prawd. spos. ster. dla danego węzła
eval(['pojew=EC.',gn,'.CLall;']) %wyciągnięcie ze str. EC poj. ekw. całkowitej dla br.
for k=1:prod(1:gin) % po wszystkich sposobach podłączeń we. (kombinacje: gin! (silnia)
n_pdw=zeros(1,1); % nowa pdw z pozamienianymi wart. prawd. dla nowej kol. we do br.
for iw=1:pow2(gin) % po wierszach, wszystkie możliwe wektory we bramki
for ik=1:pow2(gin) % po kolumnach, wszystkie możliwe wektory we bramki
n_pdw(iw,ik)=pdw(getfield(TNA,['gate',gn(5)],{k,iw}),... %nowe miejsca
getfield(TNA,['gate',gn(5)],{k,ik})); % "starych" wartości prawd.
end
end
nec{j,1}=DW{j,1}; % 1-sza kol. celi zawiera nazwy węzłów
nec{j,2}(k)=sum(sum(n_pdw.*pojew)); % 2-ga kol. zawiera wektor z poj. ekw. dla
% kolejnych "k" sposobów podłączeń wejść do br. sterującej dany węzeł
NEC{j,k}=n_pdw.*pojew; % CELA : wiersz - nr węzła, kol. - nr spos. podł. we.
end
nec_mimao(j,1:2)=[min(nec{j,2}),max(nec{j,2})]; % wartości min i max poj. ekw. dla
nec_mimao(j,4:5)=[min(find(nec{j,2}==nec_mimao(j,1))),min(find(nec{j,2}==nec_mimao(j,2)))];
nec_mimao(j,3)=nec{j,2}(1); % wartości poj. ekw. dla oryg. sposobu połączeń (NETLISTA)
end
cec=sum(nec_mimao,1);
cec=[cec(1:3),0]; % trzy wartości: ekw_poj_max, ekw_poj_min, ekw_poj_org oraz Cin=0
%----- Ekwiwalentna pojemność WEJŚCIOWA układu -----
% 1. Wybór tych bramek, których wejścia są podłączone do węzłów we. UKŁADU - szukanie nazw
% węzłów we. ukł z "inn" wśród wszystkich węzłów w celi DW
inn=feval(fun,'wej'); %odczyt nazwy węzłów we. do UKŁADU z pliku "fun" opisującego układ
j0=1; % indeks tymczasowy, liczba we wszystkich bramek, które są podł. do we UKŁ.
CIN=cell(1,4); %pusta cela opisująca węzły wejściowe UKŁADU
for j1=1:size(DW,1) % po wierszach celi DW, czyli po wszystkich węzłach w układzie
for j2=1:str2num(DW{j1,2}(5)) % po wej. do bramki, liczba we. do br. z nazwy bramki
if any(strcmp(DW{j1,4}{j2},inn)) % czy węzeł we br. jest na liście węzłów we UKŁ.
CIN{j0,:}={DW{j1,4}{j2},j1,DW{j1,2},['Cin',char(64+j2)]};
j0=j0+1;
end
end
end
% 2. Obliczanie ekw. pojemności wejściowej UKŁADU
for j=1:size(CIN,1) % po wierszach z CIN,po liczbie wejść bramek podłączonych do wej. UKŁ.
poj_tmp=eval(['getfield(EC.',CIN{j,3},',''',CIN{j,4},''');']);%ekw.poj.dla odp. we br.
cin_tmp=sum(sum(poj_tmp.*DW{CIN{j,2},3}));
inc(j,:)={CIN{j,1},cin_tmp};
cec(1,4)=cec(1,4)+cin_tmp;
end
%*****

```

7. Obliczanie prawdopodobieństwa sposobów sterowania bramek (węzłów) w układzie z wykorzystaniem tabel czterowartościowych według algorytmu 4.3

```

%*****
% pdw.a.m - Probability of Driving Way - algorytm ALTERNATYWNY (z wyk. tabel czterowart.)
% oblicza DW - inf. o prawd. sposobów ster. bramek na podstawie informacji o strukturze
% układu "fun" dla danego rozkładu prwad. zmian wekt. we "roz", oblicza też Switch. Act.
% ZMIENNE
% fun - nazwa pliku z opisem badanego układu
% pin - Primary INputs - liczba wejść do układu
% roz - nazwa ROZkładu prawdopodobieństwa zmian wektorów we.
% NL - NetLista układu odczytana z opisu funkcji, cela (nn x 3)
% nn - Number of Nodes - liczba węzłów w układzie odczytane z NL
% wej - nazwy węzłów WEJściowych do układu
% wek - liczba wszystkich WEKtorów dla n-wejść do układu
% T4 - struktura z tabelami czterowart. dla wszystkich węzłów łącznie z wej. do układu,
% tabela T4 ma wym. (wek x wek x 2), (:,:,1) - stan węzła przed zmianą,
% a (:,:,2) -stan węzła po zmianie, lub ta para bitów jako wielowartościowa
% (dokładnie to 4 wartości: 00 - 0, 11 - 1, 01 - zb. nar., 10 - zb. opa.)
% nac - Number of All Changes of input vectors (sum(sum(trans))
function [TA]=nam_pdw(fun,roz,katalog)
%-----
%----- ODCZYT DANYCH WEJŚCIOWYCH: układ, rozkład prawd. zmian wektorów we. -----
% jeśli NIE podano ich w wywołaniu funkcji
if (exist('katalog')==1) && (isempty(katalog)==0)
katalog=[katalog,'\'];
else
katalog=[];
end
end

```

```

if exist('fun')==0
    fun=input('Podaj nazwę pliku opisującego układ: ','s'); %nazwa m-pliku opisującego ukł.
end
piny=feval(fun,'ile'); % odczyt liczby we. do układu z powyższego pliku opisu układu
pin=piny(1,1); % liczba wejść
wek=pow2(pin); % liczba wszystkich wektorów we. dla N-we. układu
% ładowanie rozkładu zmian wektorów wejściowych
if exist('roz')==0
    roz=input('Podaj nazwę rozkładu: ','s');
end
roz1=[roz,num2str(pin)]; % uzupełnienie liczby we. do układu
trans=load(['rozklady\'',roz1,'.mat']); % ładowanie pliku
trans=getfield(trans,roz1); % trans - macierz rozk. zmian wektorów wej.
nac=sum(sum(trans)); %liczba wszystkich zmian wekt. wej. w macierzy "trans"
trans=trans./nac; % teraz macierz "trans" zawiera prawdopodobieństwo zmian wekt. we
wej=feval(fun,'wej'); % odczyt nazw wejść do układu z pliku opisu układu
if size(wej,2)==pin % sprawdzenie liczby wejść
else
    fprintf('Błąd w pliku opisu funkcji: %s, zła liczba wejść\n',fun);
end
NL=feval(fun,'kol'); % odczyt struktury (netlisty) układu z pliku opisu układu
nn=size(NL,1); % liczba węzłów (bez wejść, ale te już znamy z "pin" oraz "wej")
%----- DEKLARACJE i OBLICZENIA POCZĄTKOWE -----
% tworzenie tabel T4 dla wejść układu
wd=(0:wek-1); % wszystkie możliwe wektory dla "pin"-wejść do układu (wart. dziesiętne)
wb=logical(rem(floor(wd*pow2(1-pin:0),2)); % wekt. binarnie - macierz liczbowa (logiczna)
for zm=1:pin % po wszystkich zmiennych wejściowych, ich nazwy są w "wej"
    for xw=1:wek % po wierszach właśnie tworzonej tabeli T4
        for xk=1:wek % po kolumnach tabel T4
            T4.(wej{zm})(xw,xk,:)=wb(xw,zm),wb(xk,zm)];
        end
    end
end
%----- PRZETŁA GŁÓNA (czyli tworzenie tabel T4 i tabel "pdw" dla węzłów w układzie) -----
for xn=1:nn % po węzłach w układzie
    lwb=str2num(NL{xn,2}(5)); % liczba wejść do bramki z jej nazwy
%----- okreslanie stanu węzła w 4-wartościowej logice, czyli tworzenie T4 dla węzła "xn"--
    T4.(NL{xn,1})(:,:1)=T4.(NL{xn,3}{1})(:,:1); %1-szy bit w T4 dla węzła na weA br.
    T4.(NL{xn,1})(:,:2)=T4.(NL{xn,3}{1})(:,:2); %2-szy bit w T4 dla węzła na weA br.
    inda=T4.(NL{xn,3}{1})(:,:1).*pow2(lwb-1); %ind. wekt. aktualnego do obl. tab. "pdw"
    indn=T4.(NL{xn,3}{1})(:,:2).*pow2(lwb-1); %ind. wekt. następnego do obl. tab. "pdw"
    for xb=2:lwb
        switch NL{xn,2}(1:3) % typ bramki
            case 'nan'
                T4.(NL{xn,1})(:,:1)=T4.(NL{xn,1})(:,:1) & T4.(NL{xn,3}{xb})(:,:1);
                % 1-bit w T4 dla kolejnych wejść do bramki
                T4.(NL{xn,1})(:,:2)=T4.(NL{xn,1})(:,:2) & T4.(NL{xn,3}{xb})(:,:2);
                % 2-bit w T4 dla kolejnych wejść do bramki
            case 'nor'
                T4.(NL{xn,1})(:,:1)=T4.(NL{xn,1})(:,:1) | T4.(NL{xn,3}{xb})(:,:1);
                T4.(NL{xn,1})(:,:2)=T4.(NL{xn,1})(:,:2) | T4.(NL{xn,3}{xb})(:,:2);
            end
            inda=inda+T4.(NL{xn,3}{xb})(:,:1).*pow2(lwb-xb); % obl. wart. dziesiętnej ind. do
            indn=indn+T4.(NL{xn,3}{xb})(:,:2).*pow2(lwb-xb); % tabel akt. bramki "pdw"
        end
        T4.(NL{xn,1})=~(T4.(NL{xn,1})); % bo NANDY i NORy, to jest już stan węzła "xn"
%----- obliczanie tabeli aktywności bramki ("pdw") sterującej węzeł "xn" -----
        inda=inda+1; % wymóg Matlab
        indn=indn+1;
        taw=zeros(pow2(lwb)); % inicjalizacja tab. akt. bramki
        for xw=1:wek % po wierszach tab. sterowania układu (czyli wszystkie możliwe wekt. we)
            for xk=1:wek % po kolumnach
                taw(inda(xw,xk),indn(xw,xk))=taw(inda(xw,xk),indn(xw,xk))+trans(xw,xk);
            end
        end
        TA{xn,1}=taw;
    end
%----- obliczenie pamięci zajętej przez wszystkie zmienne -----
    zm=whos;
    pam=0; % użyta pamięć na zmienne
    for k=1:size(zm,1) % po liczbie zmiennych
        pam=pam+zm(k,1).bytes;
    end
    fprintf('%.3f kB\n',pam/1024)
%*****

```

8. Obliczanie prawdopodobieństwa sposobów sterowania dla bramek pierwszego poziomu w układzie (algorytm 4.4)

```

%*****
% pdwk.m - oblicza prawd. sposobów sterowania dla kostek wej. (bramki pierwszego poziomu)
%          oblicza też tabele "T4" dla wyjść bramek realizujących kostki
% ZMIENNE
% TU - Tabela aktywności (pdw) dla Układu, z którego jest kostka
% K - Kostki, macierz znakowa kostek złożona z: {0, 1, -}, każdy wiersz to kostka
% lk - Liczba Kostek w tabeli "K"
% N - liczba we do układu (li. zmiennych funkcji) jest to li. WSZYSTKICH elementów kostki
% k - wymiar kostki - liczba "-" w kostce
% n - liczba wej. do bramki (liczba pozycji niebędących kreskami w kostce K)
% TB - Tabele aktywności ("pdw") dla bramek realizujących kostki z "K" i li. we. bramek
%       realizujących kostki ("TB" to cela o wym.: li.kostek x 2), w 3-kolumnie tabele T4
%       czyli czterowartościowe tabele stanów wy. bramek realizujących kostki
%*****
function TB=pdwk(K,TU)
%-----
%----- sprawdzenia wstępne -----
[lk,N]=size(K); % liczba i rozmiar kostek, N jest też liczbą wejść do układu
wek=pow2(N); % liczba wszystkich wektorów we. dla N-we. układu
[xtu,ytu]=size(TU); % sprawdzenie rozmiarów tabeli TU
if xtu==wek && ytu==wek
    % OK! - wymiary dobre
else
    fprintf('Tabela aktywności układu ma złe wymiary!\n');
    fprintf('Dla kostki: (%s) powinna być kwadratowa o wymiarach: %dx%d\n',K(1,:),wek,wek);
    TB=[];
    return
end
tu=TU./sum(sum(TU)); % prawdopodobieństwo zmian (w TU jest liczba zmian poszcz. wektorów)
%----- TWORZENIE tabel T4 dla wejść układu -----
wd=(0:wek-1)'; % wszystkie możliwe wektory dla "N"-wejść do układu (wart. dziesiętne)
wb=logical(rem(floor(wd*pow2(1-N:0)),2)); % wektory binarnie - macierz liczbowa (logiczna)
for zm=1:N % po wszystkich zmiennych wejściowych,
    for xw=1:wek % po wierszach właśnie tworzonej tabeli T4
        for xk=1:wek % po kolumnach tabel T4
            T4w.(['x',num2str(zm)])(xw,xk,:)=wb(xw,zm),wb(xk,zm)]; % "T4w" - dla wejść
        end
    end
end
%----- przygotowanie indeksów dla tabel TU i TB -----
for x=1:lk % po liczbie kostek w "K"
    ik=find(K(x,:)=='-'); % indeksy kresek "-" w kostce x-tek kostce z "K"
    k=size(ik,2);
    n=N-k; % liczba wejść do bramki
    TB{x,1}=zeros(pow2(n)); % inic. tabeli aktywności dla bramki realizującej x-tą kostkę
    % wybór i ewentualna negacja odpowiednich kolumn z "wi" według x-tej kostki z "K"
    TB{x,2}=n; % wymiar kostki
    x1=1; % indeks tymczasowy
    for x2=1:N
        switch K(x,x2)
            case '0'
                wik(:,x1)=~wb(:,x2); % negacja
                x1=x1+1;
            case '1'
                wik(:,x1)=wb(:,x2);
                x1=x1+1; % bez negacji
            case '-'
                % pomijamy
        end
    end
    wikd{x,1}=bin2dec(char(wik+'0'))+1; %dziesiętne wart. ind. dla TB, "+1" wymóg Matlaba
    clear wik ik n
end
%----- PĘTLA GŁÓWNA -----
for x=1:lk % po liczbie kostek w "K"
    stw=logical(ones(wek,wek,2)); %wartości pocz. do obl. stanów wy bramki dla x-tej kost.
    for xw=1:wek % po wierszach tabeli TU
        for xk=1:wek % po kolumnach tabeli TU
%----- obliczanie (xw,xk) elementu tabeli TB dla bramki realizującej kostkę "x" -----
            TB{x,1}(wikd{x,1}(xw),wikd{x,1}(xk))=...
                TB{x,1}(wikd{x,1}(xw),wikd{x,1}(xk))+tu(xw,xk); % elementy "tu" dodajemy
            % do odpowiednich wskazywanych przez wikd elementów TB
        end
    end
end

```

```
%----- obliczanie stanu wy bramki dla kostki "x", czyli tabelę "T4", zakładamy, że bramki
%                                             realizujące kostki to NANDy
    for x1=1:N                               % po liczbie wejść czyli li. el. kostki
        switch K(x,x1)
            case '1' % gdy "1" w kostce na "x1"-pozycji
                stw=stw & T4w(['x',num2str(x1)]);
            case '0' % gdy "0" w kostce na "x1"-pozycji
                stw=stw & ~T4w(['x',num2str(x1)]);
            case '-' % gdy "-" w kostce na "x1"-pozycji
                % to pomijamy tą zmienną
        end
    end
    TB{x,3}=~stw;
end
%*****
```

Załącznik B

Wyniki oceny parametrów energetycznych – pojemność ekwiwalentna – podstawowych bramek CMOS, wykorzystywanych w testach.

Tabela 5.1. Pojemność ekwiwalentna inwertera

C_{inA} [fF]			C_{Lint} [fF]			C_{Lall} [fF]		
v_a	v_n		v_a	v_n		v_a	v_n	
0	0	4,860	0	0	0,054	0	0	4,914
1	0	0	1	5,107	0	1	5,107	0

v_a – aktualny wektor wejściowy do bramki,

v_n – następny wektor wejściowy do bramki

Tabela 5.2. Całkowita pojemność ekwiwalentna NAND 2-we

C _{Lall} [fF]					
V _a	V _n	00	01	10	11
00		0	0,866	1,389	5,733
01		2,387	0	3,971	4,744
10		2,140	3,695	0	4,192
11		7,229	4,618	4,747	0

Tabela 5.3. Całkowita pojemność ekwiwalentna NOR 2-we

C _{Lall} [fF]				
V _a ^{V_n}	00	01	10	11
00	0	7,315	7,744	9,734
01	6,682	0	5,672	1,682
10	9,230	6,712	0	2,101
11	10,858	4,101	2,512	0

Tabela 5.4. Całkowita pojemność ekwiwalentna NAND 3-we

		C _{Lall} [fF]							
V _a	V _n	000	001	010	011	100	101	110	111
000		0	1,503	1,357	1,918	1,369	2,356	2,806	9,061
001		1,715	0	3,426	1,056	3,084	1,309	4,857	7,953
010		1,482	3,111	0	1,953	2,962	4,889	1,446	7,499
011		3,200	1,877	2,060	0	4,680	3,359	3,857	6,851
100		1,729	2,712	3,360	4,365	0	1,588	1,616	7,572
101		3,442	1,728	5,399	3,652	1,716	0	3,666	6,385
110		3,296	5,673	1,872	3,996	1,597	3,964	0	5,935
111		7,748	6,317	6,173	4,675	5,718	4,305	3,934	0

Tabela 5.5. Całkowita pojemność ekwiwalentna NOR 3-we

		C _{Lall} [fF]							
v _a	v _n	000	001	010	011	100	101	110	111
000		0	11,593	16,198	14,565	19,950	15,452	18,997	16,681
001		12,993	0	12,374	1,664	15,431	2,361	14,910	3,099
010		18,366	11,196	0	2,217	11,084	10,525	1,526	3,739
011		20,804	8,026	2,693	0	13,567	8,116	7,865	1,524
100		23,860	16,453	10,984	12,302	0	2,729	2,124	3,506
101		20,145	7,261	16,491	9,860	2,196	0	4,381	1,288
110		26,178	18,811	8,008	10,220	3,113	6,208	0	2,217
111		30,245	15,708	15,636	8,001	9,701	3,600	2,693	0

Tabela 5.6. Całkowita pojemność ekwiwalentna NAND 4-we

		C_{Lall} [fF]															
v_n		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
v_a																	
0000	0	1,450	1,394	2,011	1,271	2,256	1,966	2,458	1,361	2,292	2,574	2,936	2,724	3,765	3,858	10,995	
0001	1,641	0	3,335	1,156	2,928	1,291	3,908	1,721	3,001	1,303	4,494	2,280	4,365	2,663	5,793	9,887	
0010	1,299	2,829	0	1,960	2,872	4,232	0,927	1,645	2,660	3,885	1,302	1,942	4,300	6,113	2,599	9,569	
0011	2,944	1,637	1,929	0	4,506	3,118	2,817	0,859	4,302	2,936	3,225	1,302	5,944	4,642	4,918	8,954	
0100	1,314	2,245	2,952	3,488	0	1,546	1,082	2,591	2,777	3,740	4,515	5,766	1,442	2,157	2,651	9,468	
0101	2,956	1,314	4,837	2,608	1,639	0	2,998	1,433	4,417	2,749	6,434	4,697	3,085	1,169	4,573	8,329	
0110	2,614	3,987	1,622	2,279	1,660	3,415	0	2,601	4,079	5,641	3,120	4,851	3,156	4,873	1,523	8,051	
0111	4,258	2,955	3,542	1,602	3,229	1,929	1,932	0	5,720	4,422	5,008	3,091	5,022	3,720	4,268	7,897	
1000	1,570	2,502	2,861	3,200	3,107	4,039	3,900	4,816	0	1,541	1,413	2,091	1,533	2,483	2,678	9,563	
1001	3,210	1,571	4,762	2,545	4,818	3,180	5,880	4,093	1,638	0	3,332	1,151	3,172	1,474	4,603	8,398	
1010	2,872	4,106	1,572	2,215	4,755	6,435	2,906	4,092	1,299	2,819	0	1,959	3,105	4,889	1,440	8,079	
1011	4,515	3,178	3,503	1,571	6,313	5,010	4,822	3,396	2,940	1,636	1,928	0	4,751	3,445	3,728	7,402	
1100	2,886	3,863	4,746	5,836	1,706	2,639	2,945	4,468	1,411	2,343	3,248	4,310	0	1,534	1,489	8,044	
1101	4,525	2,930	6,658	5,163	3,346	1,706	4,788	3,747	3,054	1,413	5,157	3,607	1,640	0	3,409	6,804	
1110	4,461	6,730	3,247	5,476	3,386	5,626	1,723	4,315	2,861	5,153	1,737	3,896	1,591	3,854	0	6,521	
1111	9,153	7,681	7,839	6,446	7,757	6,253	5,917	5,028	7,306	5,837	6,016	4,604	5,679	4,213	3,845	0	

Tabela 5.7. Całkowita pojemność ekwiwalentna NOR 4-we

		C_{Lall} [fF]															
v_n		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
v_a																	
0000	0	14,915	21,522	18,611	27,394	19,926	25,010	20,699	32,021	20,037	25,828	22,796	30,555	21,942	27,117	22,614	
0001	16,883	0	16,409	1,996	21,321	3,022	19,677	3,195	25,751	2,869	20,257	5,547	24,250	4,716	21,645	4,538	
0010	24,190	14,492	0	2,591	15,372	13,504	1,492	4,050	19,317	11,916	2,617	5,207	18,327	15,367	2,828	5,348	
0011	27,316	10,706	3,448	0	18,795	10,752	10,246	1,458	22,691	9,246	11,947	2,616	21,378	12,607	12,172	2,755	
0100	31,732	21,748	14,495	15,937	0	3,186	2,363	3,911	12,905	13,439	12,917	14,378	1,501	4,718	3,928	5,408	
0101	26,268	9,314	22,123	13,017	2,877	0	5,280	1,318	15,989	10,205	21,226	12,896	9,557	1,513	7,168	2,815	
0110	34,615	24,652	10,816	13,404	4,168	7,776	0	2,594	17,129	17,623	11,215	14,159	11,054	9,268	1,496	4,087	
0111	37,804	20,943	14,376	10,804	7,185	4,747	3,444	0	20,118	15,028	17,934	11,583	14,240	6,495	5,411	1,494	
1000	39,279	29,380	21,845	23,236	14,070	17,188	15,648	17,057	0	3,234	3,070	4,547	1,889	4,981	3,293	5,171	
1001	25,235	9,664	25,215	14,689	20,745	14,261	21,147	14,284	2,873	0	5,739	1,863	4,716	1,800	6,458	2,578	
1010	33,240	23,359	9,844	12,428	21,126	23,580	12,749	15,333	3,540	7,168	0	2,593	5,890	9,462	1,258	3,852	
1011	36,481	19,571	13,296	9,844	25,009	21,553	16,144	12,695	6,597	4,145	3,442	0	8,888	6,441	4,716	1,258	
1100	42,284	32,087	24,861	26,290	10,845	14,079	13,327	14,755	4,100	7,217	6,974	9,047	0	3,092	1,815	3,910	
1101	37,057	19,435	33,283	23,306	13,668	10,840	16,090	12,159	6,961	4,102	10,350	6,455	2,885	0	5,197	1,320	
1110	45,197	35,086	21,257	23,848	15,049	18,752	10,835	13,426	7,880	11,451	4,686	7,268	4,154	7,733	0	2,593	
1111	51,952	31,385	31,333	21,252	23,945	15,622	14,495	10,832	15,823	8,437	8,174	4,690	7,176	4,715	3,446	0	

Załącznik C

Skrypty *algebraic* i *rudded* programu SIS oraz biblioteka *nandnor4.gen* wykorzystywane do optymalizacji układów w rozdziale trzecim.

```
# script AGEBRAIC
sweep
eliminate 5
simplify -m nocomp -d
resub -a
gkx -abt 30
resub -a; sweep
gkx -bt 30
resub -a; sweep
gkx -abt 10
resub -a; sweep
gkx -bt 10
resub -a; sweep
gkx -ab
resub -a; sweep
gkx -b
resub -a; sweep
eliminate 0
decomp -g *

# błędy do pliku
set siserr errors.txt
# mapowanie z biblioteką
"nandnor3.gen
read_library nandnor3.gen
map
# zapis informacji o układzie do
pliku
set sisout sis_out.txt
print_stats
echo
print_level -1
echo
echo Levels of the circuits:
echo
print_level

# biblioteka nandnor4.gen

GATE not_1 1      0=!A;
GATE nand2 2      0=!(A*B);
GATE nand3 3      0=!(A*B*C);
GATE nand4 4      0=!(A*B*C*D);
GATE nor_2 2      0=!(A+B);
GATE nor_3 3      0=!(A+B+C);
GATE nor_4 4      0=!(A+B+C+D);
GATE zero 0       0=CONST0;
GATE one 0        0=CONST1;

PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
PIN * INV 1 999 1 1 1 1
```

Bibliografia

- [Abd04] A. Abdollahi, F. Fallah, M. Pedram, "Leakage Current Reduction in CMOS VLSI Circuits by Input Vector Control", IEEE Transactions on VLSI, vol. 12, no. 2, February 2004, pp. 140-154.
- [Alv98] A. Alvandpour, P. Larsson-Edefors, C. Svenson, "Separation and Extraction of Short-Circuit Power Consumption in Digital CMOS VLSI Circuits", Proceedings of International Symposium on Low Power Electronics and Design (ISLPED'98), Monterey, USA, 10-12 August 1998, pp. 245-249.
- [Asa95] K. Asada, J. Akita, "A Method for Reducing Power consumption of CMOS Logic Based on Signal Transition Probability", IEICE Transactions on Electronics, vol. E78-C, no. 4, April 1995, pp.436-440.
- [Bac05] W. W. Bachman, S. A. Huss, "Efficient Algorithms for Multilevel Power Estimation of VLSI Circuits", IEEE Transactions on VLSI, vol. 13, no. 2, February 2005, pp. 238-254.
- [Bah95] R. I. Bahar, F. Somenzi, "Boolean Techniques for Low Power Driven Re-Synthesis", Proceedings of International Conference on Computer Aided Design (ICCAD'95), San Jose, USA, 5-9 November 1995, pp. 428-432.
- [Bel95] A. Bellaouar, M. Elmasry, "Low Power Digital VLSI Design, Circuits and Systems", Kluwer, Norwell, 1995.
- [Ben00] L. Benini, G. De Micheli, "System-Level Power Optimization: Techniques and Tools", ACM Transactions on Design Automation of Electronics Systems, vol. 5, no. 2, April 2000, pp. 115-192.
- [Bis00] L. Bisdounis, O. Koufopavlou, "Short-Circuit Energy Dissipation Modeling for Submicrometer CMOS Gates", IEEE Transaction on Circuits and Systems-I: Fundamentals Theory and Applications, vol. 47, no. 9, September 2000, pp. 1350-1361.
- [Bis98] L. Bisdounis, S. Nikolaidis, O. Koufopavlou, "Propagation Delay and Short-Circuit Power Dissipation Modeling of the CMOS Inverter", IEEE Transaction on Circuits and Systems-I: Fundamentals Theory and Applications, vol. 45, no. 3, March 1998, pp. 259-270.
- [Bor95] M. Borah, M. J. Irwin, R. M. Owens, "Minimizing Power Consumption of Static CMOS Circuits by Transistor Sizing and Input Reordering", Proc. of the 8th International Conference on VLSI Design, New Delhi, India, 4-7 January 1995, pp. 294-298.
- [Bra84] R. Brayton, G. D. Hachtel, C. McMullen, A. Sangiovanni-Vincentelli, "Logic Minimization Algorithms for VLSI Synthesis", Kluwer Academic Publisher, Boston, 1984.
- [Brg85] F. Brglez, H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran", Proceedings of International Symposium on Circuits and Systems, May 1985, pp. 663-698.
- [Brz00a] I. Brzozowski, P. Bratek, P. Dziurdzia, A. Kos, "New Concept of Low Power Digital Circuits Design", Proceedings of the 7th MIXDES Conference, Gdynia, Poland, 15-17 June 2000, pp. 181-184

- [Brz00b] I. Brzozowski, A. Kos, "Synthesis Method for Power Supply Reduction in Digital Circuits", Proceedings of the International Conference on Signals and Electronic Systems (ICSES), Ustroń, Poland, 17-20 October 2000, pp. 355-360.
- [Brz00c] I. Brzozowski, A. Kos, "Energy Consumption Minimisation with New Synthesis Method", Proceedings of the 7th IEEE International Conf. on Electronics, Circuits and Systems (ICECS), Jounieh-Kaslik, Lebanon, 17-20 December 2000, pp. 605-608.
- [Brz01] I. Brzozowski, A. Kos, "Logic Synthesis Method for Power Dissipation Reduction in Combinational Digital Circuits", Bulletin of The Polish Academy of Sciences Technical Sciences, vol. 49, no. 4, 2001, pp. 581-594.
- [Brz02] I. Brzozowski, A. Kos, "New Idea of Objective Assessment of Energy Consumed by Real VLSI Gates", Proceedings of the 9th MIXDES Conference, Wrocław, Poland, 20-22 June 2002, pp. 579-584.
- [Brz03] I. Brzozowski, A. Kos, "Modelling of Dynamic Power Dissipation for Static CMOS Gates and Logic Networks", Proceedings of the 10th MIXDES Conference, Łódź, Poland, June 26-28 2003, pp. 336-341.
- [Brz04a] I. Brzozowski, J. Kołodziej, A. Kos, "Simulation Methods for Estimation of Dynamic Power Dissipation in Digital CMOS Circuits", Proceedings of the 11th MIXDES Conference, Szczecin, Poland, 24-26 June 2004, pp. 435-440.
- [Brz04b] I. Brzozowski, A. Kos, "Input Voltage Impact on Quasi-Short Power Dissipation of CMOS Gates", Proceedings of the International Conference on Signals and Electronic Systems (ICSES), Poznań, Poland, 13-15 September 2004, pp. 23-26.
- [Brz04c] I. Brzozowski, A. Kos, "Assessment and Modelling of Quasi-Short Power Dissipation in CMOS Gates", Proceedings of the XXVIII International Conference of International Microelectronics and Packaging Society (IMAPS) Poland Chapter, Wrocław, Poland, 26-29 September 2004, pp. 183-186.
- [Brz05] I. Brzozowski, A. Kos, "Power Dissipation Reduction During Synthesis of Two-Level Logic Based on Probability of Input Vectors Changes", Proc. of 15th Int. Workshop on Power and Timing Modelling, Optimization and Simulation (PATMOS 2005), Leuven, Belgium, 20-23 September, 2005, in Lecture Notes in Computer Science: Integrated Circuit and System Design, Springer Verlag, vol. 3728, pp. 456-465.
- [Brz99a] I. Brzozowski, A. Kos, "Minimisation of Power Consumption in Digital Integrated Circuits by Reduction of Switching Activity", Proceedings of the 25th EUROMICRO Conference, Milan, Italy, 8-10 September 1999, vol. 1, pp. 376-380.
- [Brz99b] I. Brzozowski, A. Kos, "Switching Activity Reduction with a New Logic Synthesis Method", XXII KKTOiUE, Stare Jabłonki, Poland, October 1999, vol. 1, pp. 143-148.
- [Bur93] R. Burch, F. N. Najm, P. Yang, T. N. Trick, "A Monte Carlo Approach for Power Estimation", IEEE Transaction on VLSI, vol. 1, no. 1, March 1993, pp. 63-71.
- [Buy05] K. M. Buyuksahin, F. N. Najm, "Early Power Estimation for VLSI Circuits", IEEE Transactions on CAD, vol. 24, no. 7, June 2005, pp. 1076-1088.
- [Cha70] K.K. Chakrabarti, A.K. Choudhury, M. S. Basu, "Complementary Function Approach to the Synthesis of Three-Level NAND Network", IEEE Transactions on Electronic Computers, vol. C-19, no. 6, June 1970, pp. 509-514.
- [Che00] W-K. Chen (ed.), "The VLSI Handbook", CRC Press LLC, Boca Raton, 2000.

- [Cir87] M. A. Cirit, "Estimating Dynamic Power Consumption of CMOS Circuits", Proceedings of the International Conference on Computer-Aided Design, 9-12 November 1987, pp. 534-537.
- [DeM94] G. De Mey, "Thermal Problems in Electronics", Proceedings of Advanced Training Course: "Mixed Design of VLSI Circuits", Dębe, Poland, 5-9 April 1994, pp. 25-35.
- [Dev90] S. Devadas, K. Keutzer J. White, "Estimation of power dissipation in CMOS combinational circuits", Proceedings of IEEE Custom Integrated Circuits Conference, 1990, pp. 19.7.1–19.7.6.
- [Dev92] S. Devadas, K. Keutzer, J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation", IEEE Transactions on CAD, vol. 11, no3, March 1992, pp. 373-383.
- [Din00] C-S. Ding, C-T. Hsieh, M. Pedram, "Improving the Efficiency of Monte Carlo Power Estimation", IEEE Transaction on VLSI, vol. 8, no. 5, October 2000, pp.584-593.
- [Din98] C-S. Ding, C-Y. Tsui, M. Pedram, "Gate-Level Power Estimation Using tagged Probabilistic Simulation", IEEE Transactions on CAD, vol. 17, no. 11, November 1998, pp. 1099-1107.
- [DMi94] G. De Micheli, "Synthesis and Optimization of Digital Circuits", McGraw-Hill, 1994.
wyd. pol.: „Synteza i optymalizacja układów cyfrowych”, tł. Z. Ażgin, WNT, 1998.
- [Eli65] D. T. Elis, "A Synthesis of Combinational Logic with NAND or NOR Elements", IEEE Transactions on Electronic Computers, vol. EC-14, no. 5, October 1965, pp. 701-705.
- [Evm02] N. Evmorfopoulos, G. Stamoulis, J. Avaritsiotis, "A Monte Carlo Approach for Maximum Power Estimation Based on Extreme Value Theory", Transaction on CAD, vol. 21, no. 4, April 2001, pp. 415-432.
- [Fis96] G. Fishman, "Monte Carlo: Concepts, Algorithms and Applications", Springer, New York, 1996.
- [Fle75] H. Fleisher, L. I. Maissel, "An introduction to array logic", IBM Journal of Research and Development, vol. 19, March 1975, pp. 98-109.
- [Fre01] A. T. Freitas, A. L. Oliveira, "Circuit Partitioning Techniques For Power Estimation Using The Full Set of Input Correlations", Proc. of the 8th IEEE Intern. Conference on Electronics, Circuits and Systems, Malta, September 2001, vol. 2, pp. 903-907.
- [Fre99] A. T. Freitas, A. L. Oliveira, J. C. Monteiro, H. C. Neto, "Exact Power Estimation Using Word-Level Transition Probabilities", Proceedings of the 9th International Workshop on Power and Timing Modelling, Optimization and Simulation (PATMOS99), Kos Island, Greece, October 1999, pp 355-364.
- [Gho92] A. Ghosh, S. Devadas, K. Keutzer, J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits", Proceedings of 29th ACM/IEEE Design Automation Conference, June 1992, pp. 253–259.
- [Gim67] J. F. Gimpel, "The Minimization of TANT Networks", IEEE Transactions on Electronic Computers, vol. EC-16, no. 1, February 1967, pp. 18-38.
- [Gup00a] S. Gupta, F. N. Najm, "Power Modeling for High-Level Power Estimation", IEEE Transactions on VLSI, vol. 8, no. 1, February 2000, pp.18-29.

- [Gup00b] S. Gupta, F. N. Najm, "Analytical Models for RTL Power Estimation of Combinational and Sequential Circuits", *IEEE Trans. on CAD*, vol. 19, no. 7, July 2000, pp. 808-814.
- [Has99] M. Hasimoto, H. Onodera, K. Tamaru, "A Power and Delay Optimization Method Using Input reordering in Cell-Based CMOS Circuits", *IEICE Transactions on Fundamentals*, vol. E82-A, no. 1, January 1999, pp. 159-166.
- [Hed87] N. Hedenstierna, K. O. Jeppson, "CMOS Circuit Speed and Buffer Optimization", *IEEE Transaction on CAD*, vol. CAD-6, no. 2, March 1987, pp. 270-281.
- [Hel63] L. Hellerman, "A Catalog of Three-Variable Or-Invert and And-Invert Logical Circuits", *IEEE Transactions on Electronic Computers*, vol. EC-12, June 1963, pp. 198-223.
- [Hil96] A. M. Hill, S-M. Kang, "Determining Accuracy Bounds for Simulation-Based Switching Activity Estimation", *IEEE Transaction on CAD*, vol. 15, no. 6, pp. 611-618
- [Hos96] R. Hossain, M. Zheng, A. Albicki, "Reducing Power Dissipation in CMOS by Signal Probability Based Transistor Reordering", *IEEE Transactions on CAD*, vol.15, no. 3, March 1996, pp. 361-368.
- [Hua96] S. Huang, K. Cheng, K. Chen, T. Lee, "A Novel Methodology for Transistor-Level Power Estimation", *Proceedings of Intern. Symposium on Low Power Electronics and Design (ISLPED'96)*, Monterey, USA, 12-14 August 1996, pp. 67-72.
- [Ima95] S. Iman, M. Pedram, "Two-Level Logic Minimization for Low Power", *Proceedings of International Conference on Computer Aided Design (ICCAD'95)*, San Jose, USA, 5-9 November 1995, pp. 433-438.
- [Ima98] S. Iman, M. Pedram, "Logic Synthesis for Low Power VLSI Design", Kluwer, Boston, 1998.
- [Kan86] S. M. Kang, "Accurate Simulation of Power Dissipation in VLSI Circuits", *IEEE Journal of Solid-State Circuits*, vol. SC-21, October 1986, pp. 889-891.
- [Lin05] Z. Lin, R. Srivaths, "RTL-Aware Cycle-Accurate Functional Power Estimation", *IEEE Transactions on CAD*, accepted for publication
- [Lis88] R. Lisanke, "Logic synthesis and optimization benchmarks", Technical report, Microelectronics Center of North Carolina, P.O. Box 12889, Research Triangle Park, NC 27709, 16 December 1988.
- [Liu04] X. Liu, M. C. Papaefthymiou, "Markov Chain Sequence Generator for Power Macromodeling", *IEEE Transaction on CAD*, vol. 23, no. 7, July 2004, pp. 1048-1062.
- [Łub98] T. Łuba, M. Nowicka, M. Perkowski, M. Rawski, „Nowoczesna synteza logiczna”, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1998.
- [Mac98] E. Macii, M. Pedram, F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization", *IEEE Transaction on CAD*, vol. 17, no. 11, Nov. 1998, pp. 1061-1079.
- [Mar00] R. Marculescu, C. Ababei, "Improving Simulation Efficiency for Circuit-Level Power Estimation", *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, Geneva, Switzerland, 28-31 May 2000, pp. I-471-I-474.
- [Mar94] R. Marculescu, D. Marculescu, M. Pedram, "Switching Activity Analysis Considering Spatiotemporal Correlations", *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, November 1994, pp. 294-299.

- [Mar95] R. Marculescu, D. Marculescu, M. Pedram, "Efficient Power Estimation for Highly Correlated Input Streams", Proceedings of the 32nd ACM/IEEE Design Automation Conference, San Francisco, USA, June 1995, pp. 628–634.
- [Mar96] D. Marculescu, R. Marculescu, M. Pedram, "Information Theoretic Measures for power Analysis", IEEE Transactions on CAD, vol. 15, no.6, June 1996, pp. 599-610.
- [Mar97] R. Marculescu, D. Marculescu, M. Pedram, "Hierarchical Sequence Compaction for Power Estimation", Proceedings of 34th Design Automation Conference (DAC'97), Anaheim, USA, 9-13 June 1997, pp. 570-575.
- [Mar98] R. Marculescu, D. Marculescu, M. Pedram, "Probabilistic Modeling of Dependencies During Switching Activity Analysis", IEEE Transactions on CAD, vol. 17, no. 2, February 1998, pp. 73-83.
- [McC56] E. J. McCluskey, "Minimization of Boolean Functions", The Bell System Technical Journal, vol. 35, no. 5, November 1956, pp. 1417-1444.
- [Mon97] J. Monteiro, S. Devadas, A. Ghosh, K. Keutzer, J. White, "Estimation of Average Switching Activity in Combinational Logic Circuits Using Symbolic Simulation", IEEE Transactions on CAD, vol. 16, no. 1, January 1997, pp.121-127.
- [Naj91] F. N. Najm, "Transition Density, A Stochastic Measure of Activity in Digital Circuits", Proceedings of 28th ACM/IEEE Design Automation Conference, June 1991, pp. 644-649.
- [Naj93] F. N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits", IEEE Transactions on CAD, vol. 12, no. 2, February 1993, pp. 310-323.
- [Naj94] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits", IEEE Transactions on VLSI, vol. 2, no. 4, December 1994, pp. 446-455.
- [Naj95] F. N. Najm, "Power Estimation Techniques for Integrated Circuits", Proceedings of International Conference on Computer Aided Design (ICCAD'95), San Jose, USA, 5-9 November 1995, pp. 492-499.
- [Nem96] M. Nemani, F. N. Najm, "Towards a High-Level Power Estimation Capability", IEEE Transaction on CAD, vol. 15, no. 6, June 1996, pp. 588-598.
- [Nos00] K. Nose, T. Sakurai, "Analysis and Future Trend of Short-Circuit Power", IEEE Transaction on CAD, vol. 19, no. 9, September 2000, pp. 1023-1030.
- [Par75] K. P. Parker, E. J. McCluskey, "Probabilistic Treatment of general Combinational Networks", IEEE Transactions on Computers, no. C-24, June 1975, pp. 668-670.
- [Ped96] M. Pedram, "Power Minimization in IC Design: Principles and Applications", ACM Transactions on Design Automation of Electronics Systems, vol. 1, no. 1, January 1996, pp. 3-56.
- [Pra96] S. C. Prasad, K. Roy, "Transistor Reordering for Power Minimization Under Delay Constraint", ACM Transactions on Design Automation of Electronic Systems, vol. 1, no. 2, April 1996, pp. 280-300.
- [Qui52] W. V. Quine, "The Problem of Simplifying Truth Functions", American Mathematical Monthly, vol. 59, no. 8, October 1952, pp. 521-531.
- [Qui55] W. V. Quine, "A Way to Simplify Truth Functions", American Mathematical Monthly, vol. 62, no. 9, November 1955, pp. 627-631.

- [Roy03] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits", *Proceedings of the IEEE*, vol. 91, no. 2, February 2003, pp. 305-327.
- [Roy93] K. Roy, S. C. Prasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations", *IEEE Transactions on VLSI*, vol. 1, no. 4, December 1993, pp. 503-513.
- [Rud89] R. Rudell, "Logic Synthesis for VLSI Design", PhD Thesis, University of California Berkeley, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, USA, April 1989. Technical Report No. UCB/ERL M89/49.
- [Sak04a] T. Sakurai, "Perspectives in Power-Aware and Large-Area Integrated Circuits for Ubiquitous Electronics", *International Symposium on Electronics for Future Generations*, Tokyo, March 10, 2004, pp.71-74.
- [Sak04b] T. Sakurai, "Perspectives of Low-Power VLSI's", *IEICE Transactions on Electronics*, vol. E87-C, no. 4, April 2004, pp.429-436.
- [Sak83] T. Sakurai, K. Tamaru, "Simple Formulas for Two- and Three-Dimensional Capacitances", *IEEE Transactions on Electron Devices*, vol. ED-30, no. 2, February 1983, pp. 183-185.
- [Sen92] E. M. Sentovich, et al., "SIS: A System for Sequential Circuit Synthesis", *Electronics Research Laboratory, University of California, Berkeley, USA*, Memorandum No. UCB/ERL M92/41, May 1992.
- [She87] B. J. Sheu, D. L. Scharfetter, P-K. Ko, M-C. Jeng, "BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors", *IEEE Journal of Solid-State Circuits*, vol. SC-22, no. 4, August 1987, pp. 558-566.
- [She92] A. Shen, A. Ghosh, A. Devadas, K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks", *Proceedings of International Conference on Computer-Aided Design (ICCAD'92)*, Santa Clara, USA, 8-12 November 1992, pp.402-407.
- [She95] W-Z. Shen, J-Y. Lin, F-W. Wang, "Transistor Reordering Rules for Power Reduction in CMOS Gates", *Proceedings of Asian and South Pacific Design Automation Conference (ASP-DAC '95)*, Makuhari, Japan, 29 August-1 September 1995, pp. 1-6.
- [Sul04] A. K. Sultania, D. Sylvester, S. S. Sapatnekar, "Transistor and Pin Reordreing for Gate Oxide Leakage Reduction in Dual T_{ox} Circuits", *Proceedings of IEEE International Conference on Computers Design: VLSI in Computers and Processors (ICCD 2004)*, San Jose, USA, 11-13 October 2004, pp. 228-233.
- [The01] G. Theodoridis, S. Theoharis, D. Soudris, C. Goutis, "A Probabilistic Power Estimation Method for Combinational Circuits under Real Gate Delay Model", *VLSI Design, Journal of Custom-Chip Design, Simulation, and Testing*, vol. 12, no. 1, 2001, pp. 69-79.
- [The98] G. Theodoridis, S. Theoharis, D. Soudris, C. Goutis, "Method for Minimizing the Switching Activity of Two-Level Logic Circuits", *IEE Proceedings – Computers and Digital Techniques*, vol. 145, no. 5, September 1998, pp. 357-363.
- [The99] G. Theodoridis, S. Theoharis, D. Soudris, C. Goutis, "A New Method for Low Power Design of Two-Level Logic Circuits", *VLSI Design, Journal of Custom-Chip Design, Simulation, and Testing*, vol. 9, no. 2, 1999, pp. 147-158.
- [Tra76] W. Traczyk, „Układy cyfrowe automatyki”, *Wydawnictwa Naukowo-Techniczne*, Warszawa, 1976, pp. 129-158.

- [Tra86] W. Traczyk, „Układy cyfrowe. Podstawy teoretyczne i metody syntezy”, Wydawnictwa Naukowo-Techniczne, Warszawa, 1986, pp. 57-62.
- [Tse97] J-M. Tseng, J-Y. Jou, “A Power Driven Two-Level Logic Optimizer”, Proceedings of Asian and South Pacific Design Automation Conference (ASP-DAC '97), Chiba, Japan, 28-31 January 1997, pp. 113-116.
- [Tse99] J-M. Tseng, J-Y. Jou, “Two-Level Logic Minimization for Low Power”, ACM Trans. on Design Automation of Electronic Systems, vol. 4, no. 1, January 1999, pp. 52-69.
- [Tsu93] C-Y. Tsui, M. Pedram, A. M. Despain, “Efficient Estimation of Dynamic Power Consumption under a Real Delay Model”, Proc. of Int. Conf. on Computer Aided Design (ICCAD'93), Santa Clara, USA, 7-11 November 1993, pp. 224-228.
- [Tsu94] C-Y. Tsui, M. Pedram, A. M. Despain, “Power Efficient Technology Decomposition and Mapping Under an Extended Power Consumption Model”, IEEE Transactions on CAD, vol. 13, no. 9, September 1994, pp. 1110-1122.
- [Tur98] S. Turgis, D. Auvergne, “A Novel Macromodel for Power Estimation in CMOS Structures”, IEEE Transaction on CAD, vol.17, no.11, November 1998, pp. 1090-1098.
- [Vee84] H. J. Veendrik, “Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits”, IEEE Journal of Solid-State Circuits, vol. SC-19, no. 4, August 1984, pp. 468-473.
- [Vru94] S. B. K. Vrudhula, H. Y. Xie, “Techniques for CMOS Power Estimation and Logic Synthesis for Low Power”, Proceedings of the International Workshop on Low Power Design, Napa Valley, USA, April 1994, pp. 21-26.
- [Wan99] Q. Wang, S. Vrudhula, G. Yeap, S. Ganguly, “Power Reduction and Power –Delay Trade Offs Using Logic Transformations” ACM Transactions On Design Automation of Electronics Systems, vol. 4, no.1, January 1999, pp. 97-121.
- [www1] http://www.europpractice.imec.be/europpractice/on-line-docs/prototyping/ti/ti_mtc0u7.html
- [www2] <http://www.austriamicrosystems.com/05foundry/indexc35.htm>
- [www3] http://www.cadence.com/products/custom_ic/veditor/index.aspx
- [Yac89] G. Y. Yacoub, W. H. Ku, “An Enhanced Technique for Simulating Short-Circuit Power Dissipation”, IEEE Journal of Solid-State Circuits, vol. 24, no. 3, June 1989, pp. 844-847.
- [Yua98] L-P. Yan, C-C. Teng, S-M. Kang, “Statistical Estimation of Average Power Dissipation Using Nonparametric Techniques”, IEEE Transaction on VLSI, vol. 6, no. 1, March 1998, pp. 65-73.