

BAZA DANYCH MODELI JEDNOSTEK FUNKCJONALNYCH UKŁADÓW CMOS DLA POTRZEB SYSTEMU REDUKCJI POBORU MOCY

Władysław SZCZEŚNIAK¹, Piotr SZCZEŚNIAK²

1. Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-952 Gdańsk
tel: +4858 3472178 fax: +4858 3471535 e-mail: wlad@pg.gda.pl
2. OBR Centrum Techniki Morskiej Gdynia, 81-109 Gdynia, ul. Dickmana 62
tel: +4858 6665403 fax: +4858 6665495 e-mail: piotr.szczesniak@ctm.gdynia.pl

Streszczenie: Praca dotyczy systemu do przechowywania oraz analizy informacji o modelach jednostek funkcjonalnych układów cyfrowych CMOS. Przedstawia podstawowe wymagania projektu systemu, jego projekt oraz omawia niektóre aspekty jego implementacji. System utworzono w oparciu o środowisko relacyjnych baz danych HSQLDB dostępne na zasadzie wolnego oprogramowania. Został on wybrany ze względu na wysoką wydajność oraz łatwość integracji z tworzonym systemem do redukcji poboru mocy cyfrowych układów CMOS. W jego implementacji zastosowano warstwę pośredniczącą ORM (Object Relational Mapping) umożliwiającą łatwe przystosowanie do współpracy z inną bazą relacyjną danych, w przypadku zmiany wymagań w trakcie eksploatacji.

Słowa kluczowe: redukcja poboru mocy, układy CMOS

1 WSTĘP

Wszystkie algorytmy redukcji poboru mocy wykorzystywane w systemie do redukcji poboru mocy w cyfrowych układach CMOS [1] wykorzystują dane wejściowe w postaci zależności zużywanej mocy od czasu realizacji operacji przez daną jednostkę funkcjonalną. Zależność ta może być wyznaczona przy pomocy technik symulacyjnych zaimplementowanych w postaci systemów takich jak np. *SPICE.

Manualne wyznaczanie tego typu charakterystyk dla wszystkich wykorzystywanych typów jednostek funkcjonalnych i technologii jest zajęciem żmudnym i podatnym na błędy.

Dane uzyskiwane w procesie symulacji są generowane w postaci plików. Zarządzanie nimi odbywa się zatem na poziomie systemu operacyjnego, a to utrudnia szybkie odnajdywanie poszukiwanych wyników, zwłaszcza po dłuższym czasie od wykonania symulacji. Istotny jest także proces konwersji wyników symulacji do formatu wymaganego przez konkretny algorytm/system. Przy próbie organizacji wyników symulacji jedynie z pomocą

folderów systemu operacyjnego dość szybko staje się problemem przechowywania tych samych wyników w kilku różnych formatach oraz koniecznością wielokrotnego powtarzania tych samych symulacji.

Rozwiązaniem przedstawionych problemów jest niżej prezentowany system integrujący program symulacyjny z bazą danych, co pozwala na przechowywanie wyników symulacji w jednym miejscu, ich wygodne przeglądanie i przeszukiwanie, a także analizę oraz ich eksport do wybranego formatu zewnętrznego.

2 PODSTAWOWE WYMAGANIA

Podstawowe wymagania w odniesieniu do projektowanego systemu zostały w skrócie przedstawione we wstępie. Poniżej zostaną one uszczegółowione oraz rozszerzone. Z racji ograniczonego miejsca wyszczególnione zostaną jedynie główne wymagania.

2.1 Symulacja

2.1.1 Projektowany system powinien zapewniać możliwość integracji z systemem komputerowej symulacji układów cyfrowych *SPICE.

2.1.2 Projektowany system powinien zapewniać możliwość uruchomienia symulacji dla grupy modeli.

2.2 Przechowywanie wyników

2.2.1 Projektowany system powinien automatycznie zapisywać wyniki symulacji do bazy danych.

2.2.2 Projektowany system powinien umożliwiać grupowanie wyników.

2.2.3 Powinien on pozwalać na usuwanie zapisanych wyników symulacji.

2.2.4 Projektowany system powinien umożliwiać wprowadzanie adnotacji do pojedynczych wyników oraz całych grup wyników.

2.3 Analiza i eksport wyników

2.3.1 Projektowany system powinien umożliwiać uruchomienie wybranej analizy lub grupy analiz dla dowolnej liczby wyników symulacji przechowywanych w bazie danych.

2.3.2 Powinien on pozwalać na wyeksportowanie wybranych wyników symulacji oraz analiz do pliku zgodnie z wybranym formatem.

2.4 Otwartość systemu

2.4.1 Projektowany system powinien być otwarty, tj. umożliwiać rozszerzanie listy dostępnych analiz oraz formatów zapisu bez konieczności ponownej kompilacji.

2.5 Wydajność i przenośność systemu

2.5.1 Konfiguracja typowego komputera PC powinna być wystarczająca do pracy projektowanego systemu.

2.5.2 Projektowany system powinien być uruchamialny (po rekompilacji lub bez niej) na następujących systemach operacyjnych: Windows XP, Linux.

2.5.3 Powinna istnieć możliwość łatwego przeniesienia zainstalowanego systemu wraz z danymi z jednej stacji roboczej na inną.

2.6 Licencjonowanie

2.6.1 Projektowany system powinien wykorzystywać jedynie produkty dostępne na zasadzie wolnego lub darmowego oprogramowania. Wymaganie to dotyczy środowiska developerskiego, bibliotek wizualizacji oraz silnika bazy danych.

3 WYBÓR TECHNOLOGII

3.1 Wybór języka programowania

Ze względu na dostępność środowiska i innych narzędzi a także łatwość implementacji i przenośność

(wymagania 2.5.1, 2.5.2 oraz 2.6.1), do implementacji systemu wybrano język Java.

3.2 Wybór silnika bazy danych

W projektowanym systemie wydajność bazy danych nie jest wymaganiem kluczowym. Większy nacisk został położony na przenośność systemu oraz zwartość jego instalacji oraz danych. Wymaganie 2.5.3 w zasadzie dyskwalifikuje silniki baz danych wymagające odrębnej instalacji i zarządzania (MySQL, PostgreSQL). Do implementacji systemu wybrany został silnik HSQLDB [4], ze względu na możliwość zagnieżdżenia go w samej aplikacji. Dzięki temu osobna instalacja silnika bazy danych nie jest wymagana, a bazy danych mogą być przenoszone pomiędzy stacjami roboczymi na takiej samej zasadzie jak zwykłe pliki.

Silnik bazy danych HSQLDB cechuje się ponadto dużą wydajnością w aplikacjach tworzonych w języku Java [3], przewyższając w większości testów inne silniki baz danych, które są dostępne na zasadzie wolnego oprogramowania.

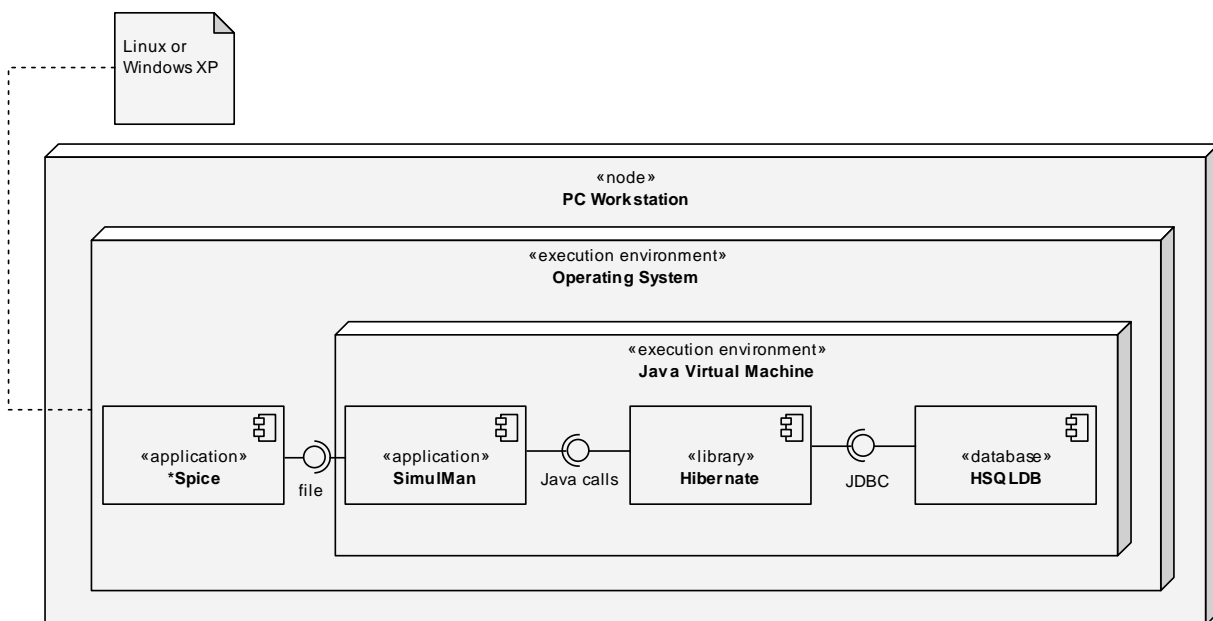
Ze względu jednak na możliwość szybkiej zmiany silnika bazy danych, drugoplanowego znaczenia wydajności systemu względem jego elastyczności oraz skrócenie czasu implementacji, zdecydowano o użyciu warstwy pośredniczącej pomiędzy obiektami Java a relacyjną bazą danych. Z dostępnych środowisk realizujących to zadanie wybrano bibliotekę Hibernate [5], gdyż jest dojrzała i można ją łatwo zastosować.

4 PROJEKT SYSTEMU

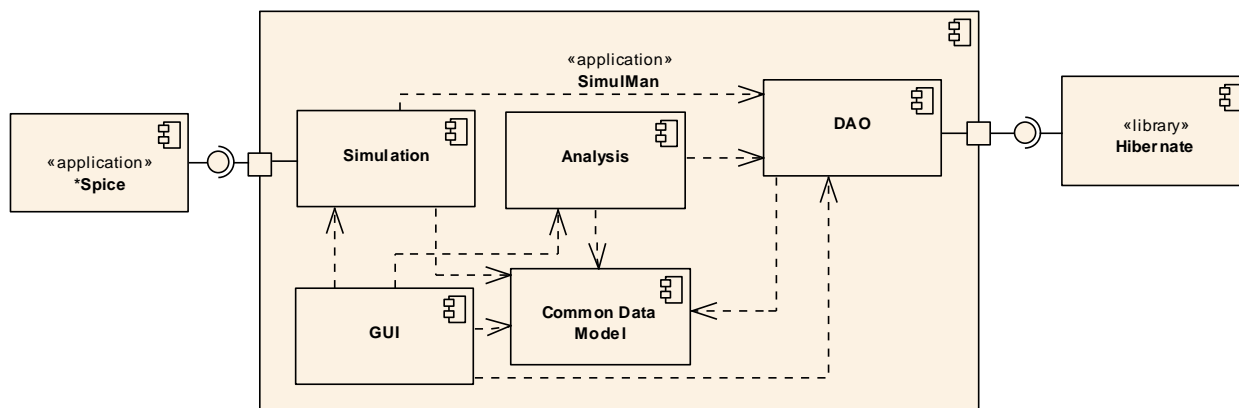
Podrozdział ten przedstawia projekt systemu w postaci diagramów UML.

4.1 Konfiguracja wdrożenia systemu

Konfigurację wdrożenia systemu wynikającą bezpośrednio z wymagań 2.5.1-2.5.3 przedstawia rys. 1.



Rys. 1. Diagram wdrożenia systemu



Rys. 2. Diagram przedstawiający główne komponenty składowe systemu

4.2 Dekompozycja funkcjonalna

W wyniku dekompozycji funkcjonalnej systemu wyróżniono dwa moduły (komponenty) główne oraz dwa moduły (komponenty) pomocnicze, opisane w tabelicy 1.

Tablica 1. Podział aplikacji na moduły

Nazwa	Opis
Common Data Model	Moduł pomocniczy zawierający wspólną definicję modelu danych.
DAO	Moduł pomocniczy zapewniający dostęp do bazy danych.
GUI	Moduł pomocniczy zapewniający graficzny interfejs użytkownika.
Simulation	Moduł główny realizujący funkcjonalność symulacji i zapisu wyników do bazy danych (za pomocą DAO)
Analysis	Moduł główny realizujący funkcjonalność analizy wyników symulacji zapisanych w bazie danych (za pomocą DAO)

Diagram przedstawiający główne i pomocnicze komponenty systemu oraz zależności między nimi oraz komponentami zewnętrznymi został przedstawiony na rys.2.

4.3 Szczegółowa architektura komponentów

4.3.1 Model struktury danych

Wspólny model danych został przedstawiony na rys. 3. Najważniejszymi jego elementami są klasy SingleResult i Plot. Klasa SingleResult reprezentuje wynik symulacji układu, zaś klasa Plot przechowuje dane jednej symulowanej funkcji.

4.3.2 Moduł DAO (Data Access Object)

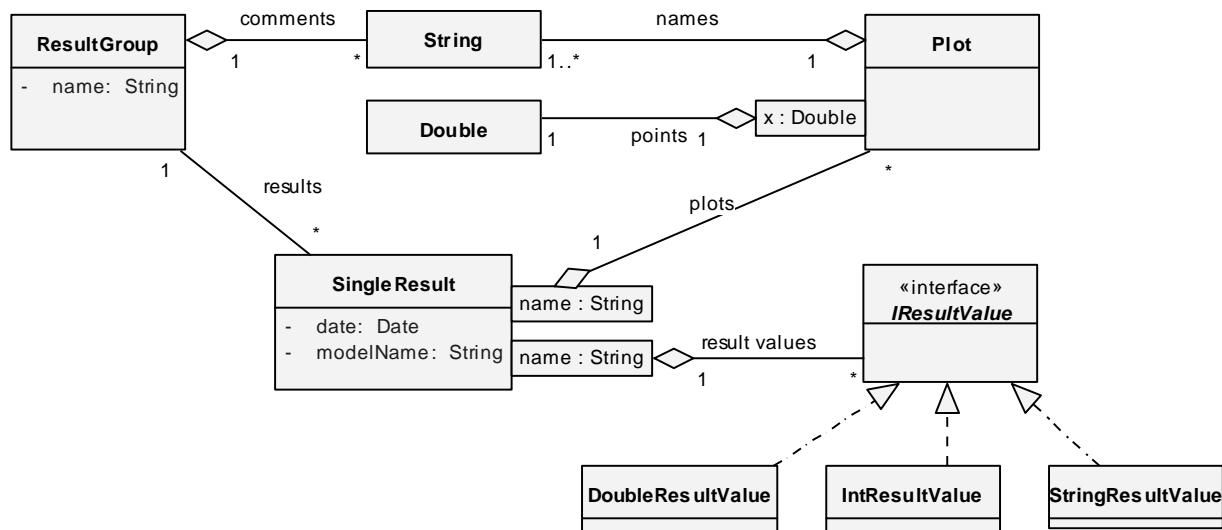
Moduł dostępu do bazy danych jest realizacją wzorca projektowego Data Access Object. Ukrywa on więc w całości fakt wykorzystania biblioteki Hibernate udostępniając innym modułom metody użytkowe.

4.3.3 Moduł GUI

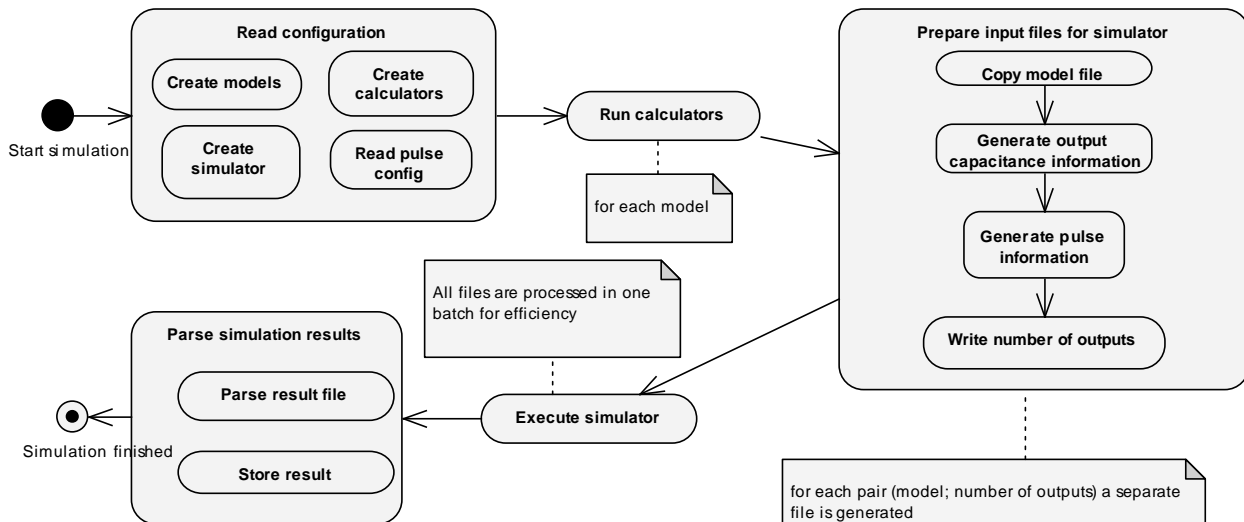
Moduł graficznego interfejsu użytkownika (Graphical User Interface – GUI), pozwala na wykorzystanie funkcjonalności modułów głównych poprzez interfejs graficzny.

4.3.4 Moduł symulacji

Moduł symulacji jest odpowiedzialny za realizację funkcjonalności związanej z przeprowadzeniem symulacji przy pomocy programu *SPICE, wczytanie jej wyników i zapis w bazie danych. Kompletny diagram akcji opisujący czynności wykonywane przez moduł przedstawia rys.4.



Rys. 3. Model danych systemu



Rys. 4. Diagram akcji dla modułu symulacji.

4.3.5 Moduł analizy

Moduł analizy jest odpowiedzialny za realizację funkcjonalności związanej z analizą wybranych wyników przechowywanych w bazie danych. Zawiera on następujące czynności wykonywane przez moduł podczas analizy: retrieve results, run analysers, run postprocessors i run writers.

5 PODSUMOWANIE I WNIOSKI

Prezentowany system spełnia postawione mu wymagania w sposób zadowalający. Jego wykorzystanie pozwala na znaczącą redukcję czasu wykonywania obliczeń. Należy nadmienić, iż w procesie ręcznego pozyskiwania wyników symulacji tylko dla jednego punktu charakterystyki wymagany czas wynosi około jednej minuty.

Zastosowanie prezentowanego systemu integrującego program symulacyjny z bazą danych, pozwala na około tysiąckrotne skrócenie czasu obliczeń.

Ponadto, dzięki otwartej architekturze system umożliwia integrację z innymi systemami, pozwalając na zwiększenie zakresu ich wykorzystania [2].

6 PODZIĘKOWANIA

Niniejsza praca została w części sfinansowana przez Ministerstwo Nauki i Informatyzacji, grant nr 3T11B 015 27.

7 BIBLIOGRAFIA

1. Szcześniak W., Szcześniak P.: Algorytmiczne metody redukcji poboru mocy w cyfrowych układach CMOS, II Krajowa Konf. Technologie Informacyjne, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, Gdańsk 2004, s. 859-866. ISBN 83-917681-5-5.
2. Szcześniak W., Szcześniak P.: Projekt architektury komputerowego systemu do redukcji poboru mocy cyfrowych układów VLSI CMOS, Mat. IV Krajowej Konf. Elektroniki, Kołobrzeg-Darłówek Wschodnie 2005, s. 567-572. ISBN 83-7365-082-2.
3. <http://polepos.sourceforge.net/>, the open source benchmark test.
4. <http://www.hsqldb.org>, strona domowa projektu HSQLDB.
5. <http://www.hibernate.org>, strona domowa projektu hibernate.

DATABASE OF CMOS FUNCTIONAL UNITS' MODELS FOR CIRCUIT POWER REDUCTION SYSTEM

The paper presents the developed software system for storing and analyzing information concerning digital CMOS circuits models'. The system has been designed with application of HSQLDB relational database system being the free software. The system was chosen because of its good performance and easy integration with the designed system for digital CMOS circuits' power reduction. The developed system uses also the ORM (Object Relational Mapping) layer which enables for easy adaptation to another relational database system, in case of requirement change during the maintenance phase.