СИСТЕМАТИЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ПАКЕТА TULIPgm ДЛЯ ПРОЄКТУВАННЯ МІКРО- ТА НАНОЕЛЕКТРОННИХ СИСТЕМ

Бондаренко І.М., Стрілкова Т.О., Грицунов О.В.,
Данілов В.С., Шевченко М.С.

Харківський національний університет радіоелектроніки,
кафедра мікроелектроніки, електронних приладів та пристроїв,
Україна

E-mail: gritsunov@gmail.com

Abstract

The approaches to the systematization of software modules and objects in the TULIPgm package, intended for modeling, analysis and optimization of design processes in microelectronic and nanoelectronic technologies are considered. A structured classification of the functional components of the package is proposed, which includes geometric and topological description modules, objects of physical and technical modeling, tools for visualization and interpretation of results, etc. Systematization ensures consistency between different levels of data representation, increases the efficiency of development and adaptation of software solutions, and contributes to expanding the capabilities of the package for interdisciplinary applications. Particular attention is paid to the issues of interface standardization and support of open data formats, which meets the modern requirements of FAIR principles in scientific research. The results can be used to optimize the design processes of micro- and nanoelectronic systems, as well as to create educational and research environments focused on the integration of innovative technologies.

Вступ

Пакет TULIPgm [1] не лише дозволяє описувати та аналізувати фізичні процеси, що мають місце при функціонуванні мікро- та наноелектронних пристроїв, але й здатен забезпечити комплексну підтримку розробника на всіх етапах проєктування цих виробів. Його функціональні можливості певною мірою охоплюють такі різновиди моделювання:

– Геометричне – створення детальних моделей структур із урахуванням просторових параметрів, що дозволяє відтворювати складні конфігурації мікро- та наноелектронних елементів.

– Фізичне – моделювання електромагнітних, електронних, теплових та інших суттєвих процесів в інтегральних схемах та наноструктурах.

– Аналіз параметрів та оптимізацію – оцінювання впливу конструктивних рішень на інтегральні характеристики, пошук оптимальних умов для досягнення високої ефективності та надійності.

– Візуалізацію результатів – інтерактивне представлення моделей і процесів, що сприяє кращому розумінню складних фізичних явищ та полегшує навчання студентів і дослідників.

Навіщо потрібна систематизація програмних модулів?

Систематизація логічних програмних модулів (ЛПМ) системи TULIPgm необхідна для орієнтування у багаточисленних модулях шляхом встановлення відповідності між ім’ям модуля та його призначенням (місцем у логічній структурі системи). Якщо кількість модулів невелика (не більше кількох десятків), у систематизації немає необхідності, оскільки інформацію про призначення модуля можна отримати з його мнемонічного імені. При великому числі модулів мнемонічне найменування їх стає менш корисним. Краще перейти до числового найменування на основі зарані розробленої ієрархічної системи, на кшталт бібліотечної універсальної десяткової класифікації (УДК). Така система зручна, оскільки при функціональній декомпозиції алгоритмів та структурній декомпозиції даних також утворюються ієрархічні конструкції, тож вдається

певною мірою забезпечити відповідність формальної класифікаційної структури реальній логічній структурі програмного забезпечення.

Коли систематизацію модулів проведено, з імені будь-якого ЛПМ можна, навіть не знаючи конкретного призначення модуля, приблизно визначити його місце у логічній структурі системи (точне визначення, як правило, неможливе, оскільки реальна логічна структура програм не вкладається у формальну схему з обмеженою кількістю рівнів). В результаті полегшується завдання пошуку розташування конкретного ЛПМ в системі, а також обернена задача – знаходження необхідних ЛПМ. Якщо виділити підмножину файлів, які відповідають певному критерію в полі імені, за наявності систематизації ця підмножина буде відповідати певній гілці ієрархічної логічної структури програмного забезпечення, що корисно у процесі розробки та використання модулів системи.

Принципи систематизації програмних модулів і секцій

Розглянемо систематизацію ЛПМ та логічних програмних секцій (ЛПС) у системі TULIPgm. На відміну від УДК, тут використовується шістнадцяткова система числення. Кількість рівнів ієрархічної структури обмежена (5 для модулів і 6 для секцій). Уся сукупність модулів ділиться на шістнадцять класів, пронумерованих цифрами другої шістнадцяткової системи від X до W (табл. 1).

Таблиця 1. Відповідність систем числення

Dec	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Hx1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hx2	X	G	H	J	K	L	M	N	P	Q	R	S	T	U	V	W

Перші вісім класів (від X до N) зарезервовані на віддалену перспективу. Модулі класів P, Q, R і S утворюють однойменні Задачі, з яких на сьогоднішній день реалізовано Задачі класів P (проєктування мікро- та наноелектронних приладів) і S (проєктування НВЧ приладів магнетронного типу). Задачі класів Q та R зарезервовано для проєктування інших типів приладів. Класи модулів T, U і V також зарезервовано на перспективу. Зрештою, модулі класу W утворюють Диспетчер.

Кожен клас ділиться на шістнадцять комплексів ЛПМ, пронумерованих у межах класу цифрами першої шістнадцяткової системи від 0 до F. Призначення конкретних комплексів залежить від класу модулів, до якого вони належать. Кожен комплекс ділиться на шістнадцять підкомплексів ЛПМ, пронумерованих від 0 до F, а кожен підкомплекс – на шістнадцять груп модулів, пронумерованих аналогічно. Група може складатися максимум із шістнадцяти ЛПМ (з номерами від 0 до F).

До складу ЛПМ можуть входити до шістнадцяти ЛПС, пронумерованих цифрами другої шістнадцяткової системи від X до W. Існують чотири види секцій:

– моніторна інформаційна секція (ІС), що включає лише неявні процедурні об’єкти (ПБ) для перезапису інформації між носіями.

– прикладна ІС, яка містить один або кілька інформаційних об’єктів (ІБ), що експортуються.

– моніторна процедурна секція (ПС), що реалізує неявні ПБ для перетворення інформації. На відміну від моніторної ІС, може включати також експортовані модулем синхронні ПБ (тільки процедури), якщо це бажано для підвищення інформаційної міцності секції [2].

– прикладна ПС, що містить один або кілька експортованих синхронних ПБ (процедур або функцій). При цьому цифра X присвоюється моніторним секціям, цифри G … J зарезервовано, цифри K … W присвоюються прикладним ЛПС.

Відповідно до вищесказаного, ім’я ЛПМ має вигляд: Y*abcde*, де Y – ознака ЛПМ; *a* – клас модуля (P … W); *b, c, d* – відповідно комплекс, підкомплекс та група модуля (0 … F); *e* – номер модуля групи (також 0 … F). Ім’я ЛПС має вигляд: *fabcde*, де *f* – номер ЛПС у межах модуля (X … W); символ *a* визначається класом модуля, якому належить секція: для інформаційних ЛПС *a* збігається з класом ЛПМ (P … W), для процедурних секцій *a* – це відповідна цифра першої шістнадцяткової системи (8 … F); символи *b, c, d, e* такі самі, як в ЛПМ.

Наприклад, до умовного ЛПМ YP0123 можуть входити: моніторна ІС XP0123, моніторна ПС X80123, прикладна ІС KP0123 і прикладна ПС K80123. Приклад класифікації модуля Диспетчера:

клас W – Диспетчер;
комплекс W3 – універсальні процедурні об'єкти (бібліотека математичних програм);
підкомплекс W31 – спеціальні функції;
група W312 – циліндричні функції;
модуль YW3121 – функції Бесселя та Неймана дійсного порядку;
секція KF3121 – реалізація функцій Бесселя та Неймана дійсного порядку.

Приклад класифікації модуля Задачі класу P:
клас P – проєктування мікро- та наноелектронних приладів;
комплекс P2 – розрахунок електромагнітних полів;
підкомплекс P21 – розрахунок статичних та квазістатичних полів;
група P213 – розв'язання рівнянь Пуассона для квазістатичних полів;
модуль YP2131 – розв'язання рівнянь Пуассона для квазістатичних полів в просторі взаємодії;
секція K82131 – реалізація розв'язання рівнянь Пуассона для квазістатичних полів в просторі взаємодії.

Розподіл програмних об'єктів (ПГБ) Диспетчера й Задач по ЛПМ виконується з урахуванням забезпечення оптимальних міцності та зчеплення [2, 3]. Повний перелік усіх комплексів, підкомплексів, груп та логічних модулів і секцій системи TULIPgm наведено в програмній документації.

Систематизація програмних об'єктів

Програмним середовищем (program environment) системи TULIPgm називається сукупність ПГБ: ІБ (будь-які дані), синхронних ПБ (процедури, функції), асинхронних ПБ (поки що в системі TULIPgm не реалізовані) та неявних ПБ (блоки обробки подій), доступних у певному місці програми. Ця сукупність може складатися з ПГБ системи, ПГБ модуля, ПГБ секції та локальних ПГБ.

До ПГБ системи відносяться об'єкти, що експортуються ЛПМ для використання іншими ЛПМ. До ПГБ модуля відносяться об'єкти (ІБ, ПБ), експортовані ЛПС деякого модуля для використання іншими секціями лише цього ж ЛПМ. Інші модулі не мають доступу до них. ПГБ секції – це внутрішні об'єкти ЛПС, призначені для використання всіма ПБ, що реалізуються цією ЛПС. Нарешті, локальні ПГБ – це об'єкти, призначені для використання тільки тим ПБ, в якому вони описані.

Пам'ять під ІБ перших трьох видів (ІБ системи, модуля та секції) розподіляється статично, або динамічно під керуванням Диспетчера. Тому значення їх зберігаються протягом усього «прогону» (Run), або поки ці ІБ не будуть примусово видалені з оперативної пам'яті. Пам'ять під локальні ІБ в окремих реалізаціях компіляторів може розподілятися зі стека. Тому значення цих ІБ визначено лише у період активності відповідного ПБ, і у загальному випадку не зберігаються між викликами ПБ.

Ім'я будь-якого ПГБ формується з урахуванням його виду. ПГБ системи іменуються, починаючи з будь-якої літери алфавіту, крім X і Y. Імена ПГБ модуля та секції починаються з літери X. При цьому друга літера імені повинна відрізнятися від X. Ім'я локального ПГБ починається літерами XX. Максимальна довжина імені ПГБ у сучасних мовах та компіляторах достатньо велика, але розробникам системи TULIPgm не рекомендується зловживати надто «мнемонічними» назвами: зазвичай достатньо дворівневих імен загальною довжиною до 31 символи.

Висновки

Використання цифрових позначень замість мнемонічних імен програмних модулів та секцій на перший погляд може здатися дивною ідеєю або, принаймні, вимагати особливого способу мислення програміста. Однак за наявності певного досвіду така ієрархічна система дозволяє, навпаки, спростити пошук потрібних файлів. Так само досвідчені бібліотекари миттєво знаходять потрібну книгу в сховищі, якщо вказати її УДК. За потреби, як компромісний варіант, можна

дублювати числові імена модулів їх мнемонічними аліасами (наприклад, у вигляді YP2131_Poisson).

Література

1. Грицунов О., Доля А. Архітектура програмної системи для проєктування виробів електронної техніки // Матеріали 12-ї Міжнародної науково-технічної конференції «Інформаційні системи та технології» (ІСТ-2023). Частина 1. Х.: ХНУРЕ, с. 90-92 (2024).

2. Ziegler C.A. Programming System Methodologies / C.A.Ziegler. Prentice-Hall, Englewood Cliffs, NJ (1983).

3. Myers G.J. Software Reliability: Principles and Practices / G.J.Myers. Wiley-Interscience, NY (1976).