

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ФУНКЦИОНАЛЬНОЙ СРЕДЫ PIM-СИСТЕМЫ НА ОСНОВЕ ТЕОРИИ НЕЧЕТКИХ МНОЖЕСТВ И ТЕОРИИ ГРАНУЛИРОВАНИЯ

Abstract: The method of construction of mathematical model, formation of its compound components and base portrait of the functional environment of PIM-system is offered. The base portrait contains a feature set (functions) and a set of priority parameters, which the projected PIM-system should have at the chosen type of integrated technology of the PIM-chip to realize the set algorithm of a solved task. The method is based on application of the theory of fuzzy sets and the theory of granulations.

Key words: PIM-systems, fuzzy sets, theory of granulation, information granules, the generalized portrait of a functional environment.

Анотація: Запропоновано метод побудови математичної моделі, формування її складових компонентів і базового портрета функціонального середовища PIM-системи. Базовий портрет містить набір ознак (функцій) і набір пріоритетних параметрів, які повинна мати проектувана PIM-система при обраному типі інтегральної технології PIM-чипа, щоб реалізувати заданий алгоритм розв'язуваного завдання. Метод засновано на застосуванні теорії нечітких множин і теорії гранулювання.

Ключові слова: PIM-системи, нечіткі множини, теорія грануляції, інформаційні гранули, узагальнений портрет функціонального середовища.

Аннотация: Предложен метод построения математической модели, формирования её составных компонентов и базового портрета функциональной среды PIM-системы. Базовый портрет содержит набор признаков (функций) и набор приоритетных параметров, которые должна иметь проектируемая PIM-система при выбранном типе интегральной технологии PIM-чипа, чтобы реализовать заданный алгоритм решаемой задачи. Метод основан на применении теории нечетких множеств и теории гранулирования.

Ключевые слова: PIM-системы, нечеткие множества, теория грануляции, информационные гранулы, обобщенный портрет функциональной среды.

1. Введение

Новый класс компьютерных систем “Процессор-в-памяти” (“Processor-in-memory” или PIM-системы) появился как следствие необходимости решения задач, которые плохо решаются или вообще не поддаются решению на комплексных системах (КС) с классической архитектурой. Однако создание таких систем стало возможным лишь при достижении соответствующего уровня интегральной технологии, что обеспечило размещение на одном кристалле памяти множества процессорных элементов, структурированных определенным образом, образуя новый вид архитектуры, отличающейся от архитектуры классических КС соответствующим набором признаков и функций [1]. Эти признаки и функции определяются, с одной стороны, соответствующими требованиями со стороны алгоритма к параметрам системы, а с другой – возможностями интегральной технологии, которая ограничивает сложность и функциональные возможности компонентов PIM-системы и, в первую очередь, процессоров, размещенных в БИС-памяти, емкость банков памяти, подключенных к процессорам, коммутационной среды между процессорами. Поэтому известные коммерческие PIM-системы строились, опираясь на возможности в соответствующий период времени интегральной технологии, прежде всего, исходя из требований к параметрам системы со стороны алгоритма решаемой задачи [2]. Известная информация по таким системам, как правило, носит описательный характер. При этом авторам не известен подход или метод построения модели, устанавливающий взаимосвязь между требованиями со стороны алгоритма к параметрам и функциям PIM-системы, которые определяются особенностями её архитектурно-структурной организации при применении соответствующего уровня интегральной технологии.

В данной статье авторы предлагают метод построения математической модели функциональной среды PIM-системы, формирования её компонентов и методологию использования этой модели для создания так называемого базового портрета проектируемой PIM-системы. Метод основан на применении теории нечетких множеств и теории гранулирования и позволяет оценить для проектируемой PIM-системы требуемый набор признаков (функций), улучшающих по сравнению с классической КС приоритетные параметры.

2. Основные фрагменты теории нечетких множеств и теории гранулирования, используемые при построении модели

Анализ признаков и результатов реализации некоторых функций, отличающих PIM-системы от классических КС, показал, что только небольшое количество результатов может принимать числовые значения, например, имеющие отношение к технологии создания БИС [2], которые получены на основе анализа зарубежных PIM-систем. Остальные параметры, в связи с тем, что они рассматриваются по отношению к соответствующим параметрам КС с классической архитектурой, определяются нечеткими значениями типа: “в пределах”, “больше”, “больше или равно”, “меньше”, “меньше или равно”, характерны для нечетких множеств. Причем некоторые из параметров могут быть определены лингвистическими переменными типа: “повышение”, “расширение”, “упрощение”, “наращивание”, “ограничение”, “усложнение” и др., что также характерно для нечетких множеств. В связи с этим мы считаем целесообразным при создании модели, отражающей особенности PIM-систем, применить теорию нечетких множеств, используемые положения которой приведены ниже.

Пусть U – универсальное множество, из элементов которого образованы все остальные множества, например, множество $\Omega \subseteq U$. Тогда нечетким множеством называется совокупность пар [3 – 4]:

$$\Omega = \{(x, \mu_{\Omega}(X)) \mid x \in U\}, \quad (1)$$

где $\mu_{\Omega}(x)$ – характеристическая функция (или функция принадлежности), определяемая на отрезке $[0;1]$, т.е. $\mu_{\Omega}(x) : U \rightarrow [0;1]$.

Значение характеристической функции указывает, является ли $x \in U$ элементом множества Ω , и может определяться как

$$\mu_{\Omega}(x) = \begin{cases} 1, & \text{если } x \in \Omega, \\ 0, & \text{если } x \notin \Omega. \end{cases} \quad (2)$$

В общем случае функция принадлежности, помимо значения в интервале $[0;1]$, может принимать значения в интервале $[-1;1]$, а также значения в дистрибутивной решетке L (L – нечеткие множества), в конечном упорядоченном множестве S (S – нечеткие множества). В [5] приведен большой набор свойств для нечетких множеств, которые могут быть использованы при синтезе PIM-систем. Примеры некоторых свойств из этого набора показаны в табл. 1.

Таблица 1. Примеры свойств (ограничений) нечетких множеств

№ п/п	Свойство	Математическое представление	Описание
1	Нормальность	$\exists x \in U : \mu_{\Omega}(x) = 1$	Должен быть хотя бы один элемент, полностью принадлежащий множеству
2	Выпуклость	$\forall a, b, x \in U : a \leq x \leq b \rightarrow \mu_{\Omega}(x) \geq \min\{\mu_{\Omega}(a), \mu_{\Omega}(b)\}$	Значение принадлежности элементов множества в любом интервале не ниже значения принадлежности в экстремальных значениях интервала
3	Одномодалность	$\exists p \in U : (\mu_{\Omega}(p) = \max \mu_{\Omega}(x) \wedge \forall q \neq p : \mu_{\Omega}(q) < \mu_{\Omega}(p))$	Существует только один элемент (прототип) множества с максимальной функцией принадлежности (членства)
4	Упорядоченность	$\forall A, B \in F : A \subseteq B \rightarrow \exists t \in U \forall x \in U : (x \leq t \rightarrow \mu_A(x) \geq \mu_B(x)) \wedge (x \geq t \rightarrow \mu_A(x) \leq \mu_B(x))$	Упорядоченность нечеткого множества отражает упорядоченность значений его принадлежности (членства)
...

При решении сложных задач, имеющих многоуровневую структуру особенно в интерпретации нечетких множеств, целесообразно использовать элементы теории информационного гранулирования, что позволяет:

1) разделить объемную задачу (проблему) на подзадачи, поддающиеся решению согласно одной из известных стратегий;

2) абстрагироваться от мелких, ненужных подробностей для лучшего понимания сущности решаемой задачи. Фактически информационные гранулы являются концептуальными объектами, которые формируются из информации как прямое следствие поисков абстракции, резюмирования и уплотнения информации;

3) использовать в качестве исходной информации данные, представленные в виде нечетких (грубых, приближенных) значений и лингвистических переменных, что практически часто встречается в практике проектирования любых объектов и в том числе – средств вычислительной техники.

Информационная гранула – группа объектов некоторого класса, соединенного на основе подобия или функциональности [6]. Гранула может состоять из объектов различных видов, таких как правила и множества правил. С математической точки зрения, гранулы могут быть обычными множествами объектов, упорядоченными множествами объектов, последовательностями объектов, таблицами объектов и др. Объекты, оформленные в виде гранул, являются продукциями, которые подвергаются различным изменениям и модификациям, и поэтому преобразование гранул – важная проблема в гранулированном вычислении.

Информационную систему можно представить в виде пары $I = (U, A)$ конечных непустых множеств объектов и атрибутов соответственно, где множество всех объектов U является областью I [6]. Элементы U (соответственно A) обозначаются u (соответственно a), возможно, – с индексами. Каждый атрибут a рассматривается как отображение $a:U \rightarrow V_a$, которое назначает для каждого объекта u значение $a(u) \in V_a$. Пусть $V = \bigcup\{V_a \mid a \in A\}$. Любую пару (a, v) , где $a \in A$ и $v \in V_a$, называют дескриптором (описателем).

С математической точки зрения, каждый объект u может быть рассмотрен как отображение $u: A \rightarrow V$ таким образом, что $a(u) \in V_a$, то есть $U \subseteq \Pi\{V_a \mid a \in A\}$. Каждый объект u является, таким образом, набором дескрипторов $u = \{(a, a(u)) \mid a \in A\}$. Поэтому гранулирование объектов на основе информации, содержащейся в I , и формирование гранул из множеств дескрипторов от I – эквивалентно.

Отображения на U , назначающие информационные гранулы объектов из U , называют *отображениями гранулирования*.

Предположим, что для каждого атрибута $a \in B \subseteq A$ показатель (метрика) и пороговая функция δ_a, f_a соответственно заданы. Тогда семейство отображений неопределенности (гранула) $\Gamma_a: U \rightarrow \wp U$, где \wp – мощность множества U , параметризованного атрибутами от B , можно определить следующим образом для любых объектов u, u' :

$$u' \in \Gamma_a u \leftrightarrow \delta_a(a(u), a(u')) \leq f_a(a(u), a(u')). \quad (3)$$

При этом u' – член *гранулы*, назначенной на u отображением Γ_a , если и только если расстояние между значениями a в u, u' , обозначенное δ_a , не превышает порогового значения $f_a(a(u), a(u'))$.

Глобальное отображение неопределенности (гранулы) Γ может быть определено с помощью операций над множествами, например, с использованием операции пересечения отображений Γ_a , то есть

$$\Gamma u \stackrel{def}{=} \bigcap \{\Gamma_a u \mid a \in B\}. \quad (4)$$

3. Математическая модель функциональной среды PIM-системы

Формальное представление модели. Обозначим набор качественных и количественных признаков (функций) функциональной среды PIM-системы функционалом E , включающим множества, которые назовем соответствующими срезами функциональной среды:

$$E = \{S, W, C, P, A, T\}, \quad (5)$$

где S – структурно-архитектурный срез, W – срез программного обеспечения, C – срез управления процессами внутри чипа, P – пользовательский срез, A – алгоритмический срез, T – технологический срез.

Каждый срез выражения (5) представлен множеством элементов, обозначенных соответствующими маленькими буквам, т.е. $s_j \in S, w_q \in W, c_h \in C, p_r \in P, a_i \in A, t_k \in T$, где j, q, h, r, i, k определяют номера элементов множеств, а их количество определяет мощности $m_j, m_q, m_h, m_r, m_i, m_k$.

Определим множество приобретенных РІМ-системой свойств, вследствие особенностей её архитектурно-структурной организации по сравнению с КС с классической архитектурой, в виде

$$\Omega = \{\beta, \psi, \gamma, \varepsilon, \lambda, z, v, b, d, y, \sigma\}, \quad (6)$$

- где β – повышение производительности РІМ-системы;
 ψ – расширение функциональных возможностей;
 γ – эффективное использование ресурсов;
 ε – упрощение диагностики и отладки;
 λ – уменьшение количества БИС*);
 z – упрощение программирования;
 v – наращивание (масштабирование) ресурсов РІМ-системы;
 b – наращивание (масштабирование) параллелизма;
 d – ограничение на использование ресурсов памяти и средств обработки;
 y – повышение трудоемкости разработки функциональной среды;
 σ – усложнение распределения ресурсов при распараллеливании алгоритма.

**) Примечание: следует иметь в виду, что уменьшение количества БИС приводит к уменьшению потребляемой мощности, габаритов, веса и повышению надежности РІМ-системы.*

В общем случае набор параметров (свойств) выражения (6) может быть изменен или дополнен, например, параметрами надежности. Это, в принципе, не изменяет сущность подхода к построению модели РІМ-систем. Авторы выбрали наиболее очевидные параметры, которые по отношению к соответствующим параметрам классических КС приобретают новые значения (свойства) благодаря особенностям архитектурно-структурной организации РІМ-систем.

Принимая во внимание информацию, приведенную выше, функциональная модель среды, с точки зрения логической интерпретации, может быть представлена в виде

$$E = \{(\forall a \in A)[\exists U \in E]: (f(U) = \omega_k, \omega_k \in \Omega) \rightarrow \Gamma_{ОБП}\} |_T, \quad (7)$$

$$U = \{S, W, C, P\},$$

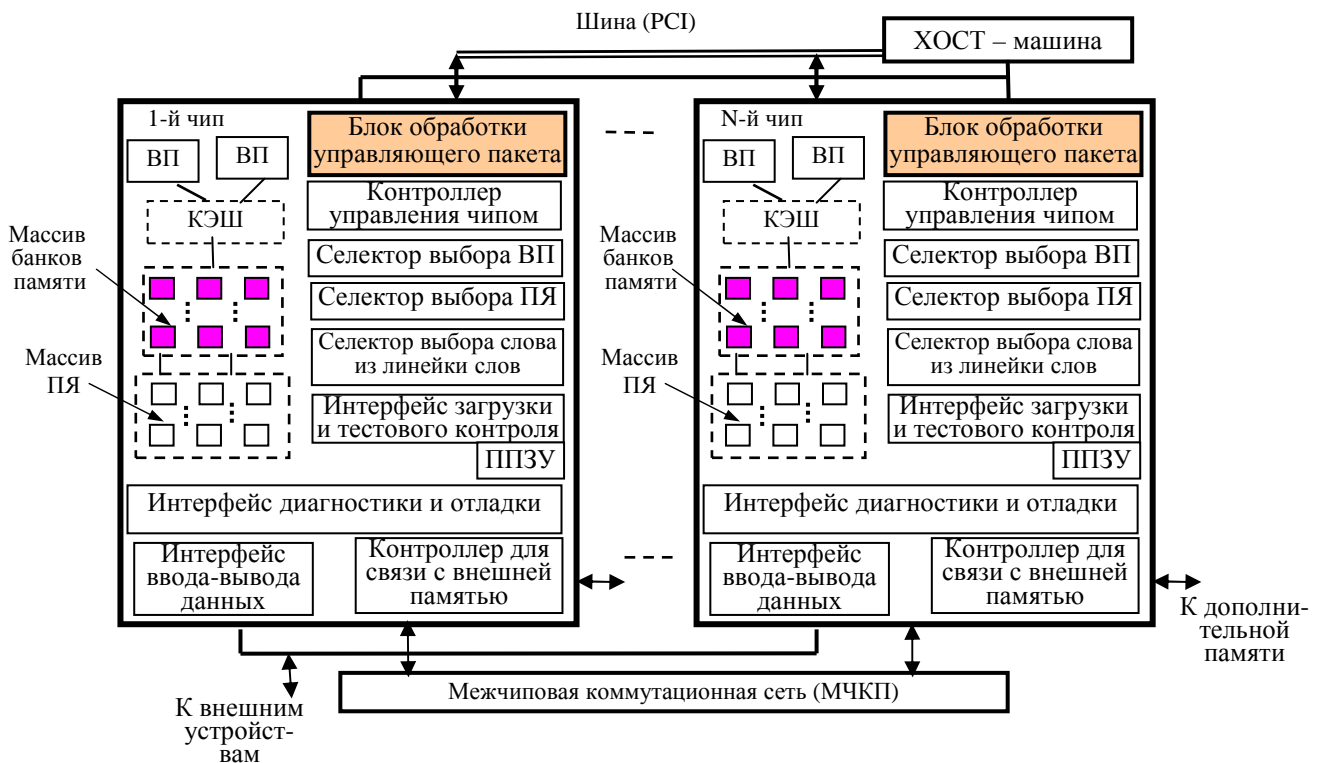
т.е. для всех значений параметров a , являющихся элементами множества A , существует множество срезов $U \in E$, отражающих признаки (функции), которые отличают РІМ-систему от КС с классической архитектурой, таких, что результаты $f(U)$ их реализации, представленные через параметры множества Ω , отображены в виде параметров обобщенного “портрета” $\Gamma_{ОБП}$ функциональной среды при ограничениях со стороны параметров технологического среза T на создание РІМ-системы.

Под обобщенным “портретом” $\Gamma_{ОБП}$ будем понимать множество доминирующих признаков (функций) функциональной среды РІМ-системы, представленных в гранулах соответствующих срезов множества U с учетом их функций принадлежности множеству $\Gamma_{ОБП}$.

Описание компонентов, входящих в состав модели. Рассмотрим более подробно основные составляющие модели (7). Прежде всего, на основе анализа качественных и количественных признаков известных PIM – систем выделим их наиболее характерные особенности по сравнению с классическими компьютерными системами, которые сгруппируем по срезам S, W, C, P множества U и представим в табл. 2 – табл. 7 соответственно [1, 2, 7, 8], принимая во внимание достижения интегральной технологии на настоящее время, обеспечивающие интеграцию компонентов на одном кристалле, отображенных на рис. 1.

Таблица 2. Структурно-архитектурный срез ($s_j \in S$)

s_j	Описание признаков (функций), отличающих PIM-систему от КС с классической архитектурой	Влияние на параметры PIM	Определяющий объект (фактор)	
s_1	Широкая полоса пропускания по каналу процессор-память (существенно больше по сравнению с классическими КС)	β_s, b_s	Архитектура и уровень интегральной технологии	
s_2	Ограниченные объемы банков памяти, размещенных на PIM-чипе, по сравнению с классическими системами (до нескольких Мбайт)	d_s		
s_3	Ограниченная мощность массовых ПЯ в составе PIM относительно традиционных вычислительных платформ (от 1-разр. до 32-разр. ПЯ с усеченным набором команд)		β_s	Тип задачи и уровень интегральной технологии
s_4	В качестве ВП в составе PIM-чипа используются как серийные, так и специализированные процессоры			
s_5	В качестве средств коммутации внутри чипа используются скоростные коммутаторы (селекторы выбора), а между чипами – высокоскоростная межчиповая сеть	λ_s	Архитектура и уровень интегральной технологии	
s_6	Контроллер управления чипом размещен на чипе памяти и обеспечивает согласованное взаимодействие функциональных узлов PIM-чипа			
s_7	Блок обработки управляющего пакета (БОУП), размещенный на чипе, используется для расшифровки поступающего в чип пакета и выработки управляющих сигналов для всех узлов, размещенных на чипе			
s_8	Интерфейс ввода-вывода данных, размещенный на чипе, используется для связи чипа системы памяти с другими подобными чипами или совместимыми с ним устройствами ввода – вывода	v_s		
s_9	Контроллер, размещенный на чипе, для связи PIM-чипа с внешней памятью используется при применении PIM-чипа в качестве обычной классической оперативной памяти, дополняющей память хост-машины	ψ_s, λ_s		
s_{10}	Интерфейс загрузки, размещенный на чипе, предназначен для загрузки в КЭШ команд ВП программы и данных			
s_{11}	Полупостоянное запоминающее устройство (ППЗУ), размещенное на чипе, используется для поддержки системной программы начальной загрузки	ε_s		
s_{12}	Интерфейс диагностики и отладки на чипе предназначен для проведения внешней экспертизы состояний процессорных элементов, размещенных на чипе системы памяти, и выполнения их отладки			
s_{13}	Компилятор языка высокого уровня (ЯВУ), размещенный на чипе, упрощает программирование	z_s, λ_s		Входной язык пользователя
s_{14}	Создание (модификация) системы команд ВП, микропрограммы для ВП, системы команд ПЯ, микропрограммы для ПЯ повышает трудоемкость разработки функциональной среды PIM-системы	y_s		Набор операций и организация процесса обработки информации
---	-----	-----	-----	



Обозначения: ВП – ведущий процессор чипа; КЭШ – память типа КЭШ; ПЯ – процессорные ядра, подключенные к банкам памяти; ППЗУ – полупостоянное ЗУ загрузки.

Рис. 1. Состав компонентов на одном кристалле, отражающий особенности организации PIM-чипа при современном уровне интегральной технологии

Таблица 3. Срез программного обеспечения ($w_q \in W$) [7]

w_q	Описание признаков (функций), отличающих PIM-систему от КС с классической архитектурой	Влияние на параметры PIM	Определяющий фактор
w_1	Модифицированная операционная система (ОС ₁), например, модифицированная Unix, Linux и т.п.	β_w, γ_w	Архитектура PIM-системы
w_2	Компилятор языка высокого уровня (КМЯ), реализованный на PIM-чипе	z_w	Уровень входного языка
w_3	Библиотеки прикладных программ различного назначения (БПП)		Тип задачи пользователя
w_4	Программные средства, поддерживающие функционирование блока обработки управляющего пакета (ПО/БОУП)	γ_w	Архитектура PIM-системы
w_5	ПО формирования управляющего пакета		Архитектура PIM-системы и тип задачи пользователя
w_6	Программные средства для поддержки работы межчиповой коммутационной сети – МЧКП (ПО/МЧКП) и иерархической системы памяти (ПО/ПАМ) в различных режимах	β_w	Архитектура PIM-системы
w_7	Программные средства: для распределения областей банков памяти (ПО/БП) для программ и данных, а также для распределения и размещения данных в эти области (ПО/РД), для дефрагментации (ПО/ДФ) и сборки "мусора" и т.д., а также – для управления работой КЭШ – ВП с учетом процедур когерентности	γ_w	
w_8	Набор специальных подпрограмм (СПП) для управления созданием, выполнением, взаимодействием и перемещением макросерверов		
w_9	Программные средства для загрузки памяти чипа и выполнения тестирования работоспособности памяти и всех компонентов чипа в целом (ПО/ТСТ)	ϵ_w	Архитектура PIM-системы и уровень технологии

w_{10}	Набор системных сервисных программ (ПО разбиения алгоритма на гранулы трех уровней: системный, узловой и блочный)	β_w, γ_w	Архитектура PIM-системы и тип задачи пользователя
w_{11}	Программные средства для поддержки функционирования интерфейса (ПО/ИНЧ), размещенного на чипе, для связи с другими чипами и при необходимости для связи с дополнительной памятью и внешними устройствами	v_w	
w_{12}	Программные средства для изменения разрядности обрабатываемых данных и масштабирования параллелизма	b_w	

Таблица 4. Срез управления процессами внутри чипа ($c_h \in C$)

c_h	Описание признаков (функций), отличающих PIM-систему от КС с классической архитектурой	Влияние на параметры PIM	Определяющий фактор
c_1	Управление процессами PIM-системы основано на концепции потоковой обработки информации и управляемого сообщением вычисления	β_c, b_c	Архитектура PIM-системы и тип задачи пользователя
c_2	Поддержка управляемого сообщением вычисления основана на концепции пакетов, которые содержат значения параметров и спецификаторы действий, а также дополнительные поля, необходимые для транспортировки, обнаружения ошибок, маршрутизации и управления контекстом		
c_3	Режимы работы PIM-чипа (в качестве средства обработки или в качестве классической памяти) определяются соответствующими полями пакета	ψ_c	
c_4	Макросервер определяется как основа организации всего вычислительного процесса в PIM	z_c, γ_c	Архитектура PIM-системы
c_5	Каждый узел PIM (PIM-чип) работает под управлением своей собственной операционной системы вместо одной для всех узлов машины, как это делается для машин с классической архитектурой (SMP). Такая операционная система является усеченным вариантом (модификацией) классической	γ_c, λ_c, y_c	
c_6	Каждый узел PIM управляет собственной памятью и вычислительными ресурсами при сохранении сильной интеграции на одном чипе БИС. Достижение рабочего равновесия между этими требованиями остается важным фактором для системных служб PIM		
c_7	Узлы PIM общаются с другими узлами путем передачи сообщений, а не чтением и записью разделяемой памяти	β_c	
c_8	Используется мультипроцессирование для быстрого удовлетворения входящих запросов на обслуживание и обеспечение перекрытия по времени выполнения вычислений, коммуникаций и доступа к памяти		
---	-----	-----	-----

Таблица 5. Пользовательский срез ($p_r \in P$)

p_r	Описание признаков (функций), отличающих PIM-систему от КС с классической архитектурой	Влияние на параметры PIM	Определяющий фактор
p_1	В основу программирования положен объектно-ориентированный подход. Пользователи создают объекты, которые устанавливают обработчики для входящих сообщений или пакетов. Операционная система PIM использует объекты как основу для вычисления	z_p	Концепция программирования
p_2	Существующие операционные системы КС должны быть доработаны для всей PIM-системы из-за многоступенчатой иерархичности ресурсов обработки и хранения информации	y_p	Архитектура PIM-системы
p_3	Для каждого узла PIM используется своя операционная система (например, модификация известной). При этом у узлов PIM нет доступа к информации о состоянии других узлов, в отличие от узлов классических КС (SMP)	y_p, v_p	
p_4	Операционная система узла PIM (модифицированный вариант стандартной) должна занимать в памяти небольшой объем так, чтобы наибольшая часть ресурсов PIM была доступна для приложений	d_p, σ_p	Архитектура PIM-системы
p_5	Узел PIM может получить отдаленные запросы памяти. ОС PIM использует быстрый запрос с низкими накладными расходами	v_p, γ_p	

P_6	Операционная система PIM должна обеспечивать возможность для разработчиков программного обеспечения создавать новые типы служб	Ψ_p	
P_7	Сервисные программы (распределение памяти, размещение данных и др.) классических КС также должны быть доработаны для использования их в среде PIM- системы	Y_p	
P_8	PIM интегрируются в очень больших количествах, обеспечивая большую возможность параллелизма и распределенных вычислений, чем их традиционные кластерные копии	v_p, b_p, β_p	
P_9	Прикладные программы, работающие в своем собственном виртуальном адресном пространстве, выполняются как асинхронные сети динамических объектов, названные макросерверами	z_p	Концепция программирования
P_{10}	Для управления созданием, выполнением, взаимодействием и перемещением макросерверов используется набор специальных подпрограмм, который располагается в любой PIM-вершине (узле)	Y_p	
P_{11}	Эффективность коммуникации делает системы PIM лучшими для мелко модульного распределенного вычисления, чем их традиционные кластерные копии	γ_p, b_p, β_p	Архитектура PIM-системы
P_{12}	Модель выполнения PIM позволяет не только данным перемещаться к вычислению (обработчику), но и вычислению к данным	γ_p	
P_{13}	Обеспечивается (по сравнению с классическими КС) снижение габаритов, веса, потребляемой мощности за счет уменьшения количества БИС для всей PIM-системы	λ_p	
---	-----	---	-----

Примечание: Информация, приведенная в табл. 5, не является руководством к действию системного разработчика и программиста, она лишь освещает особенности PIM-систем, на которые программист в первую очередь должен обратить внимание.

Таблица 6. Алгоритмический срез ($a_i \in A$)

a_i	Описание признаков (функций), отличающих PIM-систему от КС с классической архитектурой	Требуется для реализации функций	Определяющий фактор
a_1	Возможность разделения реализуемого алгоритма на три уровня: системный, узловой и блочный	β, b, γ	Тип задачи пользователя
a_2	Массовое обращение к памяти за данными на всех уровнях реализации распределенного алгоритма	γ, b, v	
a_3	Наличие часто повторяющихся участков алгоритма, реализация которых требует максимальных ресурсов памяти и производительности	β, b, v, γ	
a_4	Обработка данных большой разрядности (≥ 64 бит)	b, v	
a_5	Высокая степень параллелизма алгоритма решаемой задачи с помощью ресурсов одного кристалла (чипа)	β, γ, λ	Архитектура PIM-системы
a_6	Масштабирование параллелизма за счет наращивания PIM-чипов (до нескольких тысяч чипов)	β, b, v	
a_7	Разделение алгоритма на узловом уровне для балансировки загрузки ВП и ЯП (более мелкие модули по сравнению с классическими КС)	γ	Архитектура и квалификация программиста
---	-----	-----	-----

Таблица 7. Технологический срез ($t_k \in T$)

t_k	Описание признаков (функций), отличающих PIM-систему от КС с классической архитектурой	Требуется для реализации функций	Определяющий фактор
t_1	Тип используемого кристалла для создания PIM-системы	С жесткой логикой или с программируемыми логическими схемами (ПЛИС)	Назначение системы и уровень интегральной технологии
t_2	Технологический уровень создания элементов на кристалле	Размер элемента $\leq f(t_{2max}), \text{ мк}$	Уровень интегральной технологии
t_3	Размеры кристалла БИС	$\leq f(t_{3max}), \text{ мм}^2$	
t_4	Совместимость интегральных технологий создания логических элементов и элементов памяти на одном кристалле	Новый тип технологии	

t_5	Максимальное количество компонентов, размещаемых на кристалле	$\geq 10^7$, штук	
t_6	Максимальная задержка срабатывания элемента	$\leq f(t_{6\max})$, пкс	
t_7	Тип корпуса БИС	$f(t_{7\max})$, ГГц	
t_8	Тип корпуса БИС	Имеющиеся типы корпусов	Тип микро-процессора и требования ТЗ
t_9	Количество контактов БИС	$\geq f(t_{9\max})$, штук	Тип корпуса и требования ТЗ
t_{10}	Тип и параметры приспособления для охлаждения БИС	Имеющиеся типы систем воздушного охлаждения	Тип микро-процессора и условия эксплуатации
---	-----	-----	-----

Используя основные положения теории множеств и теории гранулирования, сформируем набор факторов (функций), определяющий так называемый функционально-структурный портрет РИМ-системы, среда которой представлена моделью, описанной выражениями (5) – (7).

Формирование информационных гранул. Используя приведенные выше положения теории нечетких множеств и теории гранулирования, построим гранулы для каждого среза модели (5), элементы которых приведены в табл. 2 – 5. Для этого сформируем сводную таблицу 8, отражающую только элементы признаков (функций) каждого среза множества U и их влияние на параметры РИМ-систем. В общем случае для формирования гранул будем считать, что функции принадлежности для всех элементов каждого среза равны 1, т.е. $\mu_s(x) = 1$ для всех $s \in S$, $\mu_w(x) = 1$ для всех $w \in W$, $\mu_c(x) = 1$ для всех $c \in C$, $\mu_p(x) = 1$ для всех $p \in P$, а значение любого параметра, принадлежащего грануле, равно значению этого параметра, размещенного в соответствующей таблице:

$$u' \in \Gamma_s u \stackrel{def}{\Leftrightarrow} s_j(u') = s_j(u); \quad u' \in \Gamma_w u \stackrel{def}{\Leftrightarrow} w_q(u') = w_q(u);$$

$$u' \in \Gamma_c u \stackrel{def}{\Leftrightarrow} c_h(u') = c_h(u); \quad u' \in \Gamma_p u \stackrel{def}{\Leftrightarrow} p_r(u') = p_r(u).$$

Таблица 8. Сводная таблица элементов всех срезов множества U

Срез ($s_j \in S$)		Срез ($w_q \in W$)		Срез ($c_h \in C$)		Срез ($p_r \in P$)	
s_j	Результат	w_q	Результат	c_h	Результат	p_r	Результат
s_1	β_s, b_s	w_1	β_w, y_w	c_1	β_c, b_c	p_1	z_p
s_2	d_s	w_2	z_w	c_2		p_2	y_p
s_3		w_3		p_3		y_p, v_p	
s_4	β_s	w_4	γ_w	c_3	ψ_c	p_4	d_p, σ_p
s_5		w_5		c_4	z_c, γ_c	p_5	v_p, γ_p

s_6	λ_s	w_6	β_w	c_5	$\gamma_c, \lambda_c,$ y_c	p_6	ψ_p
s_7		w_7	γ_w	c_6		p_7	y_p
s_8	v_s	w_8			c_7	p_8	v_p, b_p, β_p
s_9	ψ_s, λ_s	w_9	ϵ_w	c_8	β_c	p_9	z_p
s_{10}		w_{10}	β_w, γ_w			p_{10}	y_p
s_{11}		w_{11}	v_w			p_{11}	b_p, β_p, γ_p
s_{12}	ϵ_s	w_{12}	b_w			p_{12}	γ_p
s_{13}	z_s, λ_s					p_{13}	λ_p
s_{14}	y_s						

Тогда гранулы, сформированные по принципу подобия элементов множества Ω в соответствующих срезах, будут иметь вид

$$\left. \begin{aligned}
 \Gamma_{\beta}u \rightarrow U &= \xi_{\beta} \{ \beta_s, \beta_w, \beta_c, \beta_p \} \supset \{ \{s_1, s_4, s_5\}, \{w_1, w_6, w_{10}\}, \{c_1, c_2, c_8\}, \{p_8, p_{11}\} \}, \\
 \Gamma_b u \rightarrow U &= \xi_b \{ b_s, b_w, b_c, b_p \} \supset \{ \{s_1\}, \{w_{12}\}, \{c_1, c_2\}, \{p_8, p_{11}\} \}, \\
 \Gamma_z u \rightarrow U &= \xi_z \{ z_s, z_w, z_c, z_p \} \supset \{ \{s_{13}\}, \{w_2, w_3\}, \{c_4\}, \{p_1, p_9\} \}, \\
 \Gamma_y u \rightarrow U &= \xi_y \{ y_s, y_w, y_c, y_p \} \supset \{ \{s_{14}\}, \{w_1\}, \{c_5, c_6, c_7\}, \{p_2, p_3, p_7, p_{10}\} \}, \\
 \Gamma_{\lambda} u \rightarrow U &= \xi_{\lambda} \{ \lambda_s, \lambda_c, \lambda_p \} \supset \{ \{s_6, s_7, s_9, s_{10}, s_{11}, s_{13}\}, \{c_5, c_6, c_7\}, \{p_{13}\} \}, \\
 \Gamma_{\gamma} u \rightarrow U &= \xi_{\gamma} \{ \gamma_w, \gamma_c, \gamma_p \} \supset \{ \{w_4, w_5, w_7, w_8, w_{10}\}, \{c_5, c_6, c_7\}, \{p_5, p_{11}, p_{12}\} \}, \\
 \Gamma_v u \rightarrow U &= \xi_v \{ v_s, v_w, v_p \} \supset \{ \{s_8\}, \{w_{11}\}, \{p_3, p_5, p_8\} \}, \\
 \Gamma_{\psi} u \rightarrow U &= \xi_{\psi} \{ \psi_s, \psi_c, \psi_p \} \supset \{ \{s_9, s_{10}, s_{11}\}, \{c_3\}, \{p_6\} \}, \\
 \Gamma_{\epsilon} u \rightarrow U &= \xi_{\epsilon} \{ \epsilon_s, \epsilon_w \} \supset \{ \{s_{12}\}, \{w_9\} \}, \\
 \Gamma_d u \rightarrow U &= \xi_d \{ d_s, d_p \} \supset \{ \{s_2, s_3\}, \{p_4\} \}, \\
 \Gamma_{\sigma} u \rightarrow U &= \xi_{\sigma} \{ \sigma_p \} \supset \{ \{p_4\} \}.
 \end{aligned} \right\} \quad (8)$$

Объединяя гранулы (8) для всех значений $\Omega = (\beta, \lambda, b, z, \gamma, v, \psi, y, d, \epsilon, \sigma)$, получим гранулу, включающую все свойства (признаки) всех срезов множества U . Такую гранулу назовем обобщенным портретом $\Gamma_{ОБП}$ функциональной среды РІМ- системы, которая через элементы множеств может быть представлена в виде

$$\begin{aligned}
 \Gamma_{ОБП} \rightarrow U &= \cup \Gamma_{\Omega U} = \{ \{ \beta_s, \beta_w, \beta_c, \beta_p \}, \{ b_s, b_w, b_c, b_p \}, \{ z_s, z_w, z_c, z_p \}, \{ y_s, y_w, y_c, y_p \}, \\
 &\{ \lambda_s, \lambda_c, \lambda_p \}, \{ \gamma_w, \gamma_c, \gamma_p \}, \{ v_s, v_w, v_p \}, \{ \psi_s, \psi_c, \psi_p \}, \{ \epsilon_s, \epsilon_w \}, \{ d_s, d_p \}, \{ \sigma_p \} \} = \\
 &\{ \{ \{s_1, s_4, s_5\}, \{w_1, w_6, w_{10}\}, \{c_1, c_2, c_8\}, \{p_8, p_{11}\} \}, \{ \{s_1\}, \{w_{12}\}, \{c_1, c_2\}, \{p_8, p_{11}\} \}, \\
 &\{ \{s_{13}\}, \{w_2, w_3\}, \{c_4\}, \{p_1, p_9\} \}, \{ \{s_{14}\}, \{w_1\}, \{c_5, c_6, c_7\}, \{p_2, p_3, p_7, p_{10}\} \}, \\
 &\{ \{s_6, s_7, s_9, s_{10}, s_{11}, s_{13}\}, \{c_5, c_6, c_7\}, \{p_{13}\} \}, \{ \{w_4, w_5, w_7, w_8, w_{10}\}, \{c_5, c_6, c_7\}, \{p_5, p_{11}, p_{12}\} \}, \\
 &\{ \{s_8\}, \{w_{11}\}, \{p_3, p_5, p_8\} \}, \{ \{s_9, s_{10}, s_{11}\}, \{c_3\}, \{p_6\} \}, \{ \{s_{12}\}, \{w_9\} \}, \{ \{s_2, s_3\}, \{p_4\} \}, \{ \{p_4\} \} \}.
 \end{aligned} \quad (9)$$

Используя представление элементов множеств в табл. 8, можно получить гранулы для каждого среза множества U . Например, для структурно-архитектурного среза S получим

$$\Gamma_{sU} \rightarrow U = \{\{s_1, s_4, s_5\}, \{s_6, s_7, s_9, s_{10}, s_{11}, s_{13}\}, \{s_9, s_{10}, s_{11}\}, \{s_2, s_3\}, \{s_1\}, \{s_8\}, \{s_{12}\}, \{s_{13}\}, \{s_{14}\}\}.$$

Анализ выражений (8) показывает, что существуют гранулы, параметры которых присутствуют в большинстве типов срезов, в данном случае в четырех и трех типах срезов функциональной среды РИМ-системы, входящих в множество U : структурно-архитектурном (S); программного обеспечения (W); управления процессами внутри чипа (C) и пользовательском (P). Такой набор гранул назовем доминирующим, а соответствующий набор параметров $\Omega_1 = \{\{\beta, b, z, y, \lambda, \gamma, v_w\} \subseteq \Omega\}$ – определяющим, поскольку эти параметры в основном определяют так называемый функционально-структурный “портрет” РИМ-системы, т.е.

$$\Gamma_{дг} = \{\Gamma_{\beta U}, \Gamma_{bU}, \Gamma_{zU}, \Gamma_{yU}, \Gamma_{\lambda U}, \Gamma_{\gamma U}, \Gamma_{vU}, \Gamma_{\psi U}\}. \quad (10)$$

Оценим влияние параметров алгоритмического среза A на функционально-структурный “портрет” РИМ-системы $\Gamma_{дг}$. Как правило, от алгоритма исходит вполне определенный набор требований к параметрам системы. В данном случае, поскольку мы опираемся на особенности РИМ-систем по сравнению с классическими КС, то следует выделить лишь требования на изменения соответствующих параметров множества $\Gamma_{дг}$, которые продиктованы особенностями реализации функций или специфических признаков РИМ-системы. С этой точки зрения такой набор требований к номенклатуре параметров можно назвать параметрическим шаблоном для остальных срезов функциональной среды. Для построения параметрического шаблона обратимся к табл. 6.

Гранула алгоритмического среза может быть представлена в виде

$$\Gamma_{aU'} \rightarrow U' = \xi_a \{\beta, b, \lambda, \gamma, v\} \supset \{\{a_1, a_3, a_5, a_6\}, \{a_1, a_2, a_3, a_4, a_6\}, \{a_5\}, \{a_1, a_2, a_3, a_5, a_7\}, \{a_2, a_3, a_4, a_6\}\}. \quad (11)$$

При этом набор параметров параметрического шаблона будет иметь вид

$$\Omega_2 = \{\beta, b, \lambda, \gamma, v\}. \quad (12)$$

Согласно (6) и (8), $\beta = \{\beta_s, \beta_w, \beta_c, \beta_p\}$, $b = \{b_s, b_w, b_c, b_p\}$, $\lambda = \{\lambda_s, \lambda_w, \lambda_c, \lambda_p\}$, $\gamma = \{\gamma_w, \gamma_c, \gamma_p\}$, $v = \{v_s, v_w, v_p\}$. Тогда набор гранул, выделенных из (10) с помощью параметрического шаблона (12), составит:

$$\begin{aligned} \Gamma_{aUш} = & \{\{\Gamma_{\beta U}\}, \{\Gamma_{bU}\}, \{\Gamma_{\lambda U}\}, \{\Gamma_{\gamma U}\}, \{\Gamma_{vU}\}\} = \{\{\beta_s, \beta_w, \beta_c, \beta_p\}, \{b_s, b_w, b_c, b_p\}, \\ & \{\lambda_s, \lambda_w, \lambda_c, \lambda_p\}, \{\gamma_w, \gamma_c, \gamma_p\}, \{v_s, v_w, v_p\}\} = \{\{\{s_1, s_4, s_5\}, \{w_1, w_6, w_{10}\}, \{c_1, c_2, c_8\}, \\ & \{p_8, p_{11}\}\}, \{\{s_1\}, \{w_{12}\}, \{c_1, c_2\}, \{p_8, p_{11}\}\}, \{\{s_6, s_7, s_9, s_{10}, s_{11}, s_{13}\}, \{c_5, c_6, c_7\}, \{p_{13}\}\}, \\ & \{\{w_4, w_5, w_7, w_8, w_{10}\}, \{c_5, c_6, c_7\}, \{p_5, p_{11}, p_{12}\}\}, \{\{s_8\}, \{w_{11}\}, \{p_3, p_5, p_8\}\}\}. \end{aligned} \quad (13)$$

Функция “ИЛИ” для всех гранул, входящих в $\Gamma_{aU_{ш}}$, определяет набор элементов глобальной гранулы Γ_U , содержащихся в срезах множества U , для изменения соответствующих признаков (функций) PИМ-системы, чтобы реализовать заданный алгоритм с требованиями к параметрам, указанными в табл. 6:

$$\Gamma_{aU} = \{ \{s_1, s_4, s_5, s_6, s_7, s_9, s_{10}, s_{11}, s_{13}\}, \{w_1, w_4, w_5, w_6, w_7, w_8, w_{10}, w_{11}, w_{12}\}, \{c_1, c_2, c_5, c_6, c_7, c_8\}, \{p_3, p_5, p_8, p_{11}, p_{12}, p_{13}\} \}. \quad (14)$$

В частности, согласно первому члену гранулы (14), обращаясь к табл. 2, получим, что для реализации алгоритма с параметрами в табл. 6 необходимы широкая полоса пропускания по каналу процессор-память, применение как серийных, так и специализированных ВП, скоростных коммутаторов внутри чипа и между чипами, а также размещенные на чипе контроллер управления чипом, контроллер для связи с внешней памятью, интерфейс ввода-вывода, БОУП, ППЗУ, компилятор языка высокого уровня. Аналогично можно идентифицировать элементы остальных членов гранулы Γ_U , обращаясь соответственно к табл.3 – табл.5. В итоге мы получим множество признаков $M_{ПР}$ функциональной среды PИМ-системы, необходимых для реализации алгоритма с соответствующими требованиями к набору параметров согласно табл. 6. Далее элементы множества $M_{ПР}$ располагаются в порядке приоритетов (функций принадлежности данному множеству), в соответствии с порядком приоритетов идентичных параметров технического задания (ТЗ). При этом приоритетность каждого параметра либо может быть заложена в ТЗ, либо определяется экспертным путем. Упорядоченные таким образом элементы множества $M_{ПР/ТЗ}$ образуют базовый портрет (БП) функциональной среды PИМ-системы для рассматриваемого алгоритма решаемой задачи и выбранного типа интегральной технологии.

Таким образом, процесс построения модели функциональной среды PИМ-системы, её составных элементов и формирование на основе этой модели базового портрета функциональной среды PИМ-системы для рассматриваемого алгоритма решаемой задачи и выбранного типа интегральной технологии можно представить в виде следующих процедур:

1. Анализ технического задания на создание PИМ-системы и определение приоритетности требований к её параметрам.
2. Анализ алгоритма решаемой задачи и определение требований к набору параметров функциональной среды PИМ-системы со стороны алгоритма.
3. Определение множества признаков Ω , отличающих PИМ-систему от классических КС.
4. Определение типа и состава срезов функционала E .
5. Построение логической модели функциональной среды PИМ-системы в соответствии с (7).
6. Анализ существующих интегральных технологий, пригодных для построения PИМ-систем, и выбор i -ой технологии.

7. Построение таблиц признаков (функций) по всем срезам множества E , отличающих PIM-систему от КС с классической архитектурой, и установление влияния признаков на параметры функциональной среды PIM-системы.

8. Построение сводной таблицы элементов всех срезов множества U .

9. Формирование набора гранул по принципу подобия (идентичности) элементов множества Ω в соответствующих срезах сводной таблицы.

10. Формирование обобщенного портрета $\Gamma_{ОБП}$ функциональной среды PIM-системы.

11. Формирование доминирующего набора гранул $\Gamma_{ДГ}$ и определяющего набора параметров $\Omega_1 \subseteq \Omega$.

12. Формирование параметрического шаблона $\Omega_1 \subseteq \Omega$ алгоритма и выделение из $\Gamma_{ДГ}$ набора гранул $\Gamma_{aUш}$ по шаблону Ω_2 .

Построение глобальной гранулы Γ_U как функции “ИЛИ” всех элементов $\Gamma_{aUш}$.

Идентификация по элементам гранулы Γ_U признаков (функций), описанных в сформированных ранее таблицах соответствующих срезов, и формирование базового портрета (БП) функциональной среды PIM-системы.

Следует подчеркнуть, что приведенные в табл. 2 – 5 признаки и функции и их влияние на параметры компьютерной системы рассматривались для современного уровня интегральной технологии, например, уровня технологии MOSIS (0,18 мкм), которая использовалась при создании чипа DIVA. Мы предположили, что такая технология обеспечила возможность размещения на одном кристалле (чипе) всех компонентов структуры, приведенных на рис. 1. Аналогичные процедуры можно выполнить и для других технологий, например, BiCMOS (0,55мкм), CMOS (0,8мкм) и др. Естественно, при этом, из-за более низкой степени интеграции, следует учитывать более жесткие ограничения как на количество компонентов на кристалле, так и на качественные и количественные признаки, определяющие изменение параметров вследствие особенностей PIM-системы. В результате привязки к интегральной технологии мы получим множество обобщенных портретов $\Gamma_{ОБП}(t_k \in T)$, а также множество обобщенных шаблонов $\Gamma_{aUш}(t_k \in T)$, среди которых, используя шаблон технического задания, построенный аналогично алгоритмическому шаблону, а также экспертные оценки функций принадлежности параметров соответствующим множествам, можно определить множество базовых портретов (БП) функциональной среды и соответственно квазиоптимальную конфигурацию PIM-системы с соответствующими свойствами (признаками), указанными в этих БП. Данная процедура требует отдельного рассмотрения, и её изложение предполагается в последующих публикациях.

4. Выводы

PIM-системы – новый класс устройств, опыт разработки и применения которых мало изучен ввиду отсутствия соответствующей информации, тем более, что отечественные системы такого класса не разрабатывались. Благодаря особенностям их архитектурно-структурной организации, PIM-

системы по сравнению с классическими КС приобретают новые признаки (функции), изменяя тем самым в лучшую сторону основные (с пользовательской точки зрения) параметры, такие, как производительность, большая разрядность обрабатываемых данных, возможность параллельной обработки огромного количества данных (несколько тысяч слов) и др. Однако привнесение новых признаков (функций) сопровождается и привнесением соответствующих затрат, прежде всего затрат на разработку (или модификацию) системного и прикладного программного обеспечения, а также затрат аппаратных ресурсов на поддержку реализации новых функций (например, соответствующих контроллеров, блока обработки управляющего пакета и др.). В количественном отношении изменение параметров оценить практически невозможно, поскольку речь идет об изменении параметров по отношению к КС с классической архитектурой, которые на данных задачах не могут быть апробированы хотя бы потому, что эти задачи на классических КС не поддаются решению. Поэтому речь может идти об изменении параметров в категориях “больше”, “меньше”, “больше или равно” и др., что переводит синтез систем такого класса в теорию нечетких множеств. При этом большое количество вновь обретенных признаков (факторов), влияющих на изменение каждого параметра, определяет необходимость выделения главного и абстрагирования от второстепенного, что приводит к целесообразности применения, помимо теории нечетких множеств, теории гранулирования. В такой интерпретации подход к разработке и созданию PIM-систем авторам не известен.

В данной статье авторы предлагают метод построения математической модели функциональной среды PIM-системы, формирования и оценки компонентов этой модели с использованием теории нечетких множеств и теории гранулирования, что позволяет построить обобщенный портрет и на его основе – базовый портрет функциональной среды PIM-системы для рассматриваемого алгоритма решаемой задачи и выбранного типа интегральной технологии. Полученный таким образом базовый портрет является опорным решением при проектировании PIM-системы, поскольку он дает достаточно полное представление о признаках (функциях), которые должна реализовать проектируемая функциональная среда, а также набор приоритетных параметров, с помощью которых эта среда оценивается.

СПИСОК ЛИТЕРАТУРЫ

1. Архитектурно-структурная организация компьютерных средств класса “Процессор-в-памяти” / А.В. Палагин, Ю.С. Яковлев, Б.М. Тихонов и др. // Математичні машини і системи. – 2005. – № 3. – С. 3 – 6.
2. Палагин А.В. Системы памяти с интеграцией функций хранения и обработки информации (PIM-системы) / А.В. Палагин, Ю.С. Яковлев, Б.М. Тихонов. – К., 2006. – 33 с. – (Препр./НАН Украины. Ин-т кибернетики им. В.М. Глушкова; 2006-3).
3. Яхьяева Г.Э. Нечеткие множества и нейронные сети: Учебное пособие. – М.: Интернет-университет информационных технологий: БИНОМ, Лаборатория знаний, 2006. – 316 с.
4. Колмогоров А.М., Фомин С.В. Элементы теории функций и функционального анализа. – М.: Наука, 1976. – 544 с.
5. Mencar Corrado Theory of Fuzzy Information Granulation: Contributions to Interpretability Issues. – http://www.di.uniba.it/~mencar/download/research/tesi_mencar.pdf.
6. Gomolińska Anna. Rough information granules in social agent system modelling. – <http://logika.uwb.edu.pl/studies/stud9/ag.pdf>.
7. Елисеева Е.В., Яковлев Ю.С. О концепции построения программной среды PIM-систем // УСИМ. – 2008. – № 4. – С. 58 – 67.
8. Paul W. Schermerhorn. A platform for prototyping PIMOS system services. – <http://www.cse.nd.edu/Reports/2002/TR-02-01.ps>.

Стаття надійшла до редакції 23.09.2008