

УДК 629.7.05

А.Р. Дегтярев, Г.В. Медведев

АЛГОРИТМ РАСПРЕДЕЛЕНИЯ ЗАДАЧ В МНОГОПРОЦЕССОРНЫХ КОМПЛЕКСАХ ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ

Дегтярев Алексей Робертович, аспирант Ульяновского государственного технического университета, окончил факультет информационных систем и технологий УлГТУ, инженер ОАО «УКБП». Специализируется в области создания и разработки аппаратуры систем летательных аппаратов и наземной техники. [e-mail: alexmind@rambler.ru].

Медведев Геннадий Викторович, доктор технических наук, профессор. Окончил Белгородский технический институт строительных материалов. Профессор кафедры «Измерительно-вычислительные комплексы» УлГТУ. Область научных интересов – разработка измерительных приборов, оборудования для автоматизации производства приборов, организация НИР и ОКР. Имеет монографии, большое число статей и патентов в области приборостроения. [e-mail: ivk@ulstu.ru].

Аннотация

В статье кратко рассмотрены проблемы распределения функциональных задач в многопроцессорных системах. Показано, что без разработки комплексных алгоритмов распределения невозможно построение перспективных отказоустойчивых систем интегрированной модульной авионики (ИМА). Рассмотрены подходы к решению проблемы декомпозиции функциональных задач, а также специфика данной проблемы применительно к интегрированным комплексам бортового оборудования (КБО). Предложен алгоритм распределения функциональных задач между доступными аппаратными ресурсами многопроцессорных комплексов ИМА по критерию минимума загрузки сети. Для примера взят КБО вертолета, построены соответствующие графы его функциональной и аппаратной структур. Показано распределение приведенных функциональных задач, полученное на основе предложенного алгоритма. Алгоритм может использоваться для построения и исследования крейтов модульной авионики, а также для разработки математических моделей взаимодействия аппаратной и программной частей комплекса ИМА.

Ключевые слова: многопроцессорные вычислительные системы, распределение задач, алгоритм, интегрированная модульная авионика.

TASK DISTRIBUTION ALGORITHM FOR MULTIPROCESSOR SYSTEMS OF IN-TEGRATED MODULAR AVIONICS

Aleksey Robertovich Degtyarev, Post-graduate Student of Ulyanovsk State Technical University; graduated from the Faculty of Information Systems and Technologies of Ulyanovsk State Technical University; an engineer of PJSC Ulyanovsk Instrument Manufacturing Design Bureau; specializes in the field of creation and development of aircraft and land vehicle hardware. e-mail: alexmind@rambler.ru.

Gennadiy Viktorovich Medvedev, Doctor of Engineering, Professor; graduated from Belgorod Technical Institute for Building Materials; Professor at the Department of Measuring Computer Systems at Ulyanovsk State Technical University; specializes in the field of creation and development of measuring instruments and equipment intended for automation of device-manufacturing processes, R & D organizations; an author of monographs and great number of articles and patents in the mentioned field. e-mail: ivk@ulstu.ru.

Abstract

This paper considers briefly functional tasks distribution problems in multiprocessor systems. It's shown that the construction of perspective reliable Integrated Modular Avionics System (IMA) is impossible without development of integrated distribution algorithms. The approaches to the functional tasks decomposition problem solving, and a specific character of this problem with regard to integrated complexes of airborne equipment are examined. The functional tasks distribution algorithm among available IMA multiprocessor hardware resources is proposed in order to provide minimum network loading. Helicopter hardware complex is considered as an example. Respective hardware and functional graphs are built and the given tasks are distributed with presented algorithm. This algorithm can be used for IMA crates building and research, and for IMA hardware-to-software interactions mathematical models development.

Key words: multiprocessor system, task distribution, algorithm, Integrated Modular Avionics.

ВВЕДЕНИЕ

Современный КБО, заменяющий традиционную федеративную архитектуру, представляет собой более сложный многопроцессорный комплекс ИМА с периферийными устройствами и средствами индивидуального математического и программного обеспечения [1].

Поскольку система ИМА работает под управлением операционной системы реального времени, а прикладные функции (самолетовождение, управление общесамолетным оборудованием и др.) реализованы в виде отдельных приложений разного уровня сложности и критичности, разделяющих общие вычислительные и информационные ресурсы [2], то становится актуальной задача разработки математических методов оптимального распределения этих приложений между доступными аппаратными ресурсами. Размещение программных приложений и массивов данных влияет на загрузку процессоров и запоминающих устройств модулей, входящих в состав крейта, поэтому при выборе для комплекса бортовой авионики каналов связи с удовлетворительной пропускной способностью и модулей с достаточной производительностью процессоров и емкостью памяти нужно исходить из некоторого определенного варианта распределения приложений и массивов данных по доступным аппаратным ресурсам. Задача размещения актуальна и в ходе функционирования системы, например при вводе в систему новых прикладных функций и соответствующих им программных приложений. Необходимость в перераспределении возникает также при выходе из строя отдельных модулей в крейте. В этих случаях правильное решение задачи размещения способствует поддержанию работоспособности системы и выполнению ей критичных функций.

Построение комплекса с эффективным распределением обеспечит возможность динамической реконфигурации программных и аппаратных ресурсов, что в свою очередь минимизирует резервирование и обеспечит максимально возможный уровень безопасности и эффективности завершения полета или его дальнейшего выполнения в случае отказов [2].

1 СПЕЦИФИКА ДЕКОМПОЗИЦИИ ЗАДАЧ В СИСТЕМЕ ИМА

В общем случае, функционирование вычислительной системы ИМА характеризуется тремя показателями загрузки: сети передачи данных, процессоров и памяти. Каждый из этих показателей зависит от размещения приложений и массивов данных в системе. Поэтому КБО ИМА может быть оптимизирован либо по одному из показателей загрузки (два других в этом случае при решении задачи вводятся в ограничения), либо проводится многокритериальная оптимизация по всем трем показателям. В данной работе рассматривается частный алгоритм размещения приложений по минимуму критерия загрузки сети передачи данных.

Размещение приложений и массивов данных по минимуму загрузки сети передачи данных позволяет уменьшить

время, затрачиваемое на обмен данными между модулями по каналам связи, а следовательно, время реализации и запаздывания прикладных функций (что является критичным фактором в некоторых приложениях). При этом значительно повышается надежность выполнения функций ввиду уменьшения времени использования и числа обращений к каналам связи. При решении задачи по указанному критерию и при наличии каналов связи заданной пропускной способности снижение загрузки каналов позволяет повысить достоверность передачи данных путем введения избыточности, а также увеличивает надежность выполнения функций.

В [3] предлагается методика декомпозиции вычислительной системы на функциональные элементы, которая заключается в выделении независимых подграфов из многосвязного графа операций обмена и обработки информации. При этом предполагается реализация этих подграфов на отдельных вычислителях. Однако такой подход не учитывает специфику организации вычислительно процесса в операционной системе реального времени (ОСРВ). Поскольку каждая прикладная функция реализована в виде сложного программного приложения, которое не может выполняться одновременно на двух вычислителях, то нет необходимости в выделении независимых частей из множества функций и операторов (такими частями являются сами приложения). Необходимо также принимать во внимание и структуру крейта ИМА, в котором данные могут передаваться как от одного вычислительного модуля (ВМ) к другому, так и посредством общей памяти модуля запоминающего устройства (МЗУ) базы данных. Это зависит, прежде всего, от выбранного типа мультипроцессорной системы (SMP или NUMA). Этот вопрос подробно рассмотрен в работе Э. Таненбаума [4].

2 ПОСТАНОВКА ЗАДАЧИ РАЗМЕЩЕНИЯ ПРИЛОЖЕНИЙ И МАССИВОВ ДАННЫХ

Рассмотрим вычислительную сеть, состоящую из множества V типовых ВМ, входящих в состав крейта ИМА. Пусть для реализации полного набора прикладных функций КБО необходимо выполнять множество P приложений, работающих под управлением операционной системы. В памяти модулей должно быть также размещено множество S массивов, представляющих собой исходные данные и результаты операций обработки.

Заданы размеры σ_s массивов $s \in S$ и приложений σ_p , $p \in P$. В σ_s и σ_p учтен также размер рабочей памяти, необходимый для выполнения приложениями операций обработки информации. Заданы также частота ν_p обращения приложения p к тому или иному массиву данных и объем вычислений ρ_p , необходимый для реализации приложения (например, в FLOPS).

Для решения задачи размещения определим значение следующих булевых переменных:

$$x_{sV} = \begin{cases} 1, & \text{если массив } s \in S \text{ находится} \\ & \text{в памяти модуля } v \in V, \\ 0, & \text{если массив } s \in S \\ & \text{не находится в памяти модуля } v \in V, \end{cases}$$

$$x_{pV} = \begin{cases} 1, & \text{если приложение } p \in P \\ & \text{выполняется на модуле } v \in V, \\ 0, & \text{если приложение } p \in P \\ & \text{не выполняется на модуле } v \in V. \end{cases}$$

При постановке задачи размещения нужно учитывать тип сети передачи данных. Внутрикрейтовые связи, как правило, основаны на каком-либо высокоскоростном интерфейсе. Например, PCI-Express предоставляет соединение типа «точка-точка» посредством коммутатора. Так как скорость обмена между модулями по такому каналу связи порядка нескольких Гбит/с, то можно считать, что они соединены друг с другом парными связями по типу «каждый с каждым».

Загрузка λ_v памяти модуля $v \in V$ определяется размерами массивов данных, приложений и рабочей памяти, необходимой для выполнения операций приложений, размещенных в этом модуле:

$$\lambda_v = \sum_{s \in S} \sigma_s x_{sV} + \sum_{p \in P} \sigma_p x_{pV}. \quad (1)$$

При решении задачи должны быть соблюдены следующие ограничения:

- емкость памяти модуля L_v должна быть достаточной для выполнения приложений и хранения массивов данных, размещенных в нем:

$$\sum_{s \in S} \sigma_s x_{sV} + \sum_{p \in P} \sigma_p x_{pV} \leq L_v, \forall v \in V, \quad (2)$$

- быстродействие процессора R_v каждого модуля должно быть достаточным для выполнения с соответствующей частотой всех приложений, размещенных в модуле:

$$\beta_v \sum_{p \in P} \nu_p \rho_p x_{pV} \leq R_v, \forall v \in V, \quad (3)$$

где β_v – коэффициент запаса на основе экспертных оценок.

Кроме того, каждое приложение должно выполняться на одном ВМ [5]:

$$\sum_{p \in P} x_{pV} = 1, \forall p \in P. \quad (4)$$

Каждый массив должен храниться в памяти хотя бы одного модуля:

$$\sum_{v \in V} x_{sV} \geq 1, \forall s \in S. \quad (5)$$

Выражения (1)–(5) определяют условия решения задачи размещения без учета архитектуры модулей КБО.

Пусть функциональная структура описывается в виде ориентированного графа $G_F = (Y, Z)$, в котором каждому приложению или массиву ставится в соответствие вершина $y \in Y$, а информационной связи между ними – дуга $z \in Z$. Архитектура модуля КБО представляется неориентированным графом $G_S = (V, W)$, где вершины $v \in V$ соответствуют модулям, а ребра $w \in W$ – каналам связи между ними. Тогда проблему размещения приложений и

массивов данных в модулях можно рассматривать как поиск отображения множества вершин графа G_F на множество вершин графа G_S .

3 АЛГОРИТМ РАСПРЕДЕЛЕНИЯ ФУНКЦИОНАЛЬНЫХ ЗАДАЧ ПО КРИТЕРИЮ МИНИМУМА ЗАГРУЗКИ СЕТИ ПЕРЕДАЧИ ДАННЫХ

В качестве примера рассмотрим КБО вертолета. В общем случае он должен обеспечивать выполнение следующих функций:

1. Решение задач навигационного обеспечения;
2. Вертолетовождение по запрограммированному маршруту;
3. Формирование и отображение полетно-навигационной информации (ПНИ);
4. Пилотирование и управление полетом;
5. Круглосуточная обзорная система;
6. Оперативный ручной ввод заданных значений параметров полета;
7. Отсчет и отображение текущего и полетного времени;
8. Ручная и автоматическая настройка радиотехнических систем навигации и посадки;
9. Обеспечение взаимодействия с радиолокаторами управления воздушным движением;
10. Передача информации в обеспечение автоматического зависимого наблюдения;
11. Контроль состояния силовой установки и общевертолетного оборудования;
12. Управление общевертолетным оборудованием;
13. Управление системой электропитания бортового оборудования;
14. Формирование и регистрация массива полетной информации в бортовом регистраторе;
15. Функции комплекса средств связи;
16. Передача радиосигналов бедствия;
17. Полетный автоматический контроль исправности комплекса с отображением результатов контроля.

Функциональный граф G_F данного КБО представлен на рисунке 1. Порядковые номера вершин графа соответствуют порядковым номерам перечисленных выше выполняемых функций. Требуемое быстродействие R_v процессора (которое может быть получено на основе анализа количества подфункций и их сложности) для выполнения соответствующей функции указано рядом с каждой вершиной в виде целого числа от 1 до 10. В дальнейшем будем полагать, что процессор ВМ КБО обладает быстродействием 10. Так как структура крейта ИМА обычно такова, что существуют специализированные модули концентрации сигналов (МКС), МЗУ и модули графической обработки (МГО), и их аппаратная реализация отличается от базовых вычислительных модулей, то определить соответствующие им приложения в какой-либо другой модуль без изменения аппаратной структуры крейта будет невозможно. Поэтому при распределении некоторых прикладных функций необходимо учитывать невозможность их совмещения с другими функциями. Однако, в аварийной ситуации, технически можно распределить некоторые

функции базового ВМ в модули МКС и МЗУ, если соблюдаются ограничения (2) и (3), но такой режим работы здесь не рассматривается.

На первом этапе решения задачи распределения необходимо выбрать функции, требующие реализации на специальном вычислителе или вне крейта ИМА, в рассматриваемом случае это:

6. Оперативный ручной ввод заданных значений параметров полета;

16. Передача сигналов бедствия.

На втором этапе выбираются функции, требующие для своей работы всю вычислительную мощность процессора ВМ:

3. Формирование и отображение ПНИ;

4. Пилотирование и управление полетом;

5. Круглосуточная обзорная система.

Приведенные выше функции (вершины), а также инцидентные им информационные дуги, на время исключаются из функционального графа и не участвуют в дальнейшем распределении. С учетом этого граф принимает вид, представленный на рисунке 2. На рисунке 2 для обозначения дуг используется значение потока данных $v_n \sigma_n$, поскольку фактически по данному каналу связи передается массив данных размером σ_n с частотой обращения к нему v_n .

Для удобства дальнейших рассуждений можно упорядочить потоки данных по возрастанию. В рассматриваемом случае положим, что

$$v_{1,2} \sigma_{1,2} > v_{13,12} \sigma_{13,12} > v_{1,8} \sigma_{1,8} > v_{1,9} \sigma_{1,9} > v_{1,14} \sigma_{1,14} > v_{13,11} \sigma_{13,11} > v_{2,14} \sigma_{2,14} > v_{15,10} \sigma_{15,10} > v_{2,9} \sigma_{2,9} > v_{12,17} \sigma_{12,17} > v_{11,17} \sigma_{11,17} > v_{13,14} \sigma_{13,14} > v_{13,17} \sigma_{13,17} > v_{7,1} \sigma_{7,1} > v_{7,17} \sigma_{7,17}$$

Также будем считать, что объем памяти L_v каждого ВМ достаточен для хранения любого набора функциональных

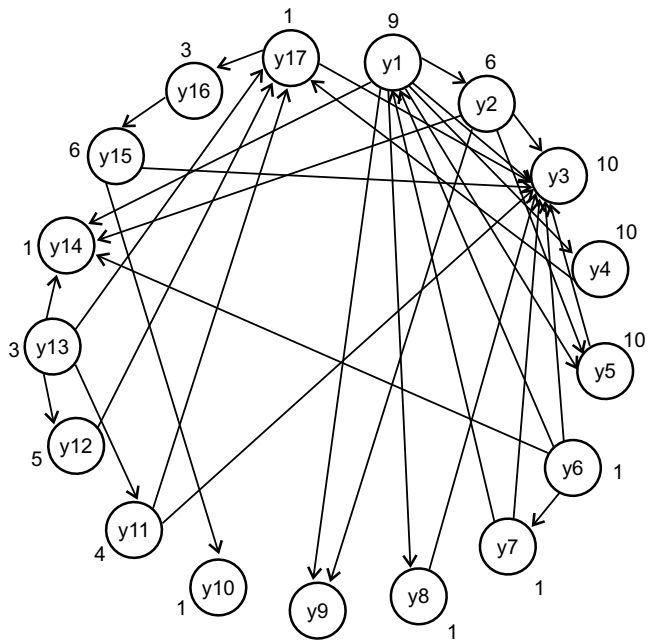


Рис. 1. Функциональный граф вычислительной системы КБО

задач, т. е. условие (1) выполняется всегда.

Алгоритм поиска размещения по критерию минимума загрузки сети передачи данных заключается в последовательном размещении функций по модулям при условии соблюдения ограничений по емкости памяти и быстродействию процессоров.

Шаг 1. Из множества всех вершин $u \in U$ выбирается пара с максимальным значением потока данных $v_n \sigma_n$.

Шаг 2. Проверяется возможность реализации выбранной пары приложений на отдельном ВМ, то есть не нарушаются ли ограничения (2) и (3). Если ограничения не нарушаются, то выбранные вершины отображаются на любую свободную вершину графа G_S . Если же одна из вершин, принадлежащих выбранной паре, уже закреплена за какой-либо вершиной графа G_S , то проверяется возможность отображения второй вершины пары на эту же вершину графа G_S . Если и в том и в другом случае условия (2) и (3) не выполняются, то выбранную пару оставляют неотображенной.

Шаги 1–2 выполняются циклически (при этом каждый раз выбираются две вершины со следующим по порядку меньшим значением потока данных) до тех пор, пока все вершины графа G_F не будут отображены на вершины графа G_S или же пока не останется одна неотображенная вершина, либо две неотображенные вершины, которые нельзя реализовать на одном вычислителе (из-за особенностей построения системы или из-за несоблюдения ограничений (2) и (3)).

Шаг 3. Оставшиеся вершины графа G_F отображают на свободные вершины графа G_S . Конец вычислений.

Таким образом, алгоритм состоит из последовательного перебора пар вершин графа G_F по убыванию значения

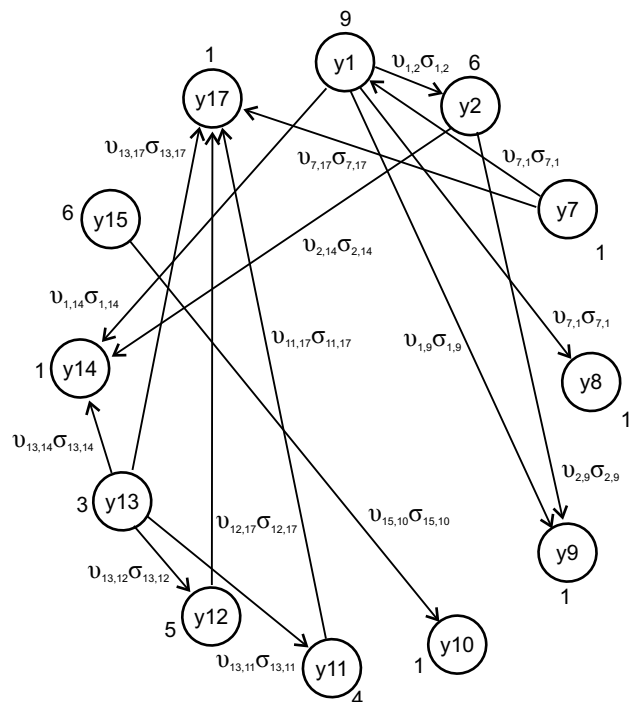


Рис. 2. Функциональный граф подсистемы ИМА

потока данных между ними и отображения указанных пар на вершины графа G_S при соблюдении ограничений на объем памяти и быстродействие процессора.

Архитектура рассматриваемого КБО в виде графа, реализованная на основе крейта ИМА с распределенными функциональными задачами, приведена на рисунке 3. При этом дуга результирующего графа G_S может представлять собой набор дуг исходного графа G_F . Происходит это в том случае, если все дуги этого набора инцидентны одной и той же паре вершин. В этом случае такой набор может быть заменен одной дугой с максимальным для них объемом передаваемого массива и максимальной частотой обращения. Такая организация позволяет снизить нагрузку на коммутатор из-за меньшего количества каналов, а также значительно упростить топологию сети.

На рисунке 3 видно, что количество дуг, соединяющих вершины, сократилось. А значит, сократилось и суммарное время передачи данных по сети и задержки при обработке информации. Вершины s_1-s_5 , s_7 , s_8 представляют собой базовые ВМ, вершина s_6 – МГО, s_9 – МКС.

ЗАКЛЮЧЕНИЕ

Предложенный алгоритм будет более эффективен при проектировании сложных систем авионики, в которых решается большее количество различных задач и используется большее число модулей.

Также следует отметить, что необходима разработка комплексного алгоритма, решающего задачу оптимального распределения задач одновременно по множеству критериев (минимальная нагрузка сети передачи данных, минимальная нагрузка процессоров, минимальная нагрузка памяти и т. д.). При этом возможно разделение многомерной функциональной задачи на отдельные подзадачи, которые требуют обеспечения каких-то частных функционалов оптимальности. То есть какая-то часть системы может быть критична к задержкам передачи данных, а какая-то – к нагрузке процессора.

Сложность современных систем авионики требует разработки принципиально новых методов обеспечения надежности, нежели простое резервирование. Предлагаемые на сегодняшний день многопроцессорные вычислительные системы с динамически перестраиваемой архитектурой, подробно описанные в работах [6–8], не могут быть применены к рассматриваемым системам ИМА, которые требуют именно динамической реконфигурации во время полета. Оптимальное распределение задач по аппаратной платформе может служить основой для разработки эффективных реконфигурируемых систем, сохраняющих свою работоспособность при множественных аппаратных отказах.

СПИСОК ЛИТЕРАТУРЫ

1. Кучерявый А.А. Бортовые информационные системы: Курс лекций / под. ред. В.А. Мишина и Г.И. Ключева. 2-е изд., перераб. и доп. – Ульяновск : УлГТУ, 2004. – 504 с.

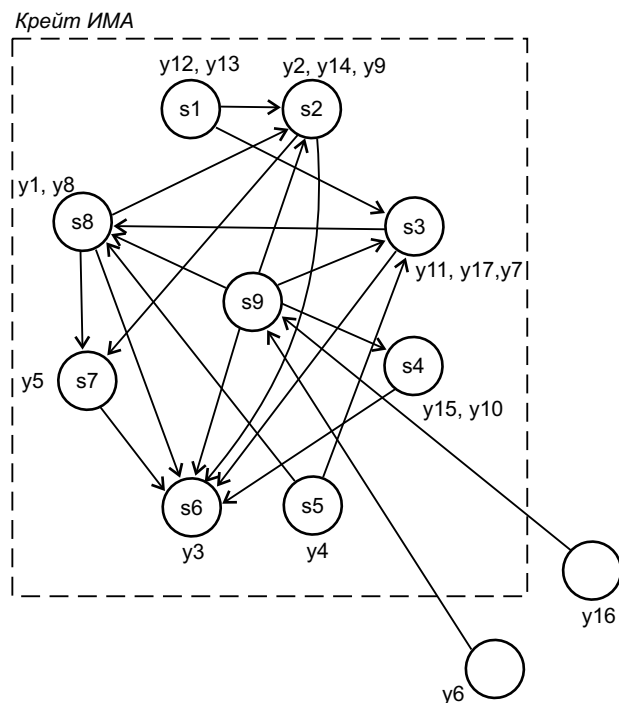


Рис. 3. Граф аппаратной структуры системы ИМА с распределенными функциональными задачами

2. Чуянов Г.А., Косьянчук В.В., Сельвесюк Н.И. Перспективы развития комплексов бортового оборудования на базе интегрированной модульной авионики // Известия ЮФУ. Технические науки. – 2013. – № 3. – С. 55–62.

3. Видин Б.В., Жаринов И.О., Жаринов О.О. Декомпозиционные методы в задачах распределения вычислительных ресурсов многомашинных комплексов бортовой авионики // Информационно-управляющие системы. – 2010. – № 1. – С. 2–5.

4. Таненбаум Э. Современные операционные системы. 3-е изд. – СПб.: Питер, 2010. – 1120 с.: ил.

5. Шенброт И.М., Алиев В.М. Проектирование вычислительных систем распределенных АСУ ТП. – М.: Энергоатомиздат, 1989. – 88 с.

6. Реконфигурируемые мультиконвейерные вычислительные структуры / И.А. Каляев [и др.]. – Ростов-на-Дону : Издательство ЮНЦ РАН, 2008. – 393 с.

7. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.

8. Семейство базовых модулей для построения реконфигурируемых многопроцессорных вычислительных систем со структурно-процедурной организацией вычислений / И.В. Беседин [и др.] // Научный сервис в сети Интернет: технологии распределенных вычислений: матер. всерос. науч. конф. – М.: Издательство Московского университета, 2006. – С. 47–49.

REFERENCES

1. Kucheryavy A.A. *Bortovyye informatsionnyye sistemy: Kurs lektsiy* [Airborne Information Systems: Course].

Under the editorship of V.A. Mishin and G.I. Klyuyev, 2nd revised edition, Ulyanovsk, UISTU Publ., 2004. 504 p. Chuyanov G.A., Kosyanchuk V.V., Selvesyuk N.I. Perspektivy razvitiya kompleksov bortovogo oborudovaniya na baze integrirovannoy modulnoy avioniki [Development Prospects for Airborne Equipment Complexes on the base of Integrated Modular Avionics]. *Izvestiya YuFU. Tekhnicheskiye nauki* [Proceedings of South Federal University. Technical Sciences], 2013, no. 3, pp. 55–62.

2. Vidin B.V., Zharinov I.O., Zharinov O.O. Dekompozitsionnyye metody v zadachakh raspredeleniya vychislitelnykh resursov mnogomashinnykh kompleksov bortovoy avioniki [Decomposition Methods in Resource Allocation Problems of Multiple Computer Avionics Complexes]. *Informatsionno-upravlyayushchiye sistemy* [Information and Control Systems], 2010, no. 1, pp. 2–5.

3. Tanenbaum E. *Sovremennyye operatsionnyye sistemy* [Modern Operating Systems]. St. Petersburg, Piter Publ., 2010. 1120 p.

4. Shenbrot I.M., Aliyev V.M. *Proyektirovaniye vychislitelnykh sistem raspredelennykh ASU TP* [Computer System Engineering of Distributed Computer-Aided Control Systems]. Moscow, Energoatomizdat Publ., 1989. 88 p.

5. Kalyayev I.A. and Others. *Rekonfiguriruyemye multikonveyernyye vychislitelnyye struktury* [Re-configurable Multipipeline Computing Structures]. Rostov-na-Donu, YuNTs RAN Publ., 2008. 393 p.

6. Kalyayev A.V., Levin I.I. *Modulno-narashchivayemye mnogoprotsessornyye sistemy so strukturno-protsedurnoy organizatsiyey vychisleniy* [Module-Extensible Multiprocessors with structure-based procedural computing]. Moscow, Yanus-K Publ., 2003. 380 p.

7. Besedin I.V. and Others. Semeystvo bazovykh moduley dlya postroyeniya rekonfiguriruyemykh mnogoprotsessornykh vychislitelnykh sistem so strukturno-protsedurnoy organizatsiyey vychisleniy [Baseline Module Family for Building the Re-configurable Multiprocessors with Structure-based Procedural Computing]. *Nauchnyy servis v seti Internet: tekhnologii raspredelennykh vychisleniy: materialy vserossiyskoy nauchnoy konferentsii* [Scientific Service in the Internet Network: Distributed Calculation Technologies: Proceedings of the Russian Scientific Conference], Moscow, Izdatelstvo Moskovskogo universiteta Publ., 2006, pp. 47–49.