

УДК 681.325.5: 518.5

ФОРМАЛИЗМ АДРЕСНО – ВРЕМЕННЫХ КАРТ ДЛЯ ОПИСАНИЯ АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ МНОГОКАНАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ

Базовые объекты и операции с АТ–картами

А. М. Астапкович,

канд. техн. наук, начальник СКБ

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В статье описаны базовые объекты и операции формализма АТ-карт. АТ-карта представляет собой двумерное описание (адреса и дискретные временные интервалы) требуемых ресурсов используемого программного модуля. Предложенный формализм ориентирован на использование в программно-инструментальных средствах нового поколения с возможностями генерации кодов для многоканальных систем управления реального времени. Введены новые типы матричных операций, которые дают возможность описания изменений их размера, а также новые правила формирования значений элементов результирующих матриц.

AT-card formalism basic objects and operation are presented in the article. AT-card is a two-dimensional (addresses and discrete time intervals) description for required resources of used software module. Proposed formalism is oriented for a new software development tool with code generation ability for multi channel real time control systems. New type of matrix operations are introduced that give possibility to describe change of matrix size, also as new rules to form a resulting matrix elements.

ВВЕДЕНИЕ

Бурное развитие современной микропроцессорной техники привело к тому, что разработка низкоуровневого программного обеспечения (ПО) становится наиболее инертным звеном в цикле разработки многоканальных систем управления реального времени (СУРВ) встраиваемого класса. Повышение производительности разработчиков ПО и качества низкоуровневого программного обеспечения невозможно без автоматизации задачи синтеза программного обеспечения.

Одним из перспективных направлений развития программно-инструментальных сред нового поколения является использование технологии нисходящего программирования [1, 2] для этого класса СУРВ. Ключевой задачей, определяющей возможность ее применения, является введение адекватного этой технологии формализма, обеспечивающего учет специфики многоканальных СУРВ.

Предлагаемый в работе формализм обеспечивает возможность полуавтоматического или

автоматического синтеза программного обеспечения (ПО) за счет укрупненного описания требуемых ресурсов для каждого из автономных программных модулей. Предлагаемые описания программных модулей и операции с ними призваны обеспечить возможность автоматического анализа критической секции, а также анализа обеспечения корректности межзадачного информационного обмена в синтезируемом коде.

Рассмотренный в данной работе формализм предназначен для укрупненного описания алгоритмов функционирования многоканальных систем управления, низкоуровневое программное обеспечение для которых разрабатывается с применением технологии микрооперационных систем реального времени (mОСРВ).

Использование базовых понятий формализма АТ-карт оказалось продуктивным с чисто практической точки зрения. К настоящему времени уже накоплен опыт использования методики АТ-моделирования для решения ряда плохо формализуемых задач системной интеграции [3].

Статья представляет собой логическое продолжение работы [4] и использует введенную в ней систему базовых понятий и определений.

БАЗОВЫЕ ПОНЯТИЯ ТЕОРИИ тОСРВ

С точки зрения разработчика низкоуровневого ПО для систем управления встраиваемого класса тОСРВ представляют собой стандартную структурную основу, которая автоматически интегрируется вместе с прикладным программным обеспечением (ППО) в используемое системой управления аппаратное обеспечение. Для встраиваемых СУРВ приходится учитывать существование связи между имеющимися ресурсами, количеством параллельных контролируемых процессов и специфицированным временем реакции системы управления на внешнее воздействие. Существенное влияние на эти параметры оказывают и особенности тОСРВ, используемой в конкретной реализации. На практике наличие такой связи приводит к ограничению числа надежно контролируемых процессов в зависимости от сложности алгоритмов управления, имеющихся ресурсов в виде характеристик используемой аппаратной составляющей системы управления, а также дополнительных ограничений, зависящих от конкретных условий эксплуатации системы управления.

Рассматривая тОСРВ для встраиваемых приложений как основу внутриплатформенной (внутрифирменной) технологии разработки программного обеспечения, несмотря на все вышесказанное, необходимо тем не менее обеспечить достаточную универсальность этой управляющей структуры применительно к кругу решаемых задач. В противном случае сложно говорить о технологии разработки, коль скоро каждый вариант ППО требуется разрабатывать с нуля. Однако при этом не следует забывать, что операционная система сама по себе занимает ресурсы, а чрезмерная универсальность приводит к потере эффективности всей разработки. Эти соображения также определяют необходимость выделения некоторого подмножества ОСРВ, включающего в себя специализированные операционные системы для встроенных многоканальных систем управления с фиксированным числом каналов, для обозначения которого и была предложена аббревиатура тОСРВ. Под термином «специализированные» в данном случае подразумевается понятие «оптимизированные» под конкретные условия применения.

В случае сложного характера взаимного влияния каналов реализация алгоритма функционирования всей многоканальной СУРВ подразумевает необходимость использования общих разделяемых ресурсов, а также выполнение временных требований по скорости передачи данных от одной задачи к другой. В простейшем случае под разделяемыми ресурсами подразумевается микропроцессор.

Как явствует из примеров, рассмотренных в обзоре литературы, применение технологии тОСРВ целесообразно при сужении многообразия объектов управления до уровня одного

класса или даже до одного типа устройств. При этом применение технологии тОСРВ к низкоуровневому программному обеспечению служит для придания ему свойств мобильности. Под мобильностью здесь понимается свойство легкой переносимости ППО при модификации используемых аппаратных решений. С этой точки зрения тОСРВ является ядром соответствующей программно-инструментальной среды, обеспечивающей возможность синтеза ППО в загружаемые коды. В свою очередь, соответствующие программно-инструментальные среды должны обеспечить возможность широкого повторного использования библиотечных модулей типовых решений. Вместе с тем, конкретная микропроцессорная платформа или даже семейство используемых микропроцессоров, а также конкретная область применения накладывают ряд ограничений на детали реализации тОСРВ, структуру и организацию библиотек ППО, особенности реализации программно-инструментальных сред. Платой за соблюдение указанных ограничений является повышение производительности программистов и качества разрабатываемого ПО.

Следующим логическим шагом в этом направлении является автоматизация операций синтеза программного обеспечения, что требует развития нового класса программно-инструментальных средств, базирующихся на технологии структурного программирования. Технология структурного программирования характеризуется наличием следующих признаков [1]: проектирование логической структуры алгоритма сверху-вниз с использованием принципа абстракции и пошаговой детализации; модульное программирование, подразумевающее возможность широкого повторного использования отработанных программных единиц, структурное кодирование модулей и сквозной структурный контроль. По аналогии с вышесказанным о традиционной технологии разработки программного обеспечения, такой подход базируется на формулировке и решении широкого круга обратных задач, связанных с определением временных характеристик разрабатываемого алгоритма, ресурсными требованиями для конкретных программных модулей, выбором тактовых частот микропроцессора и т. п.

Теория ОСРВ в ее нынешнем виде оперирует такими базовыми объектами, как «задача», «процесс», «нить». Разные авторы используют эти понятия в разных контекстах [5–10], что зачастую порождает значительные трудности для понимания даже на концептуальном уровне излагаемых разными авторами результатов.

Введем собственные определения ряда ключевых объектов теории ОСРВ применительно к СУРВ встраиваемого класса для исключения двусмысленности толкования данной статьи. В большей степени предлагаемые определения ориентированы на описание алгоритмов, реализуемых средствами низкоуровневого программирования, и являются первичными объектами теории тОСРВ [11, 12].

Задача и составная задача. Будем использовать понятие «задача» в качестве первичного объекта, представляющего собой функционально законченный отрезок кода, реализующий требуемый алгоритм работы конкретных модулей микропроцессора. Программно этот отрезок кода оформляется в виде функции, подпрограммы или макроса.

Несколько задач, выполняемых с заданной последовательностью и информационно обменивающихся между собой, в соответствии с введенным определением также представляют собой задачу. Будем называть объекты такого рода составными задачами.

Базовое отличие задачи от составной задачи заключается в том, что две и более составные задачи могут использовать одну и ту же задачу для собственной реализации. Соответственно, в таком случае эта задача становится разделяемым ресурсом, требующим корректного использования.

Процесс. В общепринятом смысле этого слова процесс подразумевает некоторое действие, разворачивающееся во времени. В соответствии с этим будем считать, что за реализацию соответствующих алгоритмов управления по каждому из параллельных каналов ответственен соответствующий процесс. Сами процессы представляют собой последовательность задач, запускаемых в порядке определенной очередности. Таким образом, процесс – это временная последовательность выполнения задачи, в том числе и составной. В предлагаемой аксиоматике процесс есть сложный объект, формируемый из задач (в том числе из составных), выполняемых в виде, предписанном алгоритмом функционирования системы управления временной последовательности. Количество циклов запуска задач является параметром, характеризующим процесс, и имеет целочисленное значение, лежащее в диапазоне от 1 до ∞ .

Асинхронный процесс. Процесс, выполнение которого может начаться в произвольный момент времени, будем называть асинхронным процессом. Запуск его осуществляется асинхронным входным воздействием на входе системы управления. При этом количество циклов запуска задач варьируется в зависимости от реализуемого алгоритма. В системах управления жесткого реального времени такие входные сигналы, как правило, обрабатываются диспетчером прерываний, которые запускают на выполнение составную задачу выработки выходного управляющего воздействия (воздействий). В соответствии с введенными выше определениями в этом случае асинхронный процесс представляет собой сложную задачу, состоящую из задачи обработки прерывания и задачи выработки выходного воздействия.

Конкурентное пространство и конфликт. В классической теории ОСРВ пересечение адресных множеств разных задач называют критической секцией, а задачу анализа кода на наличие таких пересечений называют задачей анализа критической секции.

При автоматическом размещении кодов программных модулей в памяти программ при неудачном выборе их начальных точек возможно наложение кодов друг на друга. В этом случае можно говорить, что программные модули конкурируют между собой за память программ, которая рассматривается как конкурентное пространство.

Все ресурсы микропроцессора (память программ, память данных, внутренние регистры микропроцессора и т. п.) можно упорядочить некоторым образом за счет соответствующей нумерации, образовав тем самым адресное конкурентное пространство.

Если эту же нумерацию использовать при описании требуемых ресурсов для программных модулей, применяемых при синтезе кода, то это задаст однозначным образом также и адресное пространство каждой из задач. При произвольном варианте размещения задач в памяти микропроцессора возможно их пересечение в адресном конкурентном пространстве.

Следует отметить, что ресурсы задач не совсем однородны. Для двух простых задач, выполняющих только вычисления без взаимного информационного обмена при суммарной размерности используемого адресного пространства, меньшей памяти программ, устранить пересечение можно соответствующим смещением начальных адресов. Если задачи требуют одного и того же периферийного модуля в монопольном режиме, пересечение не устранимо при заданном типе используемого микропроцессора или структуре системы управления. Отметим, что в зависимости от решаемых задач возможны различные способы формирования конкурентного пространства.

Этот подход в работе [4] был распространен и на временную ось. Там же были введены определения А-, Т- и АТ-конфликтов.

Битовые и целочисленные АТ-карты. В работе [4] для описания используемых задач ресурсов (адресов и временных тактов) было введено понятие битовых АТ-карт. По аналогии АТ-карты, элементы которых могут принимать лишь целочисленные значения, будем называть целочисленными АТ-картами.

КЛАССИФИКАЦИЯ ОПЕРАЦИЙ С АТ-КАРТАМИ


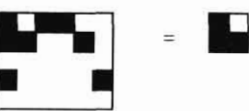
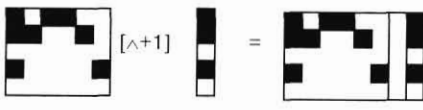
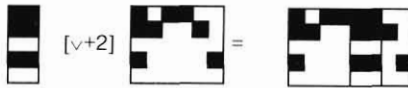
В работах [4, 11] для компактного описания требуемых ресурсов для реализации алгоритма в многоканальной системе управления введен ряд символьных операций с прямоугольными матрицами. С учетом того, что предлагаемый формализм еще не получил широкого распространения, разберем мотивацию и ряд особенностей вводимых символьных операций.

При этом будет использоваться следующая форма записи выражений для базовых операций: [тип операции] операнд (ы).

Иерархия символьных операций с АТ-картами вводится следующими определениями.

Определение 1. Операцию, влияющую на один из размеров матрицы описания АТ-карты,

Определение базовых операций с АТ-картами

Название, краткое описание, тип	Пример	Примечания
Проецирование: $[\perp A] AT = \{a\}$ Одноместная операция ранга (-1) проецирования матрицы на одну из осей: A или T	$[\perp A]$ 	Операция проецирования на определяемое множество $\{m\}$: $[\perp(A \text{ def } \{m\}) AT = \{a\}$. Проецирование возможно как на ось A, так и на ось T
Измерение размера АТ-карты $[DT] AT$, $[DA] AT$ Одноместная операция ранга (-1)	$[DT] AT = j_{\max} - j_{\min}$ $[DA] AT = j_{\max} - j_{\min}$	Реализация зависит от принятого способа представления АТ-карты
Усечение: $[D_1 \square D_3] AT = AT_1$ D_2 D_4 Двуместная операция ранга (0), изменяющая размерность матрицы описания АТ-карты в соответствии с параметрами D_1, D_2, D_3, D_4	$[\square 4]$ 	Параметры усечения – беззнаковые целые числа
Объединение: $AT_2 [\wedge + d] AT_1 = AT_3$ Двуместная операция ранга (0) формирования одной АТ-карты из двух с соответствующим смещением значений элементов суммарной матрицы. Параметры смещений зависят от значений матрицы AT_1 и D		Многоместная операция объединения N АТ-карт $\sum [\wedge + D] AT_i$. Возможно введение векторной операции $\sum [\wedge + D]$ AT_i с переменным смещением для каждого слагаемого
Врезка: $AT_2 [\vee + D] AT_1 = AT_3$ Двуместная операция ранга (0) формирования одной АТ-карты из двух с соответствующим смещением значений элементов суммарной матрицы. Параметры смещений зависят от значений матрицы AT_1, AT_2 и D		Операция ранга (+1) $[W + [D_1 - D_2]]$ множественной врезки с параметрами D_1, D_1+1, \dots, D_2 . Результатом является $(D_2 - D_1 + 1)$ АТ-карт, сформированных операциями врезки с линейно увеличивающимся параметром D. При положительных значениях параметра D операция $[\vee + D]$ становится эквивалентной операции $[\wedge + D]$ и справедливо соотношение AT_1 $[\vee + \text{abs}(D)] AT_2 = AT_2 [\wedge + \text{abs}(D)] AT_1$

будем называть одномерной, а именно А-операцией или Т-операцией. Операции, влияющие на оба размера матрицы описания АТ-карты, будем называть двумерными.

Определение 2. Операцию, производимую над одной АТ-картой, будем называть одноместной, над двумя АТ-картами – двуместной. Операции, которые производятся над большим количеством АТ-карт, будем называть многоместными операциями.

В дальнейшем потребуются и более сложные операции, которые, согласно введенному выше определению, являются двуместными, однако результатом их действия является некоторое множество АТ-карт. Естественно трактовать результат действия такой операции, как увеличение размерности исходного объекта на единицу. Для характеристики этого свойства введем следующее определение.

Определение 3. Изменение размерности результирующего объекта в результате применения операции будем характеризовать целым числом со знаком, которое называется рангом операции.

Описание ряда базовых операций с АТ-картами представлено в таблице.

БАЗОВЫЕ ОПЕРАЦИИ С АТ-КАРТАМИ

Операция проецирования на ось адресов может интерпретироваться как операция преобразования двумерного множества в одномерное упорядоченное множество с монотонно возрастающими значениями элементов и заданным шагом между элементами, равным единице. Это открывает возможность для построения ряда удобных и полезных обобщений. Рассмотрим возможные варианты одноместных операций ранга (-1). Например, определение оси A или T через некоторое определяемое множество элементов $\{m\}$ обеспечивает возможность компактной записи критерия использования (неиспользования) тех или иных адресов или временных интервалов в виде

$$A = \text{def } \{m\}$$

$$[\perp A] AT_1 = \{i\}_1. \quad (1)$$

Если множество $\{i\}_1$ пустое, то это означает, что программная единица, описываемая картой AT_1 , не использует ни одного из адресов, формирующих множество $\{m\}$. Это можно записать в лингвистическом формате современных языков программирования в виде

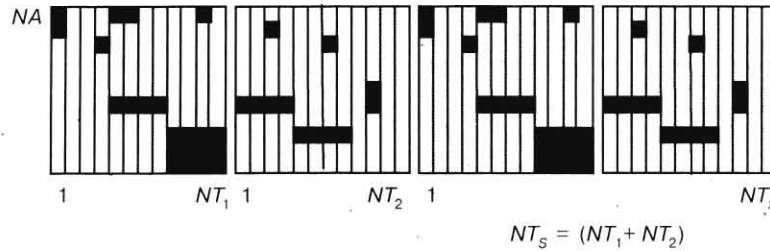
/* Программная реализация

```
.....
CALL   TASK1
CALL   TASK2
.....
```

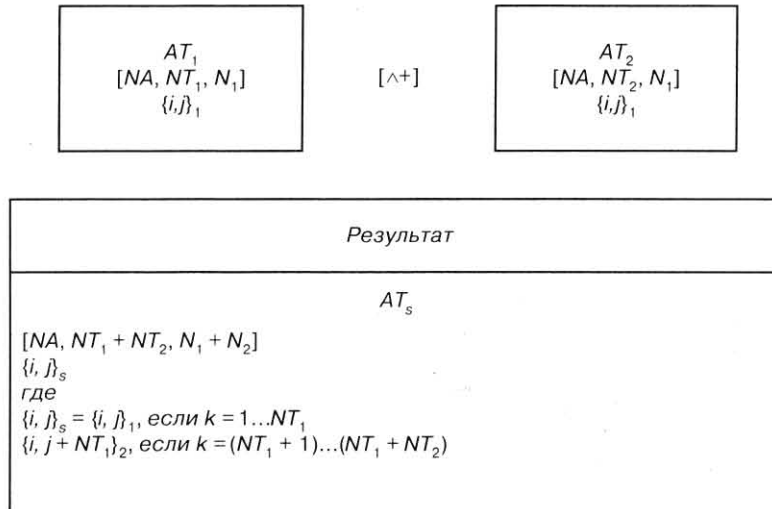
Операция объединения [Λ+]

$$AT_1 [\wedge+] AT_2 = AT_s$$

Графическая иллюстрация



Реализация операции объединения на уровне представлений



■ Рис. 1. Иллюстрация к введению операции объединения [Λ+] двух AT-карт

$$[\perp (A = \text{def } \{m\})] AT_j = \{i\}_1 = 0. \quad (2)$$

Обозначение def в данном случае интерпретируется как определяемое множество $\{m\}$ и подразумевает, что множество адресов определено тем или иным способом. Операция «равно» требует соответствующего определения. Эта же операция может быть применена к оси времени.

Операция усечения. В классическом матричном исчислении введение арифметических операций обеспечивается за счет согласованности размеров матриц. Для получения эффективного аппарата работы с AT-картами требуется обеспечить возможность уменьшать (увеличивать) размеры AT-карт.

В рамках введенной выше классификации определений операция усечения представляет собой одноместную операцию ранга (0). Как следует из приведенной в таблице иллюстрации, результатом операции усечения, выполненной над исходной AT-картой, является AT-карта

меньшего размера, зависящего от конкретного численного значения параметра в используемой операции усечения. Поскольку матрица представляет собой некоторый прямоугольник, то, соответственно, должны быть введены операции левого, правого, нижнего и верхнего усечения. Количество отбрасываемых столбцов (строк) определяется численным значением параметра операции. Последовательность выполнения операций левого, правого, верхнего и нижнего усечения не имеет значения и, соответственно, из них могут быть сконструированы операции усечения по двум, трем и четырем сторонам AT-карты.

Операция объединения [Λ+D]. Антипод операции усечения – двуместные и многоместные операции объединения AT-карт. Эти операции появляются естественным образом при описании суммарной AT-карты, описывающей используемые ресурсы системой псевдопараллельных процессов. На рис. 1 определяется и иллюстрируется двуместная операция объеди-

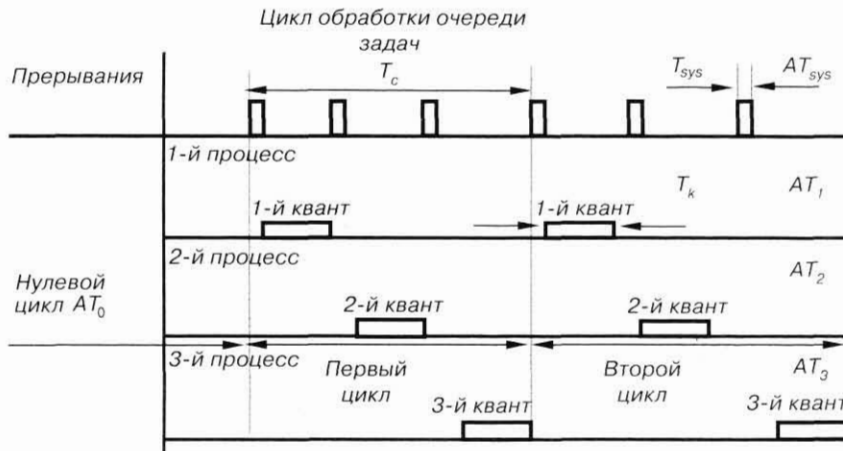


Рис. 2. Процессограмма диспетчера для мОСРВ с квантованием времени

нения двух АТ-карт. Операция объединения не коммутативна.

Для введения многоместной операции объединения рассмотрим программу, циклически опрашивающую N каналов. АТ-карта одного цикла опроса образуется путем применения операции объединения к N АТ-картам AT_j , описывающим требуемые ресурсы для подпрограммы обработки воздействий по каждому из каналов. Общая АТ-карта представляет собой бесконечную карту AT_s . Процесс ее формирования представляется многоместной операцией объединения:

$$AT_s = \sum_{k=1}^{\infty} [\wedge+] \left(\sum_{i=1}^N [\wedge+](AT_i) \right). \quad (3)$$

Внутренний цикл суммирования в формуле (3) представляет собой АТ-карту, описывающую один цикл опроса очереди задач.

Многоместная операция объединения матриц $\Sigma[\wedge+]$ позволяет компактно описать суммарную АТ-карту псевдопараллельных процессов, сформированных из задач, для запуска которых используется диспетчеризация с квантованием времени. На рис. 2 представлена процессограмма, где обработка прерываний от системного таймера, сохранение контекста текущей задачи и загрузка следующей образуют системный процесс длительностью T_{sys} , который также может быть описан своей АТ-картой AT_{sys} . В предлагаемых обозначениях это можно записать в виде

$$AT_s = AT_0 [\wedge+] \sum_{k=1}^{\infty} [\wedge+] \left(\sum_{i=1}^N (AT_{sys} [\wedge+] AT_i) \right), \quad (4)$$

где N следует положить равным трем.

Синтез программного обеспечения для многоканальных систем управления реальным временем требует средств описания конфликтов и введения меры конфликтности конкретной реализации синтезируемого ПО. Применительно к многоканальным СУРВ речь идет либо об АТ-конflikте, либо о Т-конflikте. Это позволяет использовать аппарат линейной или нелинейной оптимизации для логического проектирова-

ния алгоритмов в части корректного разделения ресурсов процессора для обеспечения специфицированных времен реакций на входные воздействия в каналах управления.

Рассмотрим обобщение операции $[\wedge+]$ в виде $[\wedge+D]$. Здесь D может принимать как положительные, так и отрицательные значения:

$$AT_1 = \{i, j\}_1, j = [1-T_1],$$

$$AT_2 = \{i, j\}_2, j = [1-T_2],$$

$$AT_s = \{i, j\}_s \{i, j\}_1, \text{ если } k = 1 \dots T_1,$$

$$\{i, j+D\}_2, \text{ если } k = (T_1 + 1) \dots (T_1 + T_2). \quad (5)$$

Очевидно, что операция $[\wedge+]$ является частным случаем этой операции при $D = 0$. При $D > 0$ происходит увеличение временной компоненты размерности матрицы AT_s , на величину, равную D , а общее количество элементов в матрице AT_s возрастает на $D \times AN$ нулевых элементов, по сравнению с простой суммой элементов матриц AT_1 и AT_2 . При этом в соответствии с определением (5) суммарная АТ-карта при $D > 0$ будет иметь разрыв, описываемый по умолчанию.

Определенная выше двуместная операция $[\wedge+D]$ позволяет уже без всякого труда ввести многоместную операцию $\Sigma[\wedge+D]$. В качестве содержательного примера полезных качеств предлагаемого обобщения рассмотрим возможность описания системного процесса для диспетчеризации с квантованием времени, приведенной на рис. 2. Системный процесс для этого случая описывается в следующем виде:

$$AT_s = AT_0 [\wedge+] \sum_{k=1}^{\infty} [\wedge+] \left(\sum_{m=1}^3 [\wedge+Tc/3](AT_{sys}) \right). \quad (6)$$

Карта AT_0 описывает фазы инициализации микропроцессора и мОСРВ. В соответствии с рис. 2 $D = Tc/3$. Описание системного процесса в виде (6) в сочетании с процессограммой на рис. 2 иллюстрируют возможность компактного и наглядного описания многосвязных бесконечных АТ-карт с помощью операции $\Sigma[\wedge+D]$.

В целях обеспечения возможности описания многосвязных АТ-карт с нерегулярной структу-

рой требуется допустить использование векторного параметра D , что никак не противоречит приведенным выше определениям. Действительно, рассмотренные операции введены через операцию сложения двух АТ-карт. Понятно, что для конкретной реализации потребуются определить размерность вектора D . Описание многосвязной АТ-карты нерегулярной структуры представляется в виде операции сложения с векторным смещением и имеет следующий вид:

$$AT = \sum_{k=1}^{N-1} [\wedge + D_k](AT_k). \quad (7)$$

Выражение (7) представляет собой определение в самом общем виде Т-операции объединения, которую целесообразно называть векторной.

Ограничений на применение предлагаемых операций к адресной компоненте АТ-карт нет. Следует лишь позаботиться об устранении неоднозначности записи для понимания, к какой именно компоненте (адресной или временной) относится выполняемая операция.

Для описания двумерных операций представляется разумным ввести следующее обозначение:

$$AT_2^{A\text{-операция}}, AT_1 = AT_s, \quad (8)$$

которое естественным образом может быть распространено и на многоместные операции.

В качестве примера использования А-операций объединения можно привести задачу описания порядка размещения кодов программных модулей в памяти программ. В этом случае существует аналогия между описанием алгоритма функционирования многоканальной системы реального времени с диспетчеризацией с квантованием по времени с помощью операций объединения и постраничным размещением в памяти программ кодов программных модулей. Двумерные операции обеспечивают возможность описания динамического распределения памяти между процессами.

Для задач этого класса требуется расширить область допустимых значений, включив в нее отрицательную полуось $D < 0$. На рис. 3 показана графическая иллюстрация выполнения операции $[\wedge - 3]T_1$ над адресно-временными картами AT_1 и AT_2 .

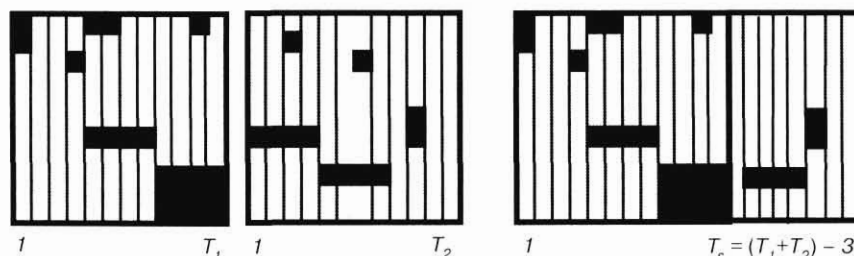
Детальный анализ определения (5) для операции $[\wedge + D]$ приводит к следующему выводу: оно корректно описывает операцию не только при $D > 0$, но и при $D < 0$ и $abs(D) < N_1$. Сложности в непосредственном использовании определения (5) возникают лишь в случае, когда $D < 0$ и $abs(D) > T_1$.

Можно предложить несколько вариантов решения этой проблемы. Самый простой из них заключается в использовании отрицательных значений индексов. В этом случае отсчет идет от первого столбца матрицы AT_1 . Появление отрицательного индекса в матрице AT_s означает смещение на большее количество временных тактов, чем T_1 . При программной реализации это приведет к необходимости использования знаковых целых переменных для описания массивов индексов. При решении прикладных задач анализ знаков выполнения операции объединения со смещением несет информацию об очередности выполнения задач. Так, наличие отрицательных индексов сигнализирует о том, что нарушена очередность запуска задач, т. е. задача, описываемая картой AT_2 , запускается раньше, чем задача, описываемая картой AT_1 . Эта информация позволяет учитывать структуру информационного обмена между задачами и их взаимного влияния.

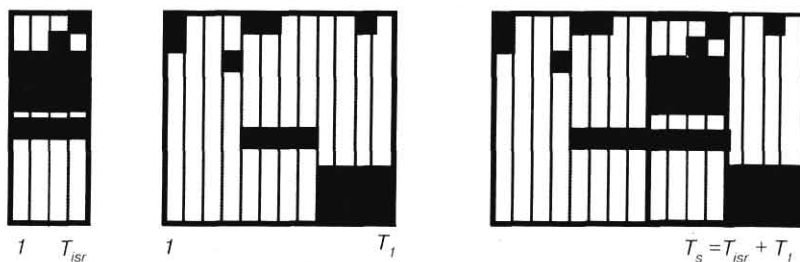
Выполнение операции $[\wedge + D]$ при отрицательных значениях D может привести к появлению А-, Т- или АТ-конфликтов. Подчеркнем, что возможность описания конфликтов в глобальном конкурентном пространстве составляет ту цель, ради которой и вводится предлагаемый формализм.

При таком подходе к синтезу алгоритма задача целочисленной оптимизации относительно некоторого функционала формулируется в пространстве независимых переменных D_k , образуемом из смещений суммируемых AT_k -карт, описывающих требуемые ресурсы для реализации алгоритмов управления по параллельным каналам управления. В качестве простейшей меры для АТ-конфликта может служить максимальное значение элемента матрицы описания суммарной AT_s -карты.

Простейший вариант реализации подразумевает использование целочисленных, например, байтовых карт для представления суммарных карт. При анализе синтезируемого кода на А- или Т-конфликты необходимо использовать введенные операции проецирования суммарной AT_s -карты на соответствующие оси. Максимальное



■ Рис. 3. Результат действия операции $\Sigma[\wedge - 3]T_1$



■ Рис. 4. Результат действия операции $AT_s = AT_{isr} [\vee - 4] AT_1$

значение получаемой целочисленной функции может служить мерой конфликтности.

Целесообразность использования символьных описаний суммарных АТ-карт вытекает из реальных перспектив разработки эффективных алгоритмов для решения широкого круга задач логического проектирования алгоритмов. При этом подразумевается, что для описания ресурсов, требующихся для реализации сложных алгоритмов, будут использоваться комбинированные символьно-списковые представления, состоящие из массивов описаний АТ-карт исходных модулей и набора выполняемых операций при их синтезе в загружаемый код.

Так, в рассмотренном выше примере символьное описание суммарной АТ-карты может быть построено с использованием описания $\{i, j\}$ карты AT_{sys} , к которому добавляется идентификатор операции $\Sigma[\wedge + D]$ с численным значением параметра D и количеством циклов, которое для ряда случаев может быть фактически бесконечным. Несмотря на то, что суммарная АТ-карта также фактически бесконечна, описание карты AT_{sys} будет очень компактно по объему.

Операция врезки $[W + [D_1 - D_2]]$. Для описания прерываний будем использовать двуместную операцию сложения двух АТ-карт, которую назовем операцией врезки и обозначим как $[\vee + D]$. Графическая иллюстрация этой операции приведена на рис. 4. Суммарная АТ-карта при выполнении этой операции формируется за счет врезки соответствующего фрагмента описания обработчика прерываний по указанному месту, определяемому параметром D . Для определенности примем соглашение, что отсчет смещения ведется от максимального значения временного интервала второго слагаемого T_1 , которым в примере на рис. 4 является карта AT_1 . При положительных значениях параметра D операция $\vee + D$ становится по результату эквивалентной операции $\wedge + D$, что может быть выражено в виде соотношения

$$AT_1[\vee + \text{abs}(D)]AT_2 = AT_2[\vee + \text{abs}(D)]AT_1. \quad (9)$$

Примененная в выражении (9) запись смещения с использованием абсолютного значения параметра D обеспечивает наглядность и устраняет причину возможных недоразумений.

Для отрицательных значений параметра D результат выполнения операции $\vee + D$ зависит от соотношения величин D , T_1 и T_2 . Как следует из

иллюстрации (рис. 5), при $\text{abs}(D) \geq (T_1 + T_2)$ справедливо соотношение

$$AT_1[\vee - \text{abs}(D)]AT_2 = AT_2[\wedge + (\text{abs}(D) - (T_1 + T_2))]AT_1. \quad (10)$$

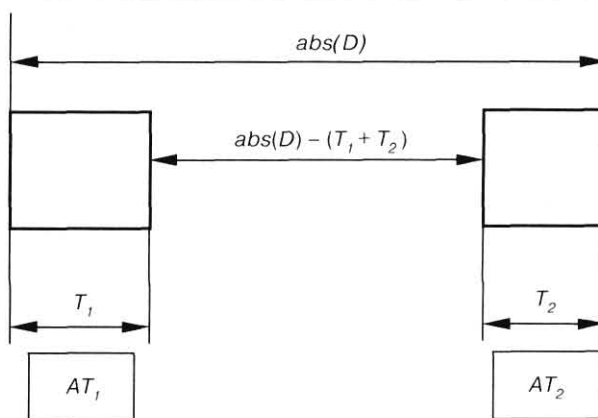
Эту ситуацию естественно назвать инверсией АТ-карт. Для полного определения операции $[\vee + D]$ обратимся к рис. 6, на котором иллюстрируется случай $\text{abs}(D) < T_1$, а само значение D отрицательно.

Первый вариант определения операции $[\vee + D]$ соответствует дисциплине с выбыванием, когда программа с большим приоритетом препятствует началу выполнения менее приоритетной. Это означает, что после появления прерывания программа обработки останавливает выполнение текущей программы, описываемой картой AT_1 . Для таких систем диспетчеризации операция $[\vee + D]$ должна быть определена как

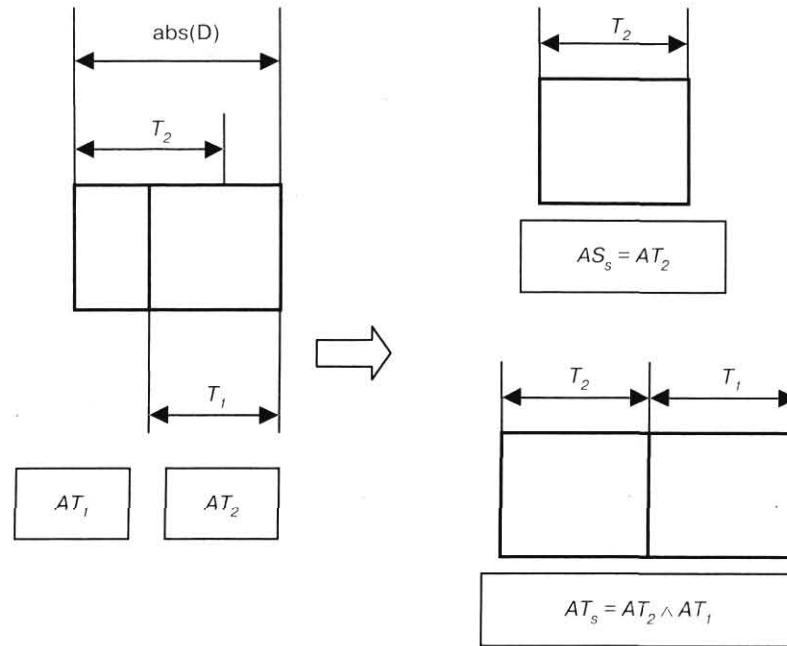
$$AT_1[\vee - \text{abs}(D)]AT_2 = AT_2, \quad \text{abs}(D) < T_1 \quad (11)$$

Другой вариант соответствует ситуации, когда в системе массового обслуживания имеется очередь (буфер). Появление высокоприоритетной заявки в этом случае приведет к тому, что текущая заявка переместится в буфер и после обслуживания заявки с более высоким приоритетом система возобновит ее обслуживание.

Для обеспечения корректного восстановления хода выполнения алгоритма, реализуемого прерванной программой, каждый раз требуется сохранять разное количество параметров, за-



■ Рис. 5. Иллюстрация соотношений параметров операции $\vee + D$ для вывода формулы (10)



■ Рис. 6. К выводу формул (11) и (12)

висящее как от особенностей алгоритма, так и от текущего значения счетчика команд. Набор регистров, обеспечивающих корректное возобновление выполнения прерванной программы, называется контекстом текущей задачи. Для этого варианта операция $[v+D]$ может быть определена как

$$AT_1[v - \text{abs}(D)]AT_2 = AT_2[\wedge]AT_1, \quad \text{abs}(D) < T_1 \quad (12)$$

После введения двуместной операции представляется разумным и совсем необременительным ввести многоместную векторную операцию $\Sigma[v+D_i]$. При этом формирование результирующей AT_s -карты следует вести с карт с большими значениями индекса i .

Двуместная операции $[v+D]$ носит явно выраженный детерминированный характер. На практике при работе с внешними асинхронными процессами момент появления сигнала прерывания не является строго детерминированным, т. е. параметр D определен неоднозначно и необходима специальная операция $W + [D_1 - D_2]$, отражающая специфику обработки прерываний и по результату действия являющаяся многозначной. Результат ее действия на две AT -карты иллюстрирует рис. 7. Операция отражает тот факт, что прерывание может с равной вероятностью произойти в любой момент времени в интервале от $(T_1 - \text{abs}(D_1))$ до $(T_1 - \text{abs}(D_2))$.

При выполнении процессором некоторой программы, описываемой соответствующей AT_1 -картой, будем считать, что она имеет конечный размер по времени, например $[T_1]$ командных циклов. В дальнейшем, не заботясь о формальной строгости, будем говорить, что процессор занят обработкой AT_1 -карты, подразумевая

выполнение соответствующей программы. Появление прерывания означает приостановку обработки текущей AT_1 -карты и начало обработки некоторой новой AT_{isr} -карты с размером $[T_{isr}]$ по t . Поскольку прерывание может появиться в любом командном цикле, то речь должна идти о некотором множестве AT -карт, формируемых из карт AT_1 и AT_{isr} .

Таким образом, операция врезки $[W + (D_1 - D_2)]$ двух AT -карт представляет собой множество из $(D_2 - D_1)$ $\{AT_s\}$ карт:

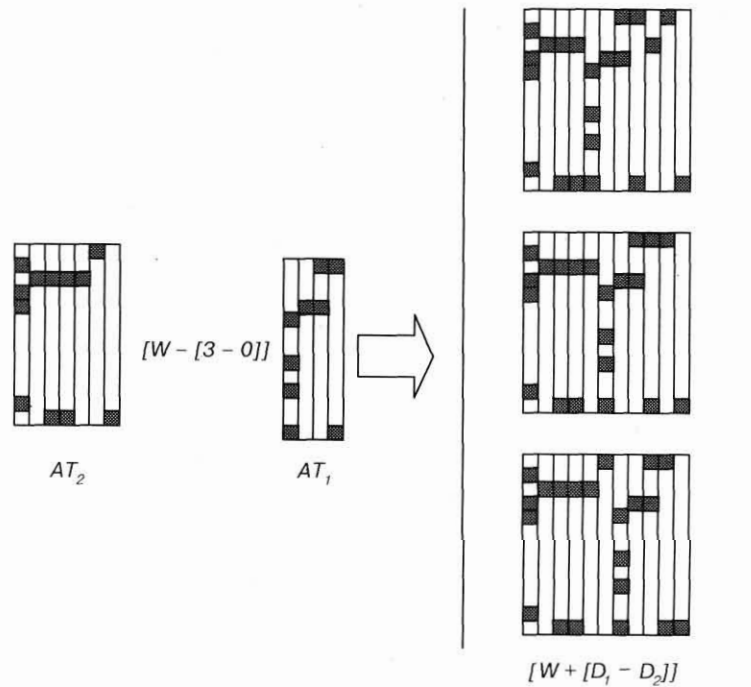
$$AT_1[W + D]AT_2 = \left\{ \begin{array}{l} AT_1[v - \text{abs}(D_1)]AT_2; \\ AT_1[v - \text{abs}(D_1) + 1]AT_2; \\ \dots \\ AT_1[v - \text{abs}(D_2 - D_1)]AT_2; \end{array} \right\} \quad (13)$$

Для упрощения примем, что параметр D_1 должен удовлетворять условию $\text{abs}(D) < T_1$. При этом каждая из AT_s -карт, формируемых в результате выполнения операции, имеет одинаковый размер по t , равный $(T_1 + T_2)$, а общее количество карт равно $\text{abs}(D_2 - D_1)$. Отметим также, что можно ввести многоместную векторную операцию, например в виде $[W + [D_1 - D_i]]$.

Выражение

$$AT_{RR} = AT_0 + \sum_{k=1}^{k=\infty} [\wedge +] \times \left(\sum_{i=1}^{i=N} [\wedge +] (AT_{sys} [\wedge +] AT_i) [W + [D_1 - D_2]] AT_{eISR} \right) \quad (14)$$

иллюстрирует возможность использования предлагаемого формализма для описания тре-



■ Рис. 7. К введению операции $[W + [D_1 - D_2]]$

буемых ресурсов для реализации алгоритма функционирования многоканальной СУРВ с диспетчером RR и одним высокоприоритетным каналом, использующим систему прерывания. Здесь карта AT_{eistr} описывает ресурсы, требуемые для реализации обработчика прерываний внешнего асинхронного сигнала. Параметр D_1 равен значению командного цикла, соответствующего прерыванию от системного таймера, запускающего очередной цикл опроса очереди задач. Параметр D_2 равен числу командных циклов, необходимых для обработки всей очереди задач, плюс число командных циклов для реализации обработчика сигналов внешних прерываний.

Операция $[W + [D_1 - D_2]]$ может привести к необходимости расширения классической теории множеств. Так, при разработке теории сетей Петри понятие «множество» тоже пришлось расширить до понятия «комплект» [13], определяемого как множество с повторяющимися идентичными элементами. Это привело к тому, что для отношения «включение» в теории комплектов была введена функция числа повторений каждого элемента. Таким образом, комплект с максимальным числом повторений, не превышающим единицу, представляет собой обыкновенное множество. Это означает, что множество представляет собой частный случай комплекта.

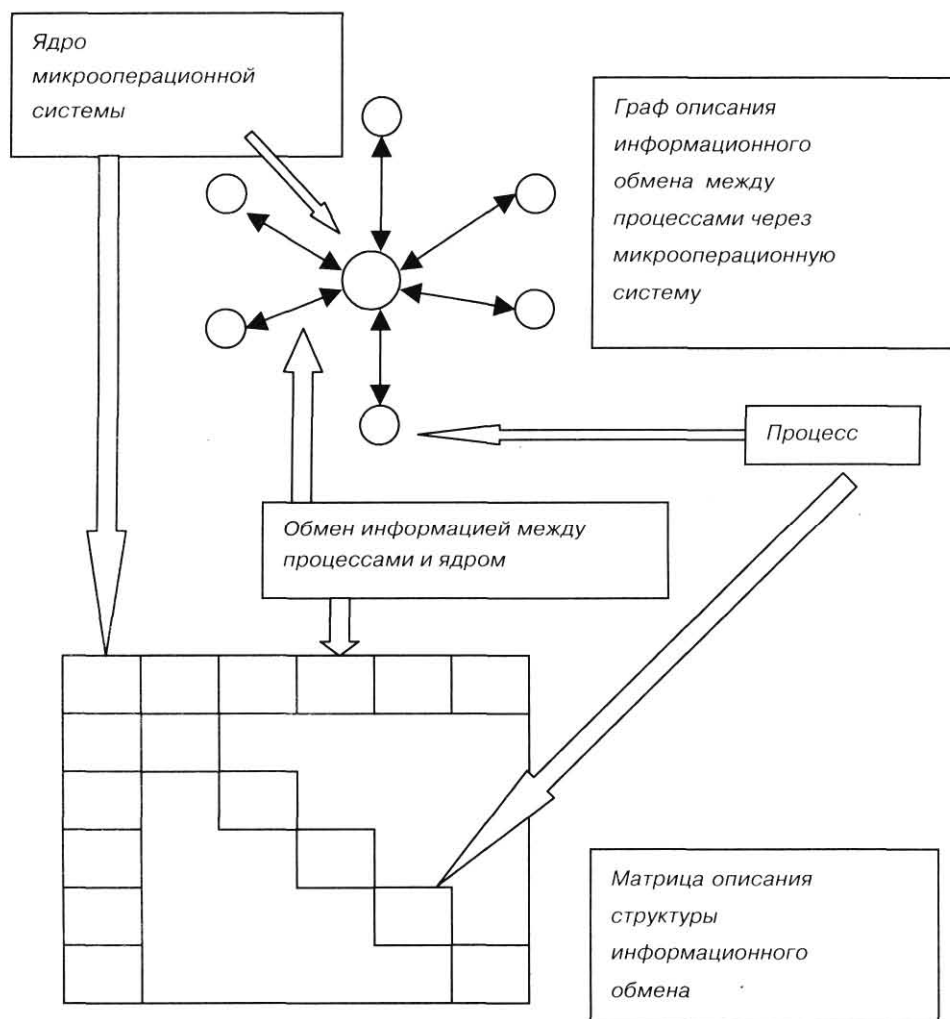
Операция сложения $[W + [D_1 - D_2]]$ может быть описана с применением формализма теории множеств, но в перспективе может оказаться целесообразным введение соответствующего расширения понятия множества (либо понятия комплекта), подразумевающего операцию объединения двух множеств с использованием отношения включения типа: «все элементы множества 1 и один из элементов множества 2».

ОПИСАНИЕ МЕЖЗАДАЧНОГО ИНФОРМАЦИОННОГО ОБМЕНА С ПОМОЩЬЮ АТ-ГРАФОВ

Рассмотрим применение формализма АТ-карт для описания межзадачного информационного обмена с автоматическим анализом возникновения гонок процессов (race conditions) или их взаимной блокировки (dead lock) [6].

Возможность использования аппарата графов и матричного формализма для описания структуры межзадачного информационного обмена иллюстрирует рис. 8. Здесь структура межзадачного информационного обмена соответствует программному обеспечению, разработанному с использованием технологии mОСРВ [11, 12]. При этом информационный обмен между задачами осуществляется средствами ОС и под ее контролем, что и отражает рис. 8. Если рассматривать матрицу описания информационного обмена как блочную, элементы которой – АТ-карты, то описание будет содержать не только информацию о факте наличия информационного обмена, но и данные об используемых адресах, а также временные характеристики. При этом универсализм описания достигается при использовании целочисленных (байтовых) АТ-карт.

Поскольку каждой матрице можно поставить в соответствие граф, структуру которого она описывает, то такие графы можно назвать АТ-графами. В зависимости от решаемой задачи осуществляется выбор базовых объектов, способа их описания и методов их взаимодействия. Использование АТ-графов в сочетании с формализмом сетей Петри [13] существенно расширяет возможности последнего применительно к задачам не только синтеза, но и анализа алгоритмов функционирования многоканальных СУРВ.



■ Рис. 8. Граф и матрица описания структуры межадачного информационного обмена через ядро операционной системы

ЗАКЛЮЧЕНИЕ

Предлагаемый формализм АТ-карт может быть использован в качестве универсальной основы для формулирования и описания широкого спектра задач логического проектирования алгоритмов для СУРВ. С помощью введенных операций суммирования АТ-карт можно описывать сложные алгоритмы диспетчеризации, оценки реализуемости специфицированных параметров системы управления, определения информационного влияния задач друг на друга и т. п. Таким образом, даже на этом уровне абстракции лишь за счет применения АТ-карт имеется возможность решения ряда задач, связанных с оценкой реализуемости алгоритма на

выбранном микропроцессоре либо автоматизации выбора микропроцессора, его тактовой частоты и т. п.

Настоящая работа представляет собой попытку систематизированного изложения нового способа описания параллельных процессов, учитывающих «двумерность» взаимодействующих объектов, описываемых как в адресной, так и во временной областях. Изначально этот аппарат разрабатывался применительно к задачам логического проектирования алгоритмов функционирования многоканальных СУРВ, но может быть использован для широкого круга прикладных задач.

Настоящая работа поддержана грантом Минобразования РФ № Т02-033-3649.

Литература

1. **Назаров С. В.** Операционные системы специализированных вычислительных комплексов. – М.: Машиностроение, 1989. – 400 с.
2. **Зелковец М., Шоу А., Геннон Дж.** Принципы разработки программного обеспечения. – М.: Мир, 1982. – 368 с.
3. **Астапкович А. М., Касаткин А. А.** Исследование возможностей ОС Linux для приложений реального времени с обработкой разнородной информации // Труды четвертой международной школы БИКАМП-03, 23–27 июня 2003 г., Санкт-Петербург. – С. 89–95.
4. **Астапкович А. М.** Формализм адресно-временных карт для описания алгоритмов функционирования многоканальных систем управления. Введение в формализм адресно-временных карт // Информационно-управляющие системы. – 2003. – № 4. – С. 6–14.
5. **Турский В.** Методология программирования. – М.: Мир, 1981 – 272 с.
6. **Tanenbaum A. S.** Operating systems/ Design and implementation. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 077632. – 1987. – P. 337.
7. **Farmer J.** A real-time operating systems for PICmicro microcontrollers // Embedded control handbook. AN585. Microchip Technology Inc., 1997. – N 5. – 142. – P. 5–105
8. **Олифер В. Г., Олифер Н. А.** Сетевые операционные системы. – СПб.: Питер, 2001. – 544 с.
9. **Стивенсон У.** Unix: взаимодействие процессов. – СПб.: Питер, 2002 – 576 с.
10. **Гордеев А. В., Штепин В. А.** Управление процессами в операционных системах реального времени. – Л: ЛИАП, 1988. – 76 с.
11. **Астапкович А. М.** Микрооперационные системы реального времени. – СПб.: Политехника, 2002. – 246 с.
12. **Астапкович А. М.** Микрооперационные системы (мОСРВ) для встроенных систем управления // В кн.: Информационно-управляющие системы и сети. Структуры. Моделирование. Алгоритмы. – СПб.: Политехника, 1999. – С. 196–224.
13. **Основы имитационного и статистического моделирования / Ю. С. Харин и др.** – Минск: Дизайн ПРО, 1997. – 288 с.

ИЗДАТЕЛЬСТВО «ПОЛИТЕХНИКА» ПРЕДСТАВЛЯЕТ

Михеев В. Р., Катышев Г. И.
СИКОРСКИЙ
 ISBN 5-7325-0564-4
 Объем 618 с.
 Формат 70×100¹/₁₆
 Тираж 3000 экз.

Книга посвящена Игорю Ивановичу Сикорскому — великому авиаконструктору XX века, с именем которого связано становление русской авиации и создание первых в мире многомоторных воздушных кораблей «Русский витязь» и «Илья Муромец». В ней большое внимание уделено русскому периоду деятельности Сикорского, его вкладу в становление и развитие всех отраслей авиации России, опередившей по ряду направлений ведущие зарубежные страны, а также рассмотрены основные этапы его жизни и деятельности в Америке, вклад конструктора в создание первых трансконтинентальных авиалайнеров и мирового вертолетостроения. Впервые дана оценка богословского наследия Сикорского, широко освещена его роль в жизни русского зарубежья.

Книга будет интересна специалистам и широкому кругу читателей.

