

Е.В. Полунина,
кандидат технических наук, доцент,
Научно-исследовательский испытатель-
ный центр подготовки космонавтов
имени Ю.А. Гагарина

В.Н. Саев,
кандидат технических наук, доцент,
Научно-исследовательский испыта-
тельный центр подготовки космонав-
тов имени Ю.А. Гагарина

ИСПОЛЬЗОВАНИЕ МОДИФИЦИРОВАННЫХ СЕТЕЙ ПЕТРИ ДЛЯ МОДЕЛИРОВАНИЯ БОРТОВЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ В КОСМИЧЕСКИХ ТРЕНАЖЕРАХ

USING OF MODIFIED PETRI NETS FOR SIMULATION OF ONBOARD COMPUTATION SYSTEMS IN SPACE SIMULATORS

Рассматривается подход к моделированию бортовых вычислительных комплексов в космических тренажерах, при котором для формализованного описания и имитационного моделирования бортовых вычислительных комплексов используются модифицированные сети Петри (И-сети). Возможности использования И-сетей иллюстрируются на примере моделирования дискретного контура управления ориентацией пилотируемого космического аппарата.

The approach to onboard computation systems simulation in space simulators at which for the formalized description and onboard computation systems simulation modelling will be used modified Petri nets (I-nets). The possibilities of using of I-nets are illustrated on the simulation example of discrete control loop by orientation of manned space vehicle.

1. Введение

Современный подход к разработке космических тренажеров состоит в детальном моделировании бортовых систем и полезной нагрузки пилотируемого космического аппарата (ПКА), а также внешней визуальной обстановки, наблюдаемой экипажем в иллюминаторы, оптические и телевизионные средства наблюдения.

К настоящему времени на основе более чем тридцатилетнего опыта построения отечественных космических тренажеров отработаны и совершенствуются технология построения тренажеров, методы и средства математического моделирования логики функционирования и физических процессов, протекающих в бортовых системах, достаточно унифицированы модели движения и модель внутренней среды пилотируемых космических аппаратов [1]. Вместе с тем, основным проблемным моментом при создании тренажеров остается моделирование бортовых вычислительных комплексов, под которыми понимается совокупность вычислительных средств — бортовых цифровых вычислительных машин (БЦВМ), спецвычислителей — и устройств их сопряжения с бортовыми системами. Проблемой является как выбор метода моделирования — полунатурное или математическое, так и реализация этих методов.

Полунатурное моделирование (применение в тренажере штатного БВК, использование штатного программного обеспечения БВК на наземном вычислительном комплексе) имеет преимущества в сроках разработки, быстрой смене версий программного обеспечения, небольших затратах на испытания тренажера, высокой степени адекватности реальному изделию. Однако использование полунатурной модели существенно

ограничивает обучающие свойства тренажера, не позволяя осуществлять приостановку тренировки, оперативный возврат в исходное или любое предшествующее состояние, масштабирование времени. К недостаткам полунатурного моделирования следует отнести также невозможность ввода расчетных нештатных ситуаций, связанных с отказами аппаратуры БВК; увеличение объема и номенклатуры технических средств в составе тренажера; большое время (от 20 до 40 минут) установки начальных условий тренировки и восстановления тренажера при сбоях технических средств и ошибках оператора.

Технический прогресс в области тренажеростроения в целом направлен в сторону перехода от полунатурного моделирования к функциональному (математическому) моделированию бортовых систем летательного аппарата. Но из-за отсутствия на настоящий момент инструментальных средств разработки моделей БВК требуются большие затраты на создание функциональной модели и ее испытания на соответствие штатному изделию, сравнимые с затратами на создание штатного программного обеспечения БВК. С учетом ограниченного времени создания тренажера этот способ в настоящее время используется только при моделировании БВК с небольшим объемом программного обеспечения и достаточно простой организацией вычислений.

Проблема функционального моделирования БВК на тренажере состоит в самом описании объекта моделирования, учете в этом описании не только реализуемых алгоритмов управления, но и организации вычислительного процесса БВК. Требуется найти такой подход, который на едином языке давал бы возможность адекватно описывать структуру БВК (как программные, так и аппаратные части) и процесс его функционирования в системе управления ПКА. Средства разработки моделей БЦВК для тренажеров ПКА должны обеспечивать моделирование в реальном и ускоренном масштабах времени, необходимую для задачи подготовки глубину моделирования, контроль и управление моделью в процессе имитации, регистрацию всех воздействий оператора на модель, минимизировать затраты на доработку модели при изменении структуры БВК и смене версий штатного программного обеспечения.

В настоящее время в различных предметных областях широко используются имитационные модели, основанные на сетях Петри и их расширениях [2,3]. Сеть Петри иерархична, имеет статическую структуру, позволяет описывать и отображать текущее состояние и процессы функционирования моделируемых систем. И хотя существующие модификации сетей Петри в силу их ограниченной мощности моделирования не позволяют моделировать БВК на тренажерах, сам аппарат сетей Петри содержит большие потенциальные возможности для формализованного описания и имитационного моделирования бортовых вычислительных средств на тренажерах ПКА и допускает модификации, направленные на расширение задач моделирования.

Ниже описывается метод создания функциональных моделей БВК в тренажерах с использованием аппарата сетей Петри. Применение данного аппарата для формализованного описания структуры и процесса функционирования программно-аппаратной платформы БВК совместно с выполнением алгоритмов управления потребовало разработки новой модификации сетей Петри, названной И-сетями (имитационные сети).

2. Определение И-сети

В качестве исходного формализма для разработки аппарата описания БЦВК, были выбраны Е-сети [4], ориентированные на решение задачи анализа и оценивания операционных систем на стадии их проектирования или выбора.

Дополнительно в И-сетях введены следующие модификации.

1. Вместо набора базовых типов переходов введен универсальный переход, условия реализации которого и правила приема меток из входных позиций могут быть

любыми функциями статусов входных и выходных позиций перехода, а правила выдачи меток в выходные позиции перехода — функциями статусов входных и выходных позиций и атрибутов принятых меток.

2. Переход может содержать процедурную часть — вычислительный алгоритм, выполняемый при срабатывании перехода. Введение в переход вычислительной части вызвано необходимостью выполнения алгоритмов рабочих программ в процессе имитации функционирования БЦВК.

3. Метка в качестве своего атрибута может содержать не только переменные, но и алгоритмы, выполнением которых управляют переходы, принявшие метку, что позволяет моделировать выполнение рабочих программ БЦВМ.

4. Структура метки не фиксированная, а произвольная, т.е. количество атрибутов и их порядок следования в метке — произвольные в рамках одной сети. В общем случае переход может принять метку одной структуры, а сформировать и передать метку в выходную позицию другой структуры.

5. В сетевой модели БЦВК можно выделить позиции, в которых постоянно находятся метки (например, списки временных заданий, значения некоторых регистров и т.д.). Для их моделирования введены информационные позиции (графически изображаются наполовину заштрихованными кружками). При приеме метки из такой позиции ее статус не обнуляется, метка сохраняется. Статус информационной позиции обнуляется только при записи в нее нулевой информации.

6. В И-сетях из каждой позиции выходит и в каждую позицию входит не более одной дуги. Наличие у позиции нескольких входных или выходных дуг может приводить к конфликтным ситуациям. Введенное ограничение на количество дуг, инцидентных позиции сети, возлагает на пользователя разрешение возможных конфликтов в сети удобным ему способом.

Формально И-сеть определяется как набор:

$$N = (Q, QI, P, I, O, M).$$

Здесь $Q = \{q_i\}$ — конечное счетное непустое множество позиций, $i = 1, l$;

$QI \subseteq Q$ — подмножество, быть может, пустое, информационных позиций;

$P = \{p_i\}$ — конечное непустое множество переходов, $i = \overline{1, n}$;

$p_i = \{L_i, PM_i, PROC_i\}$, где

L_i — логическая функция статусов входных-выходных позиций перехода;

PM_i — процедура приема меток из входных состояний переходов;

$PROC_i$ — процедура определения временного параметра перехода p_i , преобразования входных меток, формирования выходных меток и выдачи их в выходные позиции перехода;

$I \subseteq Q \times P \Rightarrow \{0, 1\}$;

$O \subseteq P \times Q \Rightarrow \{0, 1\}$;

$M = \{mq_1, mq_2, \dots, mq_l\}$ — начальная разметка, где:

$mq_i = \{a_j\}$ — конечное счетное множество атрибутов метки, при условии, что:

1) $\forall q \in Q, I(q) = \{p \in P \mid I(q, p) = 1\} = \{0, p_i\}, O(q) = \{p \in P \mid O(p, q) = 1\} = \{0, p_j\}$;

2) $I(q) \cup O(q) \neq \emptyset$;

3) $\forall p \in P, \{q \in Q \mid I(q, p) = 1\} \neq \emptyset, \{q \in Q \mid O(p, q) = 1\} \neq \emptyset$.

Условие 1 в определении сети означает, что в каждую позицию входит и из каждой позиции выходит не более одной дуги. Условие 2 показывает, что каждая позиция имеет, по крайней мере, либо входную, либо выходную дугу. Условие 3 означает, что каждый переход имеет, по крайней мере, одну входную и одну выходную позицию.

В графической форме имитационная сеть изображается как ориентированный граф со множеством вершин $Q \cup P$. Из вершины-перехода p в вершину-позицию q ве-

дет дуга тогда и только тогда, когда $O(p,q)=1$; из вершины-позиции q в вершину-переход p ведет дуга тогда и только тогда, когда $I(q,p)=1$. В первом случае позиция q называется выходной позицией перехода p , во втором — входной позицией. Позиции, которые имеют только входную или выходную дугу, называются периферийными позициями сети.

Вершины позиции изображаются кружками, информационные — наполовину заштрихованными. Вершины-переходы с временными параметрами, априори равными нулю, изображаются барьерами, остальные — прямоугольниками. Метки изображаются точками в вершинах-позициях.

Набор переходов и позиций вместе с направленными дугами описывают топологию сети. Сеть функционирует, переходя от одной разметки к другой. Метки двигаются по сети путем, так называемой, реализации переходов. Переход p_i реализуется при разметке M тогда и только тогда, когда $L_i(M)=1$. Действие перехода состоит из трех фаз.

Фаза наступления. В этой фазе принимаются метки из входных позиций перехода в соответствии с процедурой приема меток PM_i . При этом статусы простых, не информационных позиций обнуляются, метки в них уничтожаются. Информационные позиции сохраняют свои метки. Выполняется процедура перехода $PROC_i$, в которой определяется временной параметр перехода, выполняются действия над атрибутами принятых меток, формируются выходные метки. Временной параметр может быть константой, атрибутом принятой метки, а также функцией атрибутов принятых меток и модельного времени.

Фаза активности. Продолжается временной промежуток, равный временному параметру перехода. В этой фазе изменений в сети не происходит.

Фаза завершения. Сформированные на первой фазе при выполнении $PROC_i$ метки передаются в выходные позиции перехода, изменяются статусы выходных позиций. Статусам простых позиций, в которые передаются метки, присваиваются единичные значения, статусам информационных позиций могут присваиваться как единичные (в позицию поступает метка), так и нулевые (метка в позиции обнуляется) значения.

Такой принцип изменения разметки сети позволяет рассматривать переход с временным параметром, отличным от нуля, в терминах дискретного моделирования не как событие, а как некоторую активность. Фаза наступления показывает, что началась некоторая активность, т.е. минимальная порция работы, которая не может быть прервана. Изменения, вызванные данной активностью, имеют место только с наступлением фазы завершения. Фазы наступления и завершения не обладают длительностью и в терминах дискретного моделирования соответствуют событиям — изменениям состояния модели.

Введение в процедуру перехода вычислительной части вызвано необходимостью моделирования выполнения алгоритмов управления, реализуемых БЦВК. Но данная модификация позволяет также описывать и моделировать И-сетями не только дискретные системы, на что направлены классические сети Петри и известные модификации сетей Петри, но и сложные системы, включающие в себя как дискретные, так и непрерывные части.

3. Пример

Проиллюстрируем возможности И-сетевого моделирования на примере дискретного контура управления ориентацией пилотируемого космического аппарата. Будем рассматривать режим выполнения ориентации ПКА в системе координат, фиксируемой на момент включения режима. На рис. 1 приведены используемые базисы и функциональная схема режима управления.

Опорный инерциальный базис N является тем базисом, в котором задаются положения всех остальных базисов, используемых в алгоритме управления. Опорный базис совмещается с базисом S , связанным с системой координат ПКА на начальный мо-

мент времени ($N = S_0$). Выполнение ориентации осуществляется в системе координат, задаваемой базисом P .

Контур управления ориентацией включает в себя систему исполнительных органов (СИО), объект управления — космический аппарат в угловом движении (КА) и кинематические соотношения углового движения (КИП), блок датчиков угловых скоростей космического аппарата (БДУС), блок формирования результирующего закона управления (БФРЗУ). В дискретном контуре управления ориентацией выполнение блоков КИП и БФРЗУ возлагается на БЦВМ.

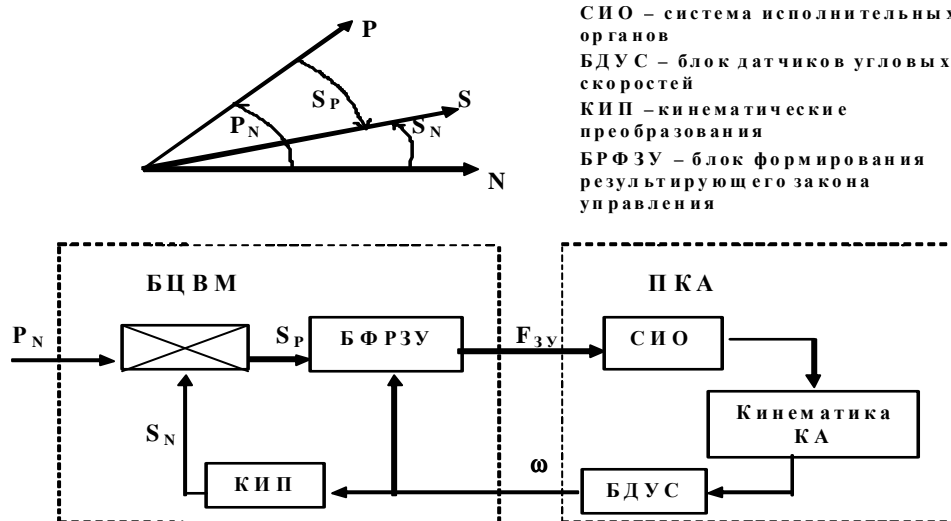


Рис. 1. Функциональная схема дискретного контура управления ориентацией

При реализации режима ориентации на каждом такте работы БЦВМ на входной регистр БЦВМ через устройство согласования поступают показания с датчиков угловых скоростей. Выполняется программа режима ориентации, в которой

а) определяется кватернион рассогласования S_P между заданным и связанным базисами исходя из кинематических соотношений:

$$P_N * S_P = S_N, \quad 2\dot{S}_N = S_N * \omega,$$

где P_N, S_N — кватернионы, задающие положение базисов P и S в базисе N ;

S_P — кватернион, задающий положение базиса S в базисе P ;

\dot{S}_N — производная кватерниона S_N ;

ω — угловая скорость движения ПКА в базисе S ;

N — опорный инерциальный базис;

б) рассчитываются управляющие воздействия $F_{ЗУ}$, как функции кватерниона рассогласования и угловой скорости:

$$F_{ЗУ} = f(S_P, \omega).$$

Управляющие воздействия передаются в выходной регистр БЦВМ, с которого через устройство согласования поступают на систему исполнительных органов.

БЦВМ с целью эффективного использования включаются в несколько контуров управления или в одном и том же контуре реализует несколько алгоритмов управления. Кроме того, на БЦВМ решаются, так называемые фоновые, не связанные с управлением движением программы. В отличие от программ, реализующих алгоритмы управления ориентацией, которые выполняются в каждом цикле работы БЦВМ, фоновые программы выполняются однократно в свободное от задач управления процессорное время.

Рабочие программы полета — программные реализации алгоритмов управления и фоновые программы — хранятся в памяти БЦВМ и выполняются в зависимости от управляющей информации либо поступающей извне, либо заложенной в памяти БЦВМ в качестве полетного задания. Организация вычислительного процесса возлагается на операционную систему (ОС), представляющую собой взаимодействие программных и аппаратных средств. Основные функции ОС — обработка прерываний, организация выполнения рабочих программ в нужной последовательности, обеспечение выполнения рабочих программ (выделение требуемого объема памяти, организация доступа к периферийным устройствам и т.д.). Операционная система работает с системными данными (списками, очередями, регистрами прерываний и пр.), называемыми средой ОС. Замыкание путей ОС, т.е. последовательность выполнения модулей ОС, зависит от данных ОС и свойств рабочих программ, управляемых ОС.

При сетевом моделировании БЦВМ модули ОС представляются вершинами-переходами сети, данные — вершинами-позициями, алгоритмы управления — метками, содержащими в качестве атрибутов имя и характеристики программы алгоритма, используемые операционной системой для ее управления и подлежащие моделированию. Состояние БЦВМ, под которым понимается состояние среды ОС и состояние выполнения рабочих программ, описывается разметкой сети.

Для простоты в рассматриваемом примере БЦВМ моделируется не полностью, а только в той части, которая обеспечивает выполнение режима ориентации. Рассматриваются только два источника прерывания: от таймера и от пульта оператора, с которого задается режим управления и уставочная информация в виде значений кватерниона разворота. Не рассматривается механизм формирования очереди фоновых программ. Не моделируются устройства сопряжения БЦВМ с БДУС и СИО. Программа режима состоит из одного программного модуля АЛГОРИТМ.

И-сетевая модель контура управления ориентацией представлена на рис. 2. Подсеть, включающая в себя переходы $p1, p2, p3, p4$ вместе с входными и выходными позициями, моделирует работу БЦВМ. Переход $p5$ моделирует СИО, угловое движение ПКА и датчики БДУС. Программа, реализующая алгоритм управления, и фоновая программа представляются метками, движущимися по сети, со следующими атрибутами: *PRG* — имя программы, *T* — время выполнения программы, *ПР* — признак разрешения прерывания программы во время выполнения (*ПР* = “да” для фоновых программ и *ПР* = “нет” для программы режима управления).

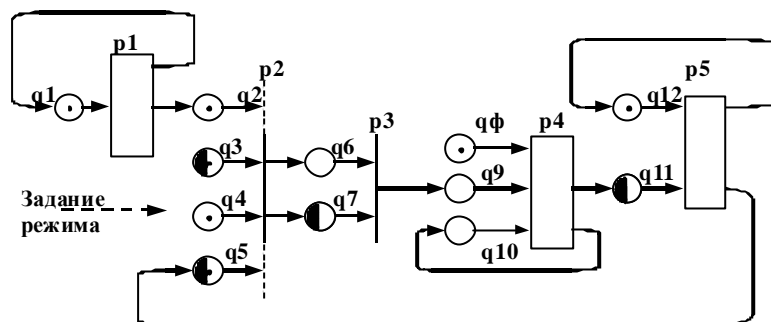


Рис. 2. И-сетевая модель дискретного контура управления ориентацией:

$T=0$ $mq1 = mq2 = 1$ $mq3 = PRG(АЛГОРИТМ)$ $T(6)$ $ПР(нет)$
 $mq4 = режим(задан)$ $l0(0,87)$ $l1(0)$ $l2(0,5)$ $l3(0)$
 $mq5 = mq12 = om1(0)$ $om2(0)$ $om3(0)$ $mqf = PRG(ФОН)$ $T(32)$ $ПР(да)$

Переход $p1$ моделирует работу таймера. Он наступает, если в позиции $q1$ есть метка, принимает метку из $q1$ и через τ_{p1} (время цикла работы БЦВМ) выдает метки в

выходные позиции перехода – $q1$ и $q2$. Для того чтобы запустить таймер, в позицию $q1$ достаточно поместить метку.

Переход $p2$ моделирует обработку прерываний. Он наступает, если хотя бы в одной из позиций $q2$, $q4$ есть метка. Если метка есть в $q4$, то переход принимает метки $mq3$, $mq4$. Если метка $mq4$ содержит уставки на разворот, то уставки передаются в глобальную область памяти, а метка, содержащая программу режима, из позиции $q3$ передается в позицию $q7$, моделирующую список циклически выполняемых программ. Если же метка есть в $q2$, но нет в $q4$, то принимаются метки $mq2$, $mq5$. Метка из позиции $q2$ передается в позицию $q6$, атрибуты метки $mq5$, моделирующие показания датчиков БДУС, передаются в глобальную область памяти, с которой работает программа режима АЛГОРИТМ.

Переход $p3$ наступает, если в позиции $q6$ есть метка. Принимает метки из позиций $q6$, $q7$. Метку $mq7$ передает в позицию $q9$ — очередь программ на выполнение к процессору.

Переход $p4$ моделирует выполнение рабочих программ. Переход наступает, если хотя бы в одной из входных позиций есть метка. Принимает одну из меток по приоритету: $mq9$, $mq10$, $mqф$. При этом, если атрибут принятой метки $ПР = \text{”да”}$, т.е. принятая программа допускает прерывание, временному параметру перехода присваивается единичное значение, значение атрибута T принятой метки уменьшается на единицу и принятая метка с измененным значением атрибута T пересылается в позицию $q10$. Так продолжается до тех пор, пока значение атрибута T не станет равным единице. Как только $T=1$, выполняется программа, указанная в атрибуте PRG принятой метки. Если же значение атрибута принятой метки $ПР = \text{”нет”}$ (программа не допускает прерывания при выполнении), временному параметру перехода присваивается значение атрибута T принятой метки, выполняется программа, указанная в атрибуте PRG , рассчитанные в программе управляющие воздействия передаются в выходную позицию $q11$ в качестве атрибутов метки $mq11$.

Переход $p5$ представляет собой модель углового движения космического аппарата в виде системы дифференциальных уравнений. Действие перехода заключается в одном шаге интегрирования системы дифференциальных уравнений. Чтобы инициировать переход, достаточно в позицию $q12$ поместить метку, содержащую в качестве атрибутов начальные условия интегрирования. Далее переход выполняется циклически с шагом, равным шагу интегрирования. Переход $p5$ принимает метки из $q12$ и $q11$, выполняет один шаг интегрирования системы дифференциальных уравнений, результат интегрирования — угловые скорости КА — передает в позиции $q12$, $q5$.

Как видно из приведенного примера, сетевая модель близка к реальной системе. В ней отражены как структура и функциональные связи моделируемой системы — топология сети, так и динамика системы — движение меток по сети и изменение их атрибутов в процессе движения.

В реальной БЦВМ имеется два уровня программ — программы операционной системы и рабочие программы полета. Программы первого уровня управляют выполнением программ второго уровня. В И-сетевой модели также присутствуют два уровня программ — процедуры переходов и программы, являющиеся атрибутами меток. Процедуры переходов управляют движением меток по сети и выполнением программ, содержащихся в метках.

Работа БЦВМ и движение ПКА происходит параллельно во времени, что также отражено в И-сетевой модели: переходы $p1$ (модель таймера БЦВМ) и $p5$ (модели СИО, БДУС и углового движения ПКА) инициируются независимо друг от друга метками в позициях $q1$ и $q12$ соответственно и протекают параллельно в модельном времени.

Модели ПКА и БЦВМ связаны информационно: в БЦВМ поступают показания с датчиков БДУС, а из БЦВМ в систему исполнительных органов выдаются управляющие воздействия. Эту связь в модели отображают позиции $q5$ и $q11$. Позиция $q5$ моделирует регистр, на который поступают показания с датчиков, а метка в позиции $q5$ мо-

делирует содержимое регистра. Позиция q_{11} и метка в позиции моделируют регистр, с которого в СИО поступают управляющее воздействие и его содержимое.

Состояние модели в любой момент времени определяется разметкой сети. Отсюда вытекает возможность контроля функционирования модели и управления процессом функционирования модели путем изменения текущей разметки. Отображая граф сети и текущую разметку, можно непосредственно наблюдать функционирование модели во времени.

Благодаря высокому уровню описания процесса функционирования БВК и возможности преобразования И-сетевых моделей на уровне схем, применение данного подхода в тренажере позволяет сократить сроки разработки, испытаний и модификации моделей БВК при изменении структуры и смене версий штатного программного обеспечения, а присущие И-сетевой модели свойства наглядности, контролируемости и управляемости — обеспечить эффективную подготовку экипажей.

ЛИТЕРАТУРА

1. Тренажерные комплексы и тренажеры / Шукшунов В.Е. [и др.]. — М.: Машиностроение, 2005. — 383с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем. — М.: Мир, 1984.
3. Мурата Т. Сети Петри: Свойства, анализ, приложения (обзор) //ТИИЭР. — 1989. — №4.
4. Noe J.D., Nutt G.J., Macro E-nets for representation of parallel systems // IEEE Trans. Comput. — 1973. — Vol. C—22. — N8.