

В. П. Кулагин, д-р техн. наук, проф., зав. кафедрой "Аппаратное, программное и математическое обеспечение вычислительных систем", e-mail: Kulagin_VP@mgupl.ru,
Е. С. Малых, магистрант кафедры "Аппаратное, программное и математическое обеспечение вычислительных систем", e-mail: kattya.miller.95@yandex.ru,
МИРЭА — Российский технологический университет, Москва

Проектирование матричных вычислительных структур с использованием сетей Петри

На примере проектирования матричных вычислительных структур предложен алгоритм синтеза моделирующих сетей Петри. Описан подход к синтезу новых структур на основе использования векторов проекции. Показано, что данный подход не позволяет получить все множество возможных сетей Петри (СП), моделирующих выполнение заданной на матричной структуре функции. Описан оригинальный алгоритм проектирования полного множества возможных СП-моделей матричных структур для заданной функции. Алгоритм основан на использовании структурного анализа и тензорной методологии. Результатом работы алгоритма является генерация программ синтеза структур СП-моделей. Показано, что среди генерируемых программ существуют такие, которые синтезируют эквивалентные СП-модели. Сформулированы свойства, выявляющие "эквивалентные" программы. В целях количественного и качественного анализа построенных матричных структур введены критерии оценки структурных и динамических свойств синтезируемых СП-моделей. Показано, что в силу экспоненциального роста числа синтезируемых СП-моделей от размерности исходной примитивной системы необходим поиск новых приемов и подходов, которые исключали бы из рассмотрения модели, не отвечающие наперед заданным требованиям.

Ключевые слова: матричные структуры, параллельные вычисления, сети Петри, тензорный анализ, программы синтеза, оценка структурных свойств

Введение

Актуальность построения параллельных структур, хорошо приспособленных для реализации вычислительно емких алгоритмов обработки сигналов и изображений, в настоящее время не вызывает сомнения. Подобные структуры обладают такими достоинствами, как модульность, регулярность, наличие локальных соединений и высокая степень конвейерной обработки. Использование для построения указанных структур систолических массивов не всегда приводит к успеху из-за недостатка систолических массивов, вызванного глобальной синхронизацией. Подобная синхронизация приводит к проблемам расфазировки и, в целом, к нарушению отказоустойчивости. Эффективным решением описанных проблем является использование параллельных структур, в которых процесс вычислений управляется потоком данных. Другими словами, в вычис-

лительном устройстве с управлением потоком данных команда начинает выполняться только в том случае, когда получены все активизирующие ее операнды [11]. Такой подход исключает необходимость в глобальном управлении и в глобальной синхронизации. Благодаря этому для матричной обработки можно использовать подход, основанный на управлении данными и автосинхронизации. Свойства сетей Петри как математической теории полностью отвечают требованиям математической модели, предназначенной для описания и дальнейшего исследования матричных структур [1, 2].

В последние годы наблюдается неослабевающий интерес к использованию аппарата сетей Петри (СП) [4–6] для решения задач, связанных с исследованием сложных систем. Можно отметить, что современные подходы, используемые при анализе и синтезе структур сложных систем, основаны на методах декомпозиции, координации и агрегации, структурном подхо-

де, подходе, основанном на теории сложности, и др. Возникающие противоречия между сложностью создаваемых современных систем и традиционными подходами к их проектированию определяют на сегодня одну из основных задач теории систем — задачу разработки методологии и создания систем автоматизированного проектирования, решающих основные задачи синтеза. Данная проблема включает в себя разработку методов и алгоритмов, помогающих вести направленный поиск оптимальных характеристик системы, а также позволяющих контролировать изменение этих характеристик в процессе проектирования. Указанные потребности привели к возникновению и интенсивному развитию системного подхода к проектированию сложных систем, в концепцию которого органически вписывается структурный подход, приобретающий еще большее значение на современном этапе [13—15, 17].

Другим важным вопросом, требующим решения при проектировании сложных систем, является представление имеющихся данных и параллельных процессов в виде специальных формальных объектов, удобных для проведения над ними вычислительных и имитационных экспериментов. С большой долей достоверности можно отметить, что большинство современных дискретных динамических систем характеризуются такими свойствами, как параллелизм, недетерминированность, многоуровневость представления, сочетание синхронных и асинхронных процессов, однородность и др. Поэтому выбор формализованного языка, в наибольшей степени учитывающего особенности современных систем, является основной задачей начального этапа проектирования.

Преимущества СП при описании указанных сложных систем очевидны и широко описаны в литературе. Недостатком данного аппарата является отсутствие инструментария для количественной оценки моделей, выраженных в терминах СП. Вместе с тем теория структур и теория сложности [7—9] предоставляют методы и алгоритмы, позволяющие строить оценочные шкалы для получения количественных характеристик элементов дискретного пространства. В работе [13] были сделаны некоторые шаги, позволяющие в рамках пространства структур СП ввести оценочные шкалы и получить количественные характеристики синтезируемых СП-моделей. В настоящей работе представлено дальнейшее развитие описанного направления исследований.

1. Описание матричной структуры

Одним из методов достижения высокой производительности систем обработки данных реального времени является использование процессорных матриц — специализированных вычислительных структур, состоящих из множества относительно простых устройств (процессорных элементов), объединенных в однородные параллельные системы. Данные процессоры предназначены для решения узкого круга задач, характеризующихся большим объемом обрабатываемой информации. Число процессорных элементов (ПЭ) определяется размерами входного массива данных, а также возможностью разбиения исходных данных на подмассивы, позволяющие проводить независимую обработку [11, 12].

В данном примере рассматривается подход к проектированию матричных процессоров, основанный на структурном анализе СП и тензорной методологии [13, 14].

2.1. Описание примера

Пусть имеется матрица $A = [a_{ij}]$ и вектор $B = [b_j]$ размерности J , необходимо найти вектор $C = AB$. Матрица A может быть разложена на сумму матриц A_i и, следовательно,

$$C = A_1B + A_2B + \dots + A_JB,$$

где A_i — матрица, у которой i -я строка совпадает с i -й строкой матрицы A , а остальные элементы — нули.

Матричное умножение может быть осуществлено за J шагов (рекурсий):

$$c_i^{(k)} = c_i^{(k-1)} + \alpha_i^{(k)} b_j^{(k)},$$

при этом

$$a_i^{(k)} = a_{ik}, b_j^{(k)} = b_k, k = 1, 2, \dots, J; c_i^{(0)} = 0.$$

Процессорная матрица Q , представленная графом потока данных [11, 16] и отображающая процедуру умножения $A[3,3]$ на вектор $B[3]$, приведена на рис. 1, а. Предполагается, что элементы матрицы A находятся в узлах графа (процессорные элементы — ПЭ), элементы вектора B подаются на ПЭ слева, результирующие значения вектора C считываются с верхней строки матричной структуры. Регистры ПЭ, содержащие промежуточные суммы $c_i^{(k)}$, в началь-

ный момент времени установлены в ноль. Структура ПЭ представлена на рис. 1, б.

Процесс начинается с ПЭ, имеющего координаты (1,1) и содержащего элемент a_{11} :

$$q_1^{(1)} = q_1^{(0)} + a_{11}b_1.$$

Вычислительный фронт распространяется к ближайшим соседним ПЭ с координатами (1,2) и (2,1), которые параллельно будут выполнять операции:

$$q_1^{(2)} = q_1^{(1)} + a_{12}b_2 \text{ и } q_2^{(1)} = q_2^{(0)} + a_{21}b_1.$$

Очевидно, что $q_j^{(0)} = 0$.

Следующий фронт активности будет у процессорных элементов (1,3), (2,2) и (3,1). Так создается движущийся вверх и вправо по процессорной матрице вычислительный волновой фронт. После того как волновой фронт проходит через все ПЭ, первая рекурсия считается выполненной.

2.2. Синтез структур

Назовем тактом интервал времени t , в течение которого происходит передача данных между соседними ПЭ, а также выполнение операций умножения и сложения в самом ПЭ. Тогда можно отметить, что первый элемент $c_1 = q_1^{(3)}$ вектора C будет получен через три такта, а вся процедура умножения матрицы на вектор (получение последнего элемента $c_3 = q_3^{(3)}$) завершится через пять тактов.

Так как в каждый момент времени выполняющие рекурсию ПЭ будут различны, на процессорной матрице (рис. 1, а) возможна организация конвейерных вычислений. Это важное замечание, позволяющее обосновать проектирование конвейерных процессоров различной структуры.

Анализ количественных и качественных характеристик процессорных матриц (временные оценки, корректность протекающих параллельных процессов, эффективность загрузки ПЭ и др.), представленных графовыми моделями, достаточно эффективно можно осуществлять с использованием СП.

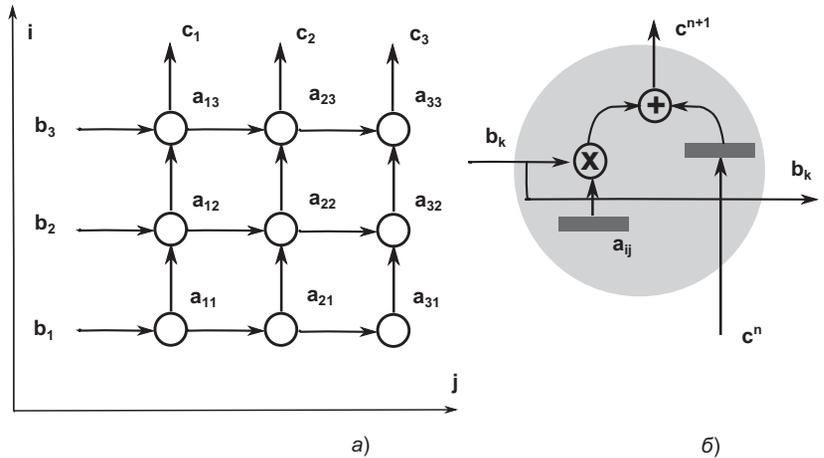


Рис. 1. Граф потока данных умножения матрицы $A(3 \times 3)$ на вектор $B(3)$ (а) и структура процессорного элемента (б)

СП N_1 , моделирующая матричную структуру Q , приведена на рис. 2. Переходы СП-модели t_{ij} соответствуют процессорным элементам структуры, позиции b^*_{ij} моделируют движение элементов вектора B , а позиции c^*_{ij} — движение частичных сумм при получении результата умножения. Предполагается, что элементы матрицы A находятся во внутренней памяти ПЭ. При анализе СП-модели видно, что на первом шаге (такте) активным является лишь переход t_{11} .

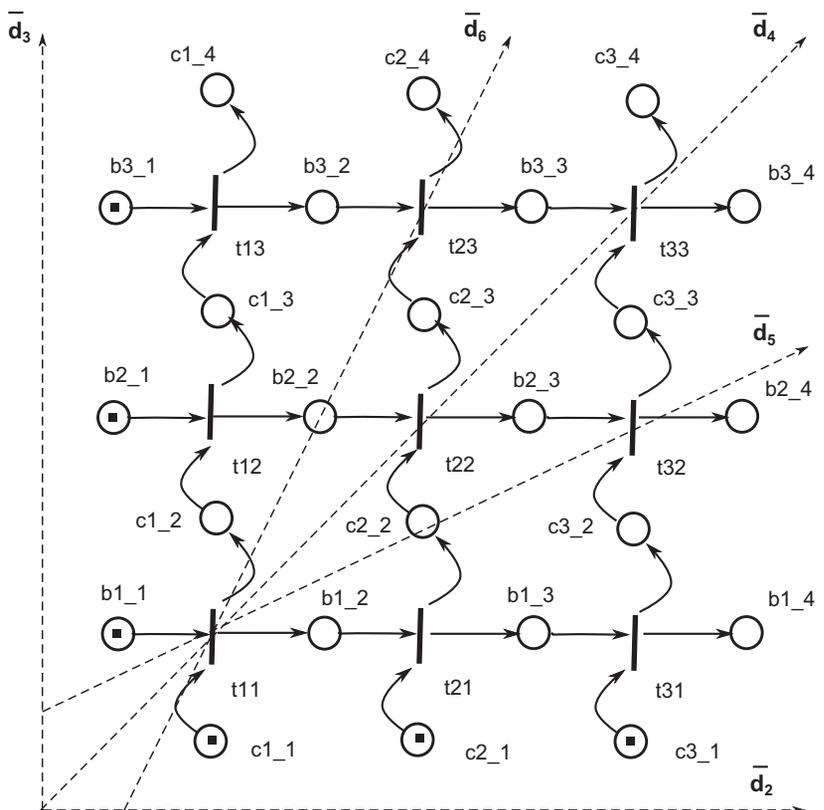


Рис. 2. СП-модель N_1 двумерной параллельной структуры умножения матрицы $A(3 \times 3)$ на вектор $B(3)$

Срабатывание данного перехода активизирует переходы t_{12} и t_{21} , затем активизируются переходы t_{13} , t_{22} , t_{31} (моделирование фронта активности ПЭ) и т. д. Процесс моделирования умножения матрицы на вектор заканчивается, когда в позициях c_{1_4} , ..., c_{3_4} появляются метки.

Анализ функционирования СП-модели N_1 показывает, что при моделировании операции умножения каждый переход срабатывает лишь один раз. Это говорит о том, что ПЭ матрицы (при выполнении одиночной операции умножения матрицы на вектор) будут заняты только в течение одного такта, и их коэффициент занятости будет составлять в среднем 1/5 часть от общего времени выполнения операции. Возникает задача проектирования других параллельных структур, выполняющих ту же функцию (умножение матрицы на вектор), но имеющих более высокие показатели загрузки ПЭ. С этой целью введем понятие вектора проекции \vec{d} ПЭ матричной структуры [11]. При описании нашего подхода мы будем придерживаться линейных проекций, при выполнении которых вершины графа, расположенные на заданной прямой, проецируются в один элемент.

Рассмотрим проекции вершин СП, моделирующей структуру Q , заданные штриховыми линиями и обозначенные $\vec{d}_2, \dots, \vec{d}_6$ (рис. 2). На рис. 3 представлены СП-модели N_2, \dots, N_6 , которые получены в результате выполнения указанных проекций. Отметим некоторые особенности интерпретации вершин данных СП-моделей:

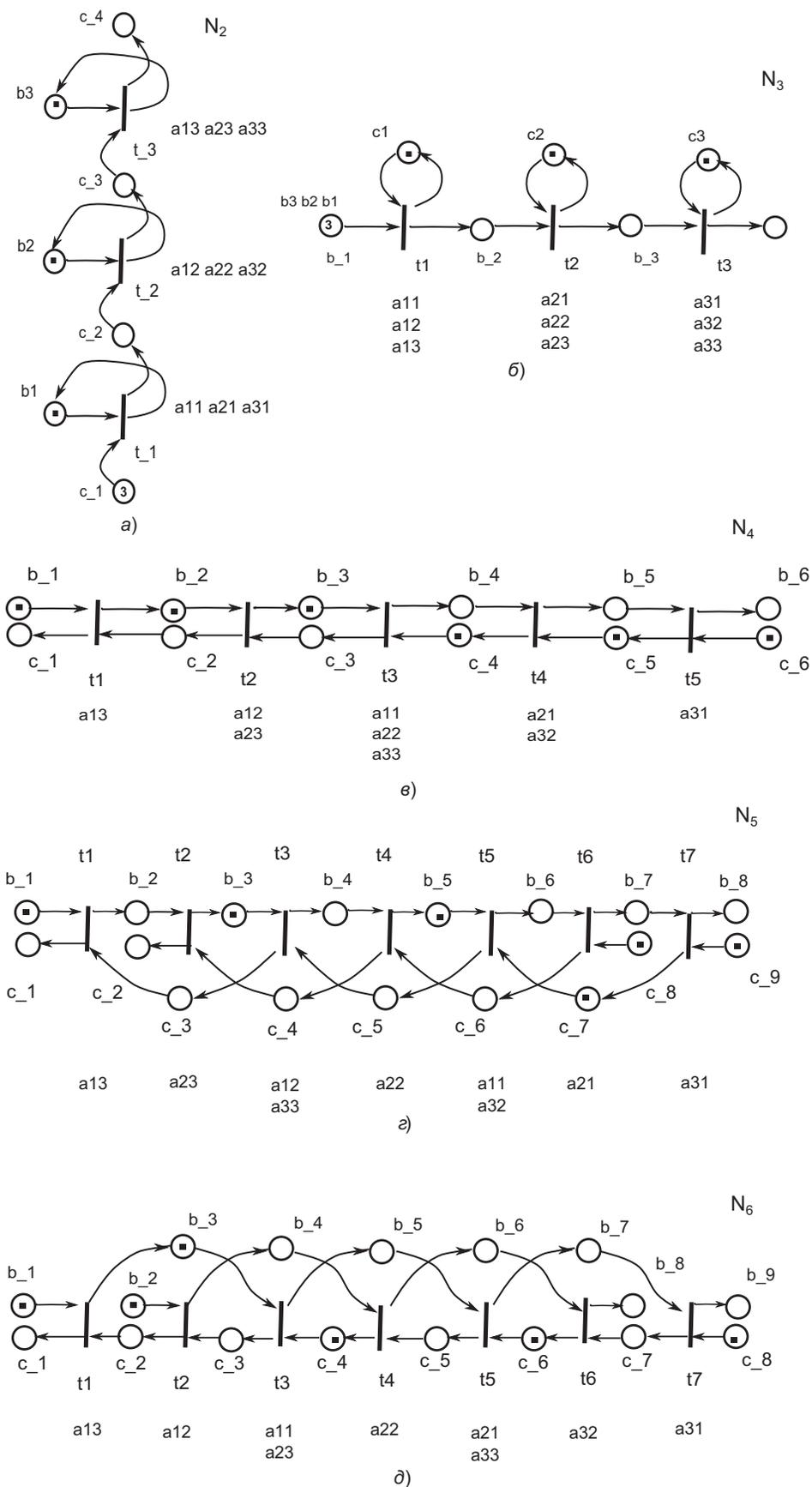


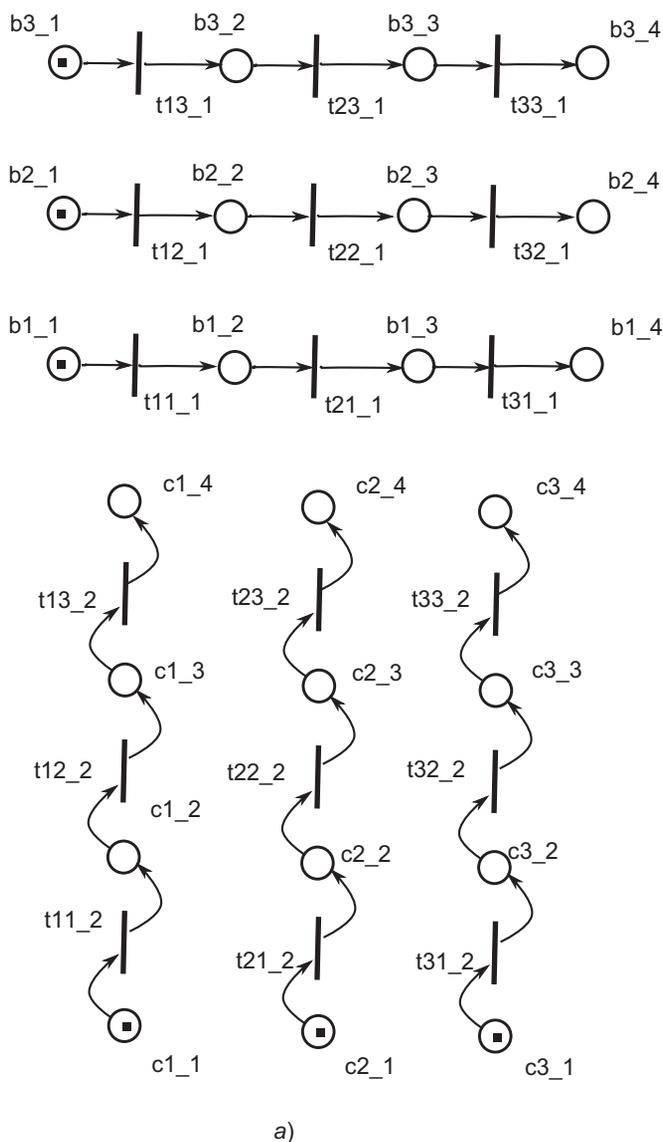
Рис. 3. СП-модели параллельных структур умножения матрицы $A(3 \times 3)$ на вектор $B(3)$, полученные в результате проекций d_2 (a), d_3 (б), d_4 (в), d_5 (г), d_6 (д)

а) на рис. 3, $a-d$ каждому переходу поставлены в соответствие элементы матрицы A , которые должны храниться во внутренней памяти ПЭ матричных структур, моделируемых данными СП-моделями;

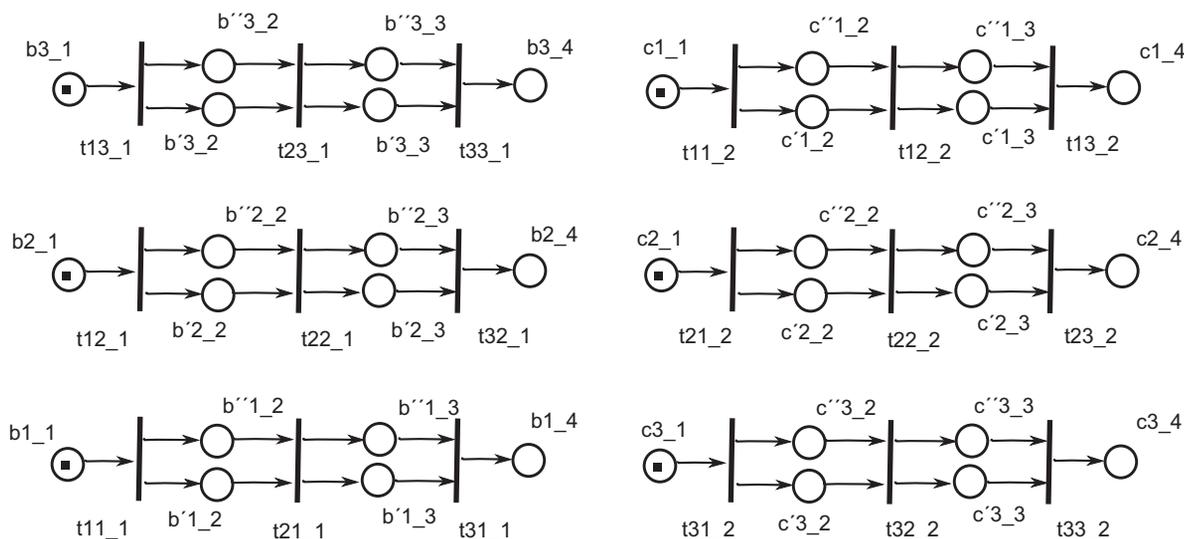
б) петли, полученные в СП-моделях N_2 и N_3 , указывают на образование в новых матричных структурах рекурсивных связей, моделирующих подачу элементов вектора B (рис. 3, a) и получение частичных сумм (рис. 3, b).

Проводя предварительный анализ СП-моделей, представленных на рис. 3, можно отметить, что они имеют одинаковое число тактов на выполнение операции умножения матрицы на вектор, но отличаются числом ПЭ и организацией входных и выходных данных.

В целях демонстрации предлагаемого подхода к проектированию сложных систем рассмотрим процедуру построения СП-моделей, представленных на рис. 3, с помощью структурного анализа и тензорной методологии. Для этого воспользуемся тензорными методами исследования СП [14]. Используя операции деления вершин СП [13], представим исходную СП-модель (см. рис. 2) в виде множества линейных базовых фрагментов (ЛБФ) (рис. 4, a). Данному множеству поставим в соответствие примитивную систему, представленную на рис. 5. В целях формирования одинаковой размерности матриц, описывающих систему ЛБФ и примитивную систему, дополним линейные базовые фрагменты эквивалентными позициями (см. рис. 4, b). Обозначим $D_{\beta'}^{\gamma}$ и D_{β}^{γ} матрицы инцидентности системы ЛБФ и примитивной системы соответственно.



а)



б)

Рис. 4. Система ЛБФ без эквивалентных позиций (а), с эквивалентными позициями (б)

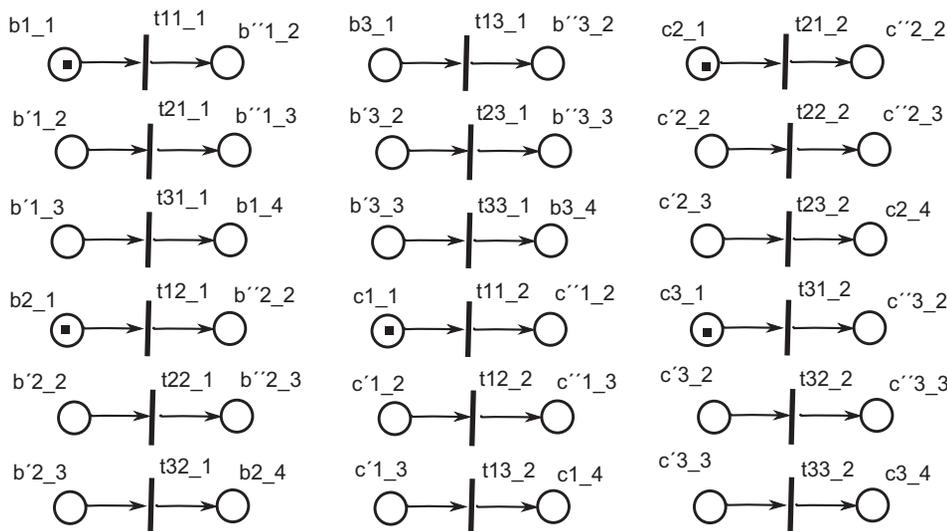


Рис. 5. Прimitивная система N_{PR}

Подставляя матрицы $D_{\beta'}^{\gamma'}$ и D_{β}^{γ} в выражение (1), можно построить тензор преобразования $C_{\beta'\gamma}^{\beta\gamma'}$ [14, 15]:

$$D_{\beta'}^{\gamma'} = C_{\beta'\gamma}^{\beta\gamma'} \times D_{\beta}^{\gamma}. \quad (1)$$

Рассмотрим программы синтеза СП-моделей, которые были представлены на рис. 3. Данные программы построены эмпирически и отображают лишь некоторые из множества возможных программ синтеза каждой из рассматриваемых СП-моделей.

Опишем общий метод построения программ синтеза всех возможных СП-моделей. В качестве исходных данных для синтеза СП-моделей выберем примитивную систему N_{PR} , состоящую из m переходов и n позиций, а также операции объединения вершин примитивной системы, описанные в работе [13].

Поставим в соответствие примитивной системе N_{PR} некоторую последовательность V ($V_T + V_P$), состоящую из $(m + n)$ элементов. Будем считать, что элемент последовательности V равен нулю, если соответствующая вершина примитивной системы N_{PR} не принадлежит подмножеству объединяемых вершин, и элемент последовательности V больше нуля (имеет положительное значение), если соответствующая вершина принадлежит подмножеству объединяемых вершин. При выполнении операции объединения вершин элементы последовательности V должны содержать следующую информацию: во-первых, участвует ли данная вершина в процессе объединения, а во-вторых, какому объединяемому подмножеству данная вершина принадлежит.

Пусть каждый элемент частей V_T и V_P последовательности V меняется в пределах от 1 до z_t и от 1 до z_p , где z_t и z_p — максимальные числа подмножеств вершин, на которое может быть разделено множество переходов и множество позиций для объединения, соответственно.

Так как минимальное число элементов, входящих в подмножество объединяемых вершин, равно 2, то величина z_p определяется следующим образом:

$$z_p = |P|/2,$$

а величина z_t :

$$z_t = \begin{cases} \frac{|T|}{2}, & \text{если } |T| \text{ — четно;} \\ \frac{|T|-1}{2}, & \text{если } |T| \text{ — нечетно.} \end{cases}$$

Пусть части V_T и V_P последовательности V содержат некоторые числа разрядности $m = |T|$ и $n = |P|$ в системах счисления z_t и z_p , соответственно. Тогда каждый разряд данных чисел будет содержать информацию как об участии вершины СП N_{PR} в процессе объединения (значение соответствующего разряда больше нуля), так и о том, какому объединяемому подмножеству данная вершина принадлежит (разряды чисел V_T или V_P , которым соответствуют вершины, входящие в одно объединяемое подмножество, имеют одинаковые значения).

В итоге значения элементов последовательности V задают комбинацию, определяющую программу синтеза СП-модели. Отсюда алгоритм генерации программ синтеза всех СП-моделей, входящих в СП-структуру с рангом $r = m$, можно представить счетчиком, состоящим из двух частей V_T и V_P , каждая из которых работает в системах счисления z_t и z_p соответственно.

Среди комбинаций, генерируемых описанным алгоритмом, существуют такие, которые задают программы синтеза эквивалентных СП-моделей. Для определения таких комбинаций рассмотрим следующие свойства последовательности V .

Свойство 1. Комбинация V , содержащая лишь один положительный элемент, отличный от остальных элементов в части V_T или в части V_P , описывает эквивалентную СП-модель.

Свойство 2. Если существует некоторая подстановка X с областью определения $\{1, 2, \dots, z_x\}$ и подстановка Y с областью определения $\{1, 2, \dots, z_p\}$, которые переводят комбинацию V в комбинацию V' :

$$V_T \xrightarrow{X} V'_T \text{ и } V_P \xrightarrow{Y} V'_P,$$

то комбинация V' описывает программу синтеза эквивалентной СП-модели.

Рассмотрим программы синтеза СП-моделей N_1, \dots, N_6 , представленных на рис. 2 и рис. 3.

Программа синтеза исходной СП-модели N_1 имеет следующий вид:

$$\begin{aligned} t11 &= t11_1 + t11_2 \\ t12 &= t12_1 + t12_2 \\ t13 &= t13_1 + t13_2 \\ t21 &= t21_1 + t21_2 \\ t22 &= t22_1 + t22_2 \\ t23 &= t23_1 + t23_2 \\ t31 &= t31_1 + t31_2 \\ t32 &= t32_1 + t32_2 \\ t33 &= t33_1 + t33_2 \end{aligned} \quad (2)$$

1	4	7	2	5	8	3	6	9	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

t11_1 t21_1 t31_1 t12_1 t22_1 t32_1 t13_1 t23_1 t33_1 t11_2 t12_2 t13_2 t21_2 t22_2 t23_2 t31_2 t32_2 t33_2

Рис. 6. Программа синтеза исходной СП-модели N_1 (часть V_T)

	t11	t21	t31	t12	t22	t32	t13	t23	t33	μ_0
b1_1	-1									1
b1_2	1									
b1_3		-1								
b1_4		1								
b2_1			-1							1
b2_2			1							
b2_3				-1						
b2_4				1						
b3_1					-1					1
b3_2					1					
b3_3						-1				
b3_4						1				
c1_1	-1									1
c1_2	1									
c1_3			-1							
c1_4			1							
c2_1		-1								1
c2_2		1								
c2_3				-1						
c2_4				1						
c3_1			-1							1
c3_2			1							
c3_3					-1					
c3_4					1					

a)

	t11	t21	t31	t12	t22	t32	t13	t23	t33	μ_0
b1_1	-1									1
b1_2	1	-1								
b1_3		1	-1							
b1_4			1	-1						
b2_1				1	-1					1
b2_2					1	-1				
b2_3						1	-1			
b2_4							1	-1		
b3_1								1	-1	1
b3_2								1	-1	
b3_3									1	-1
b3_4										1
c1_1	-1									1
c1_2	1									
c1_3			-1							
c1_4			1							
c2_1		-1								1
c2_2		1								
c2_3				-1						
c2_4				1						
c3_1			-1							1
c3_2			1							
c3_3					-1					
c3_4					1					

б)

	t11	t21	t31	t12	t22	t32	t13	t23	t33	μ_0
b1_1	-1									1
b1_2	1	-1								
b1_3		1	-1							
b1_4			1							
b2_1				-1						1
b2_2				1	-1					
b2_3					1	-1				
b2_4						1				
b3_1							-1			1
b3_2							1	-1		
b3_3								1	-1	
b3_4									1	
c1_1	-1									1
c1_2	1									
c1_3			-1							
c1_4			1							
c2_1		-1								1
c2_2		1								
c2_3				-1						
c2_4				1						
c3_1			-1							1
c3_2			1							
c3_3					-1					
c3_4					1					

в)

Рис. 7. Матрицы инцидентности исходной СП-модели N_1 :

a — матрица D_1 ; б — матрица D'_1 ; в — матрица D'_1 после удаления эквивалентных вершин

Последовательность V , соответствующая данной программе, представлена на рис. 6. Ввиду того что при синтезе СП-модели N_1 объединения позиций не происходит (разряды части V_P равны нулю), последовательность представлена только частью V_T . Рассмотрим процедуру синтеза данной СП-модели с использованием тензорного подхода.

На рис. 7, а представлена матрица инцидентности D_1 исходной СП-модели N_1 (с учетом исключения эквивалентных переходов), выраженная в системе координат примитивной системы. На рис. 7, б представлена матрица инцидентности D'_1 в исходной системе координат (после умножения матрицы D_1 на тензор $C_{\beta\gamma}^{\beta\gamma}$). На рис. 7, в представлена матрица инцидентности D'_1 после исключения эквивалентных позиций. Можно убедиться, что полученная матрица инцидентности D'_1 в точности описывает СП, представленную на рис. 2. Следует отметить, что пустые ячейки матриц инцидентности соответствуют нулевым значениям. Этот прием использован для наглядности и будет использоваться в дальнейшем.

В силу того что дальнейший синтез СП-моделей N_2, \dots, N_6 будет базироваться на исходной СП-модели N_1 , программы синтеза СП-моделей N_2, \dots, N_6 будем строить относительно матриц инцидентности D_1 и D'_1 , описывающих СП-модель N_1 .

Программа синтеза СП-модели N_2

Один из возможных вариантов программы синтеза СП-модели N_2 , дополняющий программу (2), имеет следующий вид:

$$\begin{aligned}
 t_{_1} &= t11 + t21 + t31 \\
 t_{_2} &= t12 + t22 + t32 \\
 t_{_3} &= t13 + t23 + t33 \\
 b1 &= b1_{_1} + b1_{_4} \\
 b2 &= b2_{_1} + b2_{_4} \\
 b3 &= b3_{_1} + b3_{_4}
 \end{aligned}
 \tag{3}$$

Последовательность V , задающая синтез СП-модели N_2 , представлена на рис. 8.

На рис. 9, а представлена матрица инцидентности D_2 после выполнения программы (3) (выражена в системе координат примитивной системы). Следует отметить, что матрица инцидентности в силу своего построения имеет недостаток, который проявляется в отсутствии возможности отображать петли. После выполнения программы (3) такие связи возникают. В целях отображения петель в матрице инцидентности примем обозначение $\langle \text{число} \rangle^*$, в котором знак "*" указывает на наличие петли, а параметр $\langle \text{число} \rangle$ указывает на номер операции объединения вершин в последовательности синтеза, в результате выполнения которой данная петля образовалась. Например, на рис. 9 знаками $1^*, 2^*$ и 3^* обозначено, что в результате операций объединения позиций, отмеченных в части V_p программы синтеза СП-модели N_2 значениями 1, 2 и 3, получены петли.

На рис. 9, б представлена матрица инцидентности D'_2 , полученная после умножения матрицы D_2 на тензор $C_{\beta\gamma}^{\beta\gamma}$ (перевод в исходную систему координат). Эквивалентные переходы в данной матрице не показаны. На рис. 9, в представлена матрица инцидентности D'_2 после исключения эквивалентных вершин СП-модели. Данная матрица полностью соответствует СП-модели N_2 (см. рис. 3, а).

Аналогичным образом могут быть построены СП-модели N_3, \dots, N_6 , программы синтеза которых представлены на рис. 8.

	N_2	N_3	N_4	N_5	N_6	N_7
b1_1	1		2	4	1	1
b"1_2				5	2	4
b'1_2			3	5	2	4
b"1_3				6	3	5
b'1_3			4	6	3	5
b1_4	1					2
b2_1	2		1	2		2
b"2_2			2	3	4	6
b'2_2				3	4	6
b"2_3				4	5	7
b'2_3				4	5	7
b2_4	2		4	5		3
b3_1	3					3
b"3_2			1	1	1	8
b'3_2				1	1	8
b"3_3				2	2	9
b'3_3				2	2	9
b3_4	3		3	3	3	1
c1_1		1	6	11	8	10
c"1_2				9	7	13
c'1_2			7	9	7	13
c"1_3				7	6	14
c'1_3			8	7	6	14
c1_4		1				12
c2_1		2	5		10	12
c"2_2			6	10	9	15
c'2_2				10	9	15
c"2_3				8	8	16
c'2_3				8	8	16
c2_4		2	8		7	11
c3_1		3				11
c"3_2			5	11	11	17
c'3_2				11	11	17
c"3_3				9	10	18
c'3_3				9	10	18
c3_4		3	7	7	9	10
t11	1	1	2	2	1	
t21	1	2	3		2	
t31	1	3				
t12	2	1	1	1		
t22	2	2	2			
t32	2	3	3	2		
t13	3	1				
t23	3	2	1		1	
t33	3	3	2	1	2	

Рис. 8. Программы синтеза СП-моделей

Программа синтеза СП-модели N_7

СП-модели N_2, \dots, N_6 построены на основе алгоритма свертки линейно расположенных вершин СП-модели N_1 . Покажем, что существуют другие программы синтеза СП-моделей, моделирующих работу матричной структуры при умножении матрицы A на вектор B .

Рассмотрим программу синтеза (4). Выполняя последовательность шагов, описанную при синтезе структур N_2, \dots, N_6 , можно получить новую СП-модель N_7 , матрица инцидентности которой представлена на рис. 10.

	t11	t21	t31	t12	t22	t32	t13	t23	t33		μ_0
b1_1	1*	1*	1*							1	1
b*1_2	1	1	1							2	
b'1_2	-1	-1	-1							3	
b''1_3	1	1	1							4	
b'1_3	-1	-1	-1							5	
b1_4	1*	1*	1*							6	
b2_1				2*	2*	2*				7	1
b''2_2				1	1	1				8	
b'2_2				-1	-1	-1				9	
b''2_3				1	1	1				10	
b'2_3				-1	-1	-1				11	
b2_4				2*	2*	2*				12	
b3_1							3*	3*	3*	13	1
b''3_2							1	1	1	14	
b'3_2							-1	-1	-1	15	
b''3_3							1	1	1	16	
b'3_3							-1	-1	-1	17	
b3_4							3*	3*	3*	18	
c1_1	-1	-1	-1							19	1
c'1_2	1	1	1							20	
c''1_2				-1	-1	-1				21	
c'1_3				1	1	1				22	
c''1_3							-1	-1	-1	23	
c1_4							1	1	1	24	
c2_1	-1	-1	-1							25	1
c'2_2	1	1	1							26	
c''2_2				-1	-1	-1				27	
c'2_3				1	1	1				28	
c''2_3							-1	-1	-1	29	
c2_4							1	1	1	30	
c3_1	-1	-1	-1							31	1
c'3_2	1	1	1							32	
c''3_2				-1	-1	-1				33	
c'3_3				1	1	1				34	
c''3_3							-1	-1	-1	35	
c3_4							1	1	1	36	
	t1	t1	t1	t2	t2	t2	t3	t3	t3		

a)

		t11	t21	t31	t12	t22	t32	t13	t23	t33		μ_0
b1	b1_1	1*									1	1
	b''1_2										2	
	b'1_2										3	
	b''1_3										4	
	b'1_3										5	
b2	b1_4	1*			2*						6	
	b'2_1										7	1
	b''2_2										8	
	b'2_2										9	
	b''2_3										10	
	b'2_3										11	
	b2_4				2*						12	
b3	b3_1							3*			13	1
	b''3_2										14	
	b'3_2										15	
	b''3_3										16	
	b'3_3										17	
	b3_4							3*			18	
c_1	c1_1	-1									19	1
c_2	c'1_2	1			-1						20	
	c''1_2	1			1						21	
	c'1_3				1			-1			22	
	c''1_3							-1			23	
	c1_4							1			24	
	c2_1	-1									25	1
	c'2_2	1			-1						26	
	c''2_2	1			-1						27	
c_3	c'2_3				1			-1			28	
	c''2_3				1			-1			29	
	c2_4							1			30	
	c3_1	-1									31	1
	c'3_2	1			-1						32	
	c''3_2	1			-1						33	
	c'3_3				1			-1			34	
	c''3_3				1			-1			35	
c_4	c3_4							1			36	
		t_1			t_2			t_3				

b)

Рис. 9. Матрицы инцидентности СП-модели N_2 :

a — матрица D_2 ; b — матрица D'_2 ; c — матрица D''_2 после удаления эквивалентных вершин

	t_1	t_2	t_3	μ_0
b1	1*			1
b2		2*		1
b3			3*	1
c_1	-1			3
c_2	1	-1		
c_3		1	-1	
c_4			1	

в)

$$\begin{aligned}
 b_{-1} &= b1_{-1} + b3_{-4} \\
 b_{-2} &= b1_{-4} + b2_{-1} \\
 b_{-3} &= b2_{-4} + b3_{-1} \\
 b1_{-2} &= b'1_{-2} + b''1_{-2} \\
 b1_{-3} &= b'1_{-3} + b''1_{-3} \\
 b2_{-2} &= b'2_{-2} + b''2_{-2} \\
 b2_{-3} &= b'2_{-3} + b''2_{-3} \\
 b3_{-2} &= b'3_{-2} + b''3_{-2} \\
 b3_{-3} &= b'3_{-3} + b''3_{-3} \\
 c_{-1} &= c1_{-1} + c3_{-4} \\
 c_{-3} &= c2_{-4} + c3_{-1} \\
 c_{-2} &= c1_{-4} + c2_{-1} \\
 c1_{-2} &= c'1_{-2} + c''1_{-2} \\
 c1_{-3} &= c'1_{-3} + c''1_{-3} \\
 c2_{-2} &= c'2_{-2} + c''2_{-2} \\
 c2_{-3} &= c'2_{-3} + c''2_{-3} \\
 c3_{-2} &= c'3_{-2} + c''3_{-2} \\
 c3_{-3} &= c'3_{-3} + c''3_{-3}
 \end{aligned}$$

(4)

	t11	t21	t31	t12	t22	t32	t13	t23	t33	μ_0
b_1	-1								1	1
b1_2	2	-2								
b1_3		2	-2							
b_2			1	-1						1
b2_2				2	-2					
b2_3					2	-2				
b_3						1	-1			1
b3_2							2	-2		
b3_3								2	-2	
c_1	-1								1	1
c1_2	2			-2						
c1_3				2			-2			
c_2		-1					1			1
c2_2		2			-2					
c2_3					2			-2		
c_3			-1					1		1
c3_2			2			-2				
c3_3						2			-2	

Рис. 10. Матрица инцидентности СП-модели N_7 (D'_7) после удаления эквивалентных вершин

Таблица 1

Такты работы СП-модели N_2

№ ПЭ	Такты				
	1	2	3	4	5
1 ($t_{_1}$)	$a_{11} * b_1$	$a_{21} * b_1$	$a_{31} * b_1$		
2 ($t_{_2}$)		$a_{11} * b_1 + a_{12} * b_2$	$a_{21} * b_1 + a_{22} * b_2$	$a_{31} * b_1 + a_{32} * b_1$	
3 ($t_{_3}$)			$a_{11} * b_1 + a_{12} * b_2 + a_{13} * b_3$	$a_{21} * b_1 + a_{22} * b_2 + a_{23} * b_3$	$a_{31} * b_1 + a_{32} * b_2 + a_{33} * b_3$

Таблица 2

Такты работы СП-модели N_4

№ ПЭ	Такты				
	1	2	3	4	5
1 ($t_{_1}$)			$a_{11} * b_1 + a_{12} * b_2 + a_{13} * b_3$		
2 ($t_{_2}$)		$a_{11} * b_1 + a_{12} * b_2$		$a_{21} * b_1 + a_{22} * b_2 + a_{23} * b_3$	
3 ($t_{_3}$)	$a_{11} * b_1$		$a_{21} * b_1 + a_{22} * b_2$		$a_{31} * b_1 + a_{32} * b_2 + a_{33} * b_3$
4 ($t_{_4}$)		$a_{21} * b_1$		$a_{31} * b_1 + a_{32} * b_2$	
5 ($t_{_5}$)			$a_{31} * b_1$		

Таблица 3

Такты работы СП-модели N_5

№ ПЭ	Такты				
	1	2	3	4	5
1 ($t_{_1}$)			$a_{11} * b_1 + a_{12} * b_2 + a_{13} * b_3$		
2 ($t_{_2}$)				$a_{21} * b_1 + a_{22} * b_2 + a_{23} * b_3$	
3 ($t_{_3}$)		$a_{11} * b_1 + a_{12} * b_2$			$a_{31} * b_1 + a_{32} * b_2 + a_{33} * b_3$
4 ($t_{_4}$)			$a_{21} * b_1 + a_{22} * b_2$		
5 ($t_{_5}$)	$a_{11} * b_1$			$a_{31} * b_1 + a_{32} * b_2$	
6 ($t_{_6}$)		$a_{21} * b_1$			
7 ($t_{_7}$)			$a_{31} * b_1$		

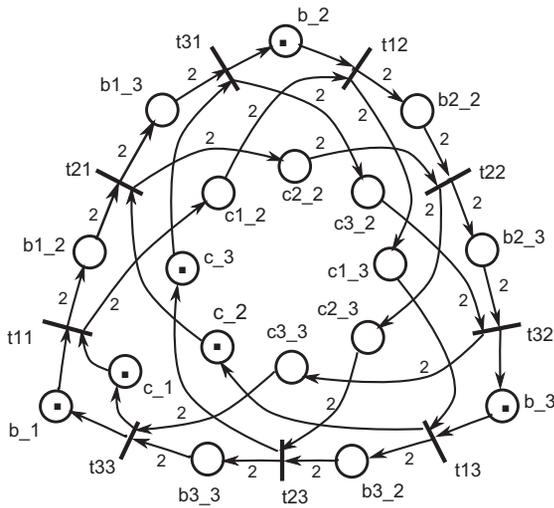


Рис. 11. СП-модель N_7 параллельной структуры умножения матрицы $A(3 \times 3)$ на вектор $B(3)$, полученной в результате выполнения программы синтеза (4)

Последовательность V , задающая синтез СП-модели N_7 , представлена на рис. 8.

2.3. Анализ структур

В ходе анализа построенных структур оценим характеристики процессорных матриц, моделируемых СП-моделями N_1, \dots, N_7 . Потактовую работу СП-моделей можно представить в виде специальных таблиц, показывающих очередность срабатывания переходов и их загрузку, организацию потока входных и выходных данных, а также условия окончания выполняемой операции. Табл. 1—4 демонстрируют функционирование некоторых построенных СП-моделей.

В целях количественного и качественного анализа построенных матричных структур введем некоторые критерии оценки структурных и динамических свойств соответствующих СП-моделей. Обозначим P_i — число тактов, в течение которых ПЭ был занят; P — общее число тактов, в течение которых выполняется операция; M — число ПЭ, входящих в анализируемую структуру.

Введем коэффициент использования i -го ПЭ:

$$k_i = \frac{P_i}{P}.$$

Тогда средний коэффициент использования ПЭ матричной структуры можно представить следующей формулой:

$$k_{cp} = \frac{M}{\sum_{i=1}^M P_i} (PM).$$

Например, для структур, представленных на рис. 3, δ и рис. 3, ϵ ,

$$k_{cp} = (1 + 2 + 3 + 2 + 1) / (5 \cdot 5) = 0,36$$

и $k_{cp} = (1 + 1 + 2 + 1 + 2 + 1 + 1) / (5 \cdot 7) = 0,26$ соответственно.

Таблица 4

Такты работы СП-модели N_7

№ ПЭ	Такты				
	1	2	3	4	5
1 (t_11)	$a_{11} * b_1$				
2 (t_21)		$a_{21} * b_1$			
3 (t_31)			$a_{31} * b_1$		
4 (t_12)		$a_{11} * b_1 + a_{12} * b_2$			
5 (t_22)			$a_{21} * b_1 + a_{22} * b_2$		
6 (t_32)				$a_{31} * b_1 + a_{32} * b_2$	
7 (t_13)			$a_{11} * b_1 + a_{12} * b_2 + a_{13} * b_3$		
8 (t_23)				$a_{21} * b_1 + a_{22} * b_2 + a_{23} * b_3$	
9 (t_33)					$a_{31} * b_1 + a_{32} * b_2 + a_{33} * b_3$

Таблица 5

Количественные показатели синтезированных СП-моделей

№	Идентификатор СП-модели	Число ПЭ	Число тактов	Коэффициент использования ПЭ (k_i)	Средний коэффициент использования ПЭ (k_{cp})
1	N_1	9	5	1/5	0,11
2	N_2	3	5	3/5	0,60
3	N_3	3	5	3/5	0,60
4	N_4	5	5	Величина переменная	0,36
5	N_5	7	5	Величина переменная	0,26
6	N_6	7	5	Величина переменная	0,26
7	N_7	9	5	Величина переменная	0,2
8	N_0	1	9	1	1

Вычисленные коэффициенты для синтезированных СП-моделей сведены в табл. 5. Для более полного сравнения в последней строке таблицы приведены значения коэффициентов для структуры N_0 , включающей один ПЭ.

На рис. 12 приведена зависимость коэффициента использования ПЭ синтезированных структур от числа ПЭ. На основе данной зависимости можно оценить, например, какими значениями будут характеризоваться структуры, полученные

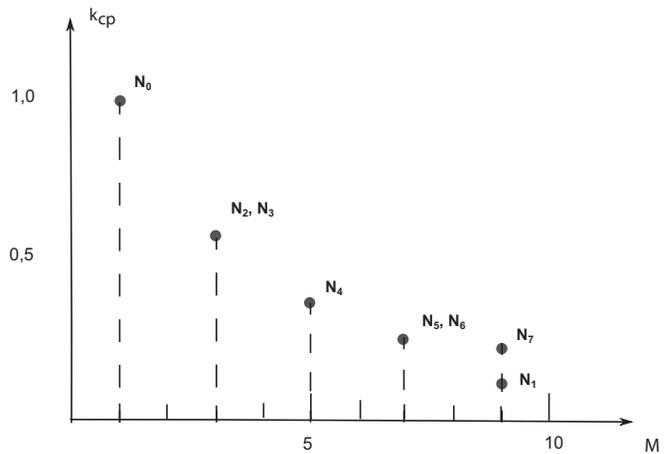


Рис. 12. Зависимость коэффициента использования ПЭ от конфигурации матричной структуры

в результате выполнения других программ синтеза, не рассмотренных в данной работе.

Заключение

Описанные в данной работе результаты дают предпосылки для построения метода синтеза новых структур параллельных вычислительных систем. В основу метода положены модели, выраженные в терминах сетей Петри. Возможность существования подобных методов проектирования продемонстрирована на синтезе структур, реализующих операцию матричного умножения. Однако класс задач, для решения которых предложенный подход актуален, более широк.

Операция проекции расположенных линейно вершин СП-моделей в одну вершину (операция свертки) позволяет не только получить структуры с новыми свойствами, но и перейти к анализу моделей, имеющих меньшую размерность. Данный переход позволяет применять известные методы анализа СП к сетям меньшей размерности, что в итоге повышает эффективность и результативность процедур анализа.

Использование возможностей структурного анализа сложных систем, и, в частности, СП-моделей, позволяет на множестве построенных моделей задать систему оценочных шкал и провести количественный анализ свойств построенных моделей, что дает основу для построения программных средств поиска оптимальных структур по заданным критериям.

Основными препятствиями, мешающими непосредственному использованию предложенных методов, являются ограничения, связанные с числом синтезируемых СП-моделей. Это число растет экспоненциально в зависи-

мости от размерности примитивной системы. Для того чтобы описанные методы можно было бы использовать на практике, *необходим поиск новых приемов и подходов, которые ограничили бы множество рассматриваемых СП-моделей*. Необходимы средства, позволяющие исключать из рассмотрения модели, не отвечающие наперед заданным требованиям. Данные требования могут быть представлены, например, в виде множества правил, определяющих доступные или запрещенные связи между позициями и переходами синтезируемых СП-моделей, или в виде некоторых численных ограничений, отсекающих неудовлетворяющие им СП-модели. Набор подобных правил мог бы составить некоторую экспертную базу данных (базу знаний), позволяющую существенно сократить число СП-моделей, представляемых для окончательного анализа.

Список литературы

1. **Котов В. Е.** Сети Петри. М.: Наука, 1984.
2. **Питерсон Дж.** Теория сетей Петри и моделирование систем. М.: Мир, 1984.
3. **Кулагин В. П.** Проблемы анализа и синтеза структур параллельных вычислительных систем // Информационные технологии. 1997. № 1. С. 2—8.
4. **32nd International Conference on Application and Theory of Petri Nets and Concurrency**, 2011. URL: <http://www.jaist.ac.jp/acsd-petrinets2011/>.
5. **33rd International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2012)**, Hamburg, Germany, June 25—29, 2012. URL: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/meetings/pn2012/>
6. **34th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2013)**, Milano, Italy, June 24—28, 2013. URL: <http://www.mc3.disco.unimib.it/petrinets2013/>.
7. **Анкудинов Г. И.** Синтез структуры сложных объектов. Логико-комбинаторный подход. Л.: ЛГУ, 1986.
8. **Биркгоф Г.** Теория структур. М.: Иностранная литература, 1952.
9. **Скорняков Л. А.** Элементы теории структур. М.: Наука, 1982.
10. **Cabasino M. P., Giua A., Seatzu N.** Structural analysis of Petri nets // Control of Discrete-Event Systems. Lecture Notes in Computer Science. № 433. Springer-Verlag, London, 2013. P. 213—233.
11. **Кун С.** Матричные процессоры на СБИС: Пер. с англ. М.: Мир, 1991. 672 с.
12. **Сверхбольшие** интегральные схемы и современная обработка сигналов: Пер. с англ. / Под ред. С. Гуна, Х. Уайтхауса, Т. Кайлата. М.: Радио и связь, 1989. 472 с.
13. **Кулагин В. П., Дубинин В. Н.** Структурный анализ сетей Петри // Информационные технологии. 2016. № 1 (22). С. 3—13.
14. **Кулагин В. П.** Тензорные методы исследования структур сетей Петри // Информационные технологии. 2015. Т. 21, № 2. С. 83—94.
15. **Кулагин В. П.** Методы построения тензоров преобразования для сетевых моделей сложных систем // Информатизация образования и науки. 2015. № 4 (28). С. 133—147.
16. **Kulagin V.** Design of control systems for parallel computing structures based on net models // 2016 International Siberian Conference on Control and Communications (SIBCON). Proceedings. M.: HSE, 2016.
17. **Drozdov D., Dubinin V., Kulagin V.** Petri nets behavioral equivalence checking in SMV // 2016 International Siberian Conference on Control and Communications (SIBCON). Proceedings. M.: HSE, 2016.

V. P. Kulagin, D. Sc., Professor, Head of the Department "Hardware, software and mathematical support of computer systems", e-mail: Kulagin_VP@mgupi.ru,
E. S. Malych, Master Student of the Department "Hardware, software and mathematical support of computer systems"; e-mail: katya.miller.95@yandex.ru,
 MIREA-Russian University of Technology, Moscow

Design of Matrix Computing Structures using Petri Nets

In article on the example of projection of matrix computing structures the synthesis algorithm of the simulativng Petri nets is offered. Approach to synthesis of new structures on the basis of use of vectors of a projection is described. It is shown that this approach doesn't allow to receive all set of the possible Petri nets modeling realization of the function set on matrix structure. The original algorithm of projection of the complete set of the possible joint ventures models of matrix structures for the given function is described. The algorithm is based on use of a structure analysis and tensor methodology. Oscillation of programs of synthesis of structures of the joint venture models is result of work of an algorithm. It is shown that among the generated programs exist such which synthesize the equivalent joint ventures models. The properties revealing the "equivalent" programs are formulated. For the purpose of the quantitative and qualitative analysis of the constructed matrix structures criteria for evaluation of structural and dynamic properties of the synthesizable joint ventures models are injected. It is shown that owing to exponential body height of number of the synthesizable joint ventures models from dimension of an initial primitive system, searching of new receptions and approaches which would exclude from consideration of model is necessary, not meeting beforehand the given requirements.

Keywords: matrix structure, parallel computations, Petri nets, tensor analysis, program synthesis, evaluation of structural properties

DOI: 10.17587/it.25.271-283

References

1. **Kotov V. E.** Seti Petri, Moscow, Nauka, 1984 (in Russian).
2. **Piterson Dzh.** Teoriya setej Petri i modelirovanie sistem, Moscow, Mir, 1984.
3. **Kulagin V. P.** Problemy` analiza i sinteza struktur parallel`ny`x vy`chislitel`ny`x sistem, *Informacionny'e Tekhnologii*, 1997, no. 1, pp. 2–8.
4. **32nd International Conference on Application and Theory of Petri Nets and Concurrency**, 2011, available at: [//www.jaist.ac.jp/acsd-petrinets2011/](http://www.jaist.ac.jp/acsd-petrinets2011/).
5. **33rd International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2012)**, Hamburg, Germany, June 25–29, 2012, available at: [//www.informatik.uni-hamburg.de/TGI/PetriNets/meetings/pn2012/](http://www.informatik.uni-hamburg.de/TGI/PetriNets/meetings/pn2012/)
6. **34th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2013)**, Milano, Italy, June 24–28, 2013, available at: [//www.mc3.disco.unimib.it/petrinets2013/](http://www.mc3.disco.unimib.it/petrinets2013/).
7. **Ankundinov G. I.** Sintez struktury` slozhny`x ob`ektov. Logiko-kombinatorny`j podhod. L.: LGU, 1986.
8. **Birkhof G.** Teoriya struktur. M.: Inostrannaya literatura, 1952.
9. **Skorniyakov L. A.** E`lementy` teorii struktur, Moscow, Nauka, 1982.
10. **Cabasio M. P., Giua A., Seatzu N.** Structural analysis of Petri nets, *Control of Discrete-Event Systems. Lecture Notes in Computer Science*, № 433, Springer-Verlag, London, 2013, pp. 217–233.
11. **Kun S.** Matrichny`e processory` na SBIS: Per. s angl, Moscow, Mir, 1991, 672 p.
12. **Sverxbol'shie integral`ny`e sxemy` i sovremennaya obrabotka signalov:** Per. s angl. / Pod red. S. Guna, X. Uajtxausa, T. Kajlata, Moscow, Radio i svyaz`, 1989, 472 p.
13. **Kulagin V. P., Dubinin V. N.** Strukturny`j analiz setej Petri, *Informacionny'e Tekhnologii*, 2016, no. 1 (22), pp. 3–13.
14. **Kulagin V. P.** Tenzorny`e metody` issledovaniya struktur setej Petri, *Informacionny'e tekhnologii*, 2015, T. 21, № 2, pp. 83–94.
15. **Kulagin V. P.** Metody` postroeniya tenzorov preobrazovaniya dlya setevy`x modelej slozhny`x sistem, *Informatizaciya obrazovaniya i nauki*, 2015, № 4 (28), pp. 133–147.
16. **Kulagin V.** Design of control systems for parallel computing structures based on net models, in: 2016 International Siberian Conference on Control and Communications (SIBCON). Proceedings, Moscow, HSE, 2016.
17. **Drozdov D., Dubinin V., Kulagin V.** Petri nets behavioral equivalence checking in SMV, in: 2016 International Siberian Conference on Control and Communications (SIBCON). Proceedings, Moscow, HSE, 2016.

УДК 001.89

DOI: 10.17587/it.25.283-292

Р. М. Юсупов, чл.-корр. РАН, науч. рук. института, e-mail: yusupov@iias.spb.su,
В. И. Воробьев, д-р техн. наук, гл. науч. сотр., e-mail: vvi@iias.spb.su,
М. Ю. Петров, вед. программист, e-mail: miha@iias.spb.su,
Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН)

Представление эволюции средств вычислительной техники в экспозиции Музея истории СПИИРАН

Описана история развития средств вычислительной техники в СПИИРАН. Приведен обзор основных экспонатов: отдельных устройств и компонентов вычислительной техники. Экспозиция отражает отдельные этапы становления и развития вычислительной техники в стране и возникновения информатики как фундаментальной науки. Систематизирован опыт использования вычислительной техники в научных исследованиях.

Ключевые слова: экспозиция музея, средства вычислительной техники, математическое обеспечение ЭВМ, компьютерные сети, центр коллективного пользования

Введение

Появление средств вычислительной техники следует считать одним из важнейших достижений цивилизации, обусловивших переход от индустриального к информационному обществу. Создание научно-технического музея истории вычислительной техники направлено на формирование понимания основных тенденций развития информационных технологий от истоков до современного их состояния. Экспозиция музея предназначена для различных возрастных групп школьников, студентов, пе-

дагогов и научных работников. Она знакомит посетителей с важнейшими вехами развития информатики и средствами вычислительной техники на различных этапах их развития.

Музей истории СПИИРАН является одним из источников научно-образовательной деятельности в области истории информатики, сохранения и демонстрации средств вычислительной техники в северо-западном регионе России [1,2]. Коллекция содержит более 1000 объектов, представленных в экспозиции. Выставленные образцы отражают основные вехи метасистемного перехода в развитии средств вычислитель-