

Г. А. Поляков, Е. Г. Толстолужская

Адаптивная технология автоматического синтеза архитектурно- и проблемно-ориентированных мультипараллельных времяпараметризованных моделей задач

АННОТАЦИЯ. В статье описывается адаптивная технология автоматического синтеза мультипараллельных времяпараметризованных моделей задач, приводятся основные этапы формального синтеза максимально параллельных временных моделей алгоритмов и времяпараметризованных глобально-параллельных моделей задач при ограничении доступного ресурса. Этапы проиллюстрированы на примере синтеза времяпараметризованных параллельных моделей задачи решения системы двух линейных уравнений по алгоритму Крамера.

Ключевые слова и фразы: времяпараметризованная параллельная модель, числовая спецификация программы, граф программы, семантико-числовая спецификация, Си-граф.

Введение

Основой решения проблемы повышения эффективности известных параллельных процессоров и многопроцессорных ВС и создания перспективных Адаптивных Самоорганизующихся Вычислительных Систем (АСВС), Интеллектуально-Вычислительных Систем (ИВС) и Адаптивных Распределенных Вычислительных Сетей является разработка технологий автоматического проектирования мультипараллельных аппаратных и программных средств [1-5].

1. Состав факторов, подлежащих учету при синтезе временных параллельных моделей алгоритмов

Перечень основных групп факторов, поддерживаемых рассматриваемыми далее методами формального синтеза параллельных моделей алгоритмов, представляет рис. 1. Рассмотрим более подробно состав факторов, входящих в различные группы.

Прикладная ориентация моделей означает:

- поддержку формального проектирования мультипараллельных аппаратных средств (включая моно- и многофункциональные спец-

процессоры с жесткой структурой, адаптивные многофункциональные спецпроцессоры с аппаратно реконфигурируемой структурой, универсальные параллельные микропроцессоры);

- поддержку формального проектирования мультипараллельных программных средств для известных параллельных процессоров и многопроцессорных ВС, а также для перспективных АСВС и ИВС;
- обеспечение совместного проектирования параллельных аппаратно-программных средств сложных систем на кристалле (*System-on-Chip, SoC*);
- обеспечение проектирования параллельного аппаратно-программного обеспечения информационно-управляющих систем (ИУС), работающих в реальном времени.



Рис. 1. Системные группы факторов, обуславливающие многообразие классов временных параллельных моделей алгоритмов

Классы и характеристики алгоритмов включают:

- поддержку двух классов алгоритмов: «частных алгоритмов» (ЧА), представляющих отдельные задачи, и «комплексных алгоритмов» (КА), представляющих взаимосвязанные совокупности частных алгоритмов, реализуемых сложными информационно-управляющими и вычислительными системами;

- формализацию описания «частных алгоритмов» и «комплексных алгоритмов» с помощью структур семантико-числовой спецификации (СЧС);
- поддержку всех типов алгоритмов (неразветвляющихся, разветвляющихся, циклических с простыми и сложными циклами различных видов).

Состав методов параллельной обработки включает:

- метод совмещения независимых операций (СО),
- метод конвейерной (КО), кодово-матричный метод (КМ),
- декомпозиционный метод (ДК),
- метод смеси алгоритмов (СА).

Характеристики элементной базы цифровой вычислительной техники:

- состав операций/функций на логическом и вентиляльном уровнях, реализованных аппаратно;
- количество входов/выходов элементов;
- коэффициенты объединения по входам и нагрузочные способности выходов;
- значения временных задержек;
- эквивалентная вентиляльная сложность элементов/компонентов;
- состав операций/функций на системном (уровень IP, Intellectual Property) и функциональном (RTL, Register Transfer Level) уровнях, реализованных аппаратно;
- интерфейсные, временные и сложностные/стоимостные параметры;
- формализация описания элементной базы на всех уровнях с помощью структур семантико-числовой спецификации (СЧС).

Классы и архитектура микропроцессоров включают:

- поддержку архитектур микропроцессоров классов «суперскалярный», с «длинной командной строкой», VLIW, с «управлением потоком данных», FLOW;
- учет для каждого микропроцессора состава используемых методов параллельной обработки данных, количества обрабатывающих устройств каждого типа, иерархии и емкости памяти, числа портов одновременного чтения/приема данных, значений тактовой частоты и/или длительности выполнения различных операций и функций, количество адресов в процессорных инструкциях, формализацию описания архитектуры микропроцессоров на с помощью структур семантико-числовой спецификации (СЧС).

Классы и архитектура параллельных мультипроцессорных ВС включают:

- поддержку архитектур параллельных ВС с симметричной мультипроцессорной обработкой и разделяемой памятью (классов SMP,

NUMA, cc NUMA) и массово параллельных систем с распределенной памятью и передачей сообщений (классов MPP, Cluster);

- учет для параллельной ВС каждого класса количества, типа и характеристик процессоров; иерархии памяти ВС; топологии и временных характеристик коммутационной среды; характеристик программного обеспечения (операционной системы, библиотек параллельного программирования, Run Time Library и т. п).

Система требований и ограничений, предъявляемых к аппаратной или программной реализации алгоритмов, включает:

требуемые значения времени выполнения алгоритмов/комплексов алгоритмов, значения тактовой частоты ввода и обработки данных, ограничения на количество (стоимость) оборудования и т. п.

2. Результаты исследования

Возможность изменения в широких пределах перечисленных факторов обуславливает необходимость разработки методов синтеза следующих вариантов параллельных времяпараметризованных моделей алгоритмов, ориентированных (в зависимости от областей практического использования) на программную и/или аппаратную реализацию:

- параллельных моделей с минимальным временем T_{min} или заданным $T_{зад}$ временем реализации алгоритма при отсутствии ограничений на тактовую частоту $F_{зад}$ и сложность (стоимость) $Q_{зад}$;
- параллельных моделей с минимальным временем T_{min} или заданным $T_{зад}$ временем реализации алгоритма при наличии ограничений на тактовую частоту $F_{зад}$ и сложность (стоимость) $Q_{зад}$;
- параллельных моделей с заданной тактовой частотой $F_{зад}$ (или заданной величиной тактового интервала $TT_{зад}$) при отсутствии или наличии ограничений на время $T_{зад}$ реализации алгоритма и сложность (стоимость) $Q_{зад}$;
- параллельных моделей с минимально возможной Q_{min} или заданной $Q_{зад}$ сложностью (стоимостью) при отсутствии ограничений на время $T_{зад}$ и на тактовую частоту $F_{зад}$;
- наиболее общего варианта – синтеза параллельных моделей, удовлетворяющих (при условии разрешимости задачи синтеза) требованиям к минимальному T_{min} или заданному $T_{зад}$ времени реализации алгоритма, к заданной тактовой частоте $F_{зад}$ и сложности (стоимости) $Q_{зад}$ аппаратной и/или программной реализации модели.

Временные параллельные модели могут использовать различные методы параллельной обработки данных. Состав фактически применяемых методов определяется, в первую очередь, особенностями выполняемых задач/алгоритмов и конкретными требованиями и ограничениями пользователей, предъявляемыми к выполнению моделей в различных

прикладных областях. Классификацию временных параллельных моделей по составу поддерживаемых методов параллельной обработки представляет рис. 2 (подробное описание различных методов параллелизма дано в [4, 5]).



Рис. 2. Классы времяпараметризованных параллельных моделей

Классификация временных параллельных моделей по составу поддерживаемых требований и ограничений показана на рис. 3.

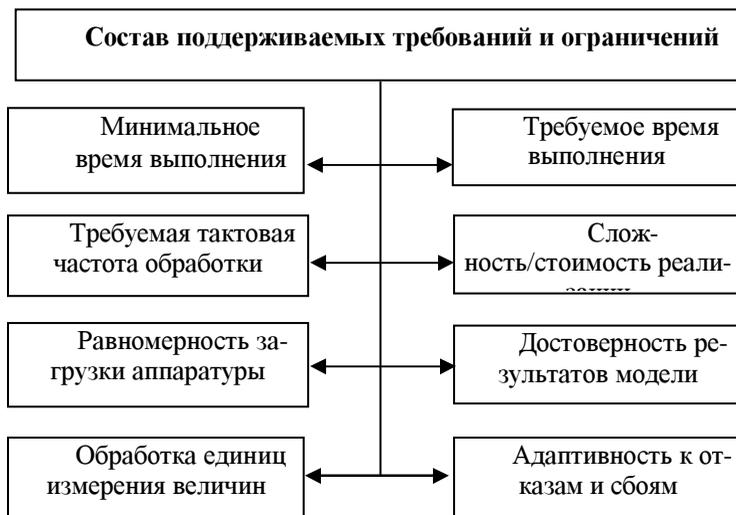


Рис. 3. Состав поддерживаемых моделями требований/ограничений

3. Технология синтеза мультипараллельных времяпараметризованных моделей задач

Основные составляющие адаптивной технологии автоматического синтеза архитектурно - и проблемно-ориентированных мультипараллельных времяпараметризованных моделей задач:

- формальный метод преобразования Си-программ в структуры семантико-числовой спецификации (СЧС);
- формальные методы синтеза архитектурно-ориентированных временных параллельных моделей задач для традиционных программно управляемых универсальных/специализированных процессоров и традиционных многопроцессорных ВС классов SMP, NUMA, cc NUMA, MPP, CLUSTER;
- формальные методы синтеза проблемно-ориентированных временных мультипараллельных моделей задач, поддерживающие проектирование аппаратно-программного обеспечения АСВС, удовлетворяющих заданным требованиям/ограничениям;
- формальные методы верификации корректности структур семантико-числовой спецификации статических и времяпараметризованных параллельных моделей алгоритмов;
- методы оценки показателей эффективности временных параллельных моделей алгоритмов и степени их соответствия заданным требованиям/ограничениям;

- методы визуализации статических и временных параллельных моделей алгоритмов.

Исходные данные синтеза времяпараметризованных мультипараллельных моделей задач:

- исходные тексты Си-программ задач;
- проблемная направленность: синтез проблемно-ориентированных временных параллельных моделей (для использования в АСВС) или архитектурно-ориентированных (для применения в рамках традиционных параллельных процессоров и многопроцессорных ВС);
- прикладная направленность: синтез временных параллельных моделей – для последующей программной реализации или для аппаратной реализации;
- единицы измерения входных и выходных физических величин Си-программ;
- состав методов параллельной обработки данных;
- принципы построения и архитектура АСВС;
- архитектура, конфигурация и временные характеристики традиционных параллельных процессоров и ВС;
- система требований и ограничений (время выполнения программы, тактовая частота обработки данных, производительность, сложность/стоимость и др.).

Выходные результаты синтеза времяпараметризованных мультипараллельных моделей:

- проблемно-ориентированные временные моно- и мультипараллельные модели задач, удовлетворяющие заданным требованиям и ограничениям, как основа формального и автоматического синтеза программных средств АСВС и оценки показателей эффективности мультипараллельных программных моделей;
- аппаратно-ориентированные временные параллельные модели как основа формального и автоматического синтеза аппаратных средств АСВС и оценки показателей эффективности мультипараллельных аппаратных моделей;
- архитектурно-ориентированные временные параллельные модели задач для программно управляемых универсальных и специализированных процессоров и традиционных многопроцессорных ВС классов *SMP*, *NUMA*, *MPP*, *CLUSTER*; оценки показателей эффективности параллельных программных моделей.

Наличие большого числа факторов, оказывающих влияние на процесс параллельной реализации алгоритмов, не позволяет оценивать эффективность их выполнения с помощью одного показателя, поэтому на практике применяется, как правило, совокупность частных показателей. В качестве таких показателей обычно используются производительность вычислительных средств реализации алгоритмов, достоверность реализации алгоритмов, точность результатов, время выполнения алгоритмов,

цена эффективного быстрогодействия, помехозащищенность, стоимость однократной реализации алгоритма, коэффициент загрузки аппаратуры.

4. Основные показатели эффективности реализации параллельных алгоритмов

Рассмотрим основные показатели эффективности реализации параллельных алгоритмов [5].

Математическое ожидание времени реализации множества P операторов произвольного алгоритма определяется выражением (1)

$$(1) \quad T(P) = \sum_{\xi=1}^w P_{\xi} T_{\xi}, (c),$$

где w – число ветвей в алгоритме, P_{ξ} – вероятность реализации ξ -й ветви; T_{ξ} – время реализации ξ -й ветви параллельного алгоритма, определяемое по формуле

$$T_{\xi} = \max_{P_j \in P(\xi)} (t_j^H + t_j), (c),$$

где $P(\xi)$ – множество операторов ξ -й ветви, t_j^H и t_j – момент начала и относительная временная глубина оператора $P_j \in P(\xi)$.

Рассмотренное соотношение обеспечивает возможность оценок максимального времени реализации алгоритма. Для этого достаточно значение P_{ξ} положить равным единице для соответствующей ветви (и равным нулю для остальных ветвей).

При проектировании параллельных аппаратных и программных средств существенное практическое значение имеет знание дисперсии времени реализации T_k ветвей различной длительности параллельного алгоритма. Этот показатель определяется следующим соотношением

$$(2) \quad D = \sum_{\xi} (T_{\xi} - T(P))^2, (c2), \xi = 1, \dots, w.$$

В ряде случаев необходимо уметь оценивать снижение временных затрат на выполнение алгоритмов (укорение) за счет перехода к их параллельной реализации. Это обеспечивается показателем DT снижения временных затрат

$$(3) \quad DT = \frac{T_{noc}(P)}{T_{нар}(P)}, \text{ (раз).}$$

В соотношении (3) $T_{noc}(P)$ и $T_{нар}(P)$ – среднее время соответственно последовательной и параллельной реализации алгоритма.

Среднее быстродействие при реализации параллельных алгоритмов определяется отношением математического ожидания числа операторов, выполняемых при реализации алгоритма с учетом различной вероятности выполнения отдельных ветвей, к математическому ожиданию времени реализации алгоритма:

$$(4) \quad B(P) = \frac{\sum_{\xi=1}^W p_{\xi} \sum_{t^H=1}^{T^{\xi}} H^{\xi}(t^H)}{\sum_{\xi=1}^W p_{\xi} T_{\xi}}, \text{ (опер/с),}$$

В соотношении (4) $H^{\xi}(t^H)$ – количество операторов ξ -й ветви алгоритма, реализация которых начинается в момент времени t^H .

Для систем реального масштаба времени ввиду их сложности большое значение имеют также показатели, характеризующие экономическую сторону применения цифровых вычислительных средств для реализации алгоритмов.

Таковыми показателями являются эквивалентная вентиляционная сложность аппаратных средств и показатель использования (загрузки) аппаратных средств при реализации различных алгоритмов. Для получения расчетных соотношений для этих показателей введем следующие обозначения:

ν – количество различных типов компонентов (выполняющих каждый определенную операцию над данными конкретной разрядности), используемых при параллельной реализации алгоритма;

$\Theta = \{\Theta_{\eta}\}$ – множество различных типов Θ_{η} компонентов, которые входят в состав параллельного устройства;

$N = \{n_{\eta}\} (\eta = \overline{1, \nu})$ – множество количеств компонентов типа Θ_{η} ;

$K = \{K_{\eta\xi\delta}\}$ – множество количеств $K_{\eta\xi\delta}$ операторов $P_j \in P$, которые выполняются компонентом типа Θ_{η} , имеющим номер δ , в процессе реализации ξ -й ветви алгоритма;

$T^0 = \{t_{\eta}^0\}(\eta = \overline{1, \nu})$ – множество значений времени выполнения операторов различных типов $\Theta_{\eta} \in \Theta$ компонентами соответствующих типов;

$Q = \{q_{\eta}\}(\eta = \overline{1, \nu})$ – множество значений эквивалентной вентиляющей сложности компонентов типов $\Theta_{\eta} \in \Theta$, входящих в состав параллельного устройства, проектируемого для аппаратной или программной реализации конкретной параллельной модели алгоритма.

С учетом введенных обозначений эквивалентную вентиляющую сложность конкретного параллельного устройства можно определить следующим соотношением

$$(5) \quad Q = \sum_{\eta=1}^{\nu} q_{\eta} \cdot n_{\eta}, \text{ (экв. вент).}$$

Среднее значение показателя загрузки компонентов каждого типа $\Theta_{\eta} \in \Theta$ можно рассчитать в соответствии со следующим выражением

$$(6) \quad S(P_{\eta}) = \frac{\sum_{\xi=1}^W (t_{\eta}^0 \cdot K_{\eta\xi} \cdot p_{\xi})}{n_{\eta} \cdot T}, \text{ (раз),}$$

где $P_{\eta} \subseteq P$ – подмножество операторов $P_j \in P$, имеющих тип Θ_{η} ,

$$K_{\eta\xi} = \sum_{\delta=1}^{n_{\eta}} K_{\eta\xi\delta}.$$

Тогда среднее значение полного показателя загрузки всех компонентов, входящих в состав параллельного устройства, можно определить следующим соотношением

$$(7) \quad S(P) = \sum_{\eta=1}^{\nu} S(P_{\eta}) = \frac{1}{T} \sum_{\eta=1}^{\nu} \frac{1}{n_{\eta}} \sum_{\xi=1}^W (t_{\eta}^0 \cdot K_{\eta\xi} \cdot p_{\xi}), \text{ (раз).}$$

Различным параллельным моделям алгоритмов соответствуют индивидуальные аппаратные и/или программные реализации, характеризующиеся различной сложностью. Необходимость отыскания на начальном этапе автоматического проектирования оптимальных вариантов аппаратных и программных решений проектной задачи делает целесооб-

разным применение в качестве частных показателей эффективности среднего значения Q_{cp} и дисперсии DQ сложностей Q_k ($k = 1 \dots m$) параллельных моделей, реализующих конкретный алгоритм и использующих конкретный состав методов параллельной обработки данных

$$(8) \quad Q_{cp} = \sum_{k=1}^m (Q_k),$$

$$(9) \quad DQ = \sum_{k=1}^m (Q_k - Q_{cp})^2,$$

где Q_k – прогнозируемая сложность реализации k -й параллельной модели, m – количество различных параллельных моделей рассматриваемого алгоритма.

Важное прикладное значение имеет задача определения числа NM процессоров, применение которых для параллельной реализации алгоритма обеспечивает достаточно большое снижение временных затрат (либо заданное снижение временных затрат) за счет параллельного выполнения алгоритма, с одной стороны, и возможно более высокое значение коэффициента использования оборудования (либо достижение заданного значения этого коэффициента), с другой. В таких случаях можно в качестве производного показателя использовать мультипликативный показатель (соотношения 7, 8) или аддитивный показатель (соотношение 9) эффективности распараллеливания

$$(10) \quad R(P) = \frac{DT(NM)}{DT_{\max}(NM)} \cdot \frac{S(NM)}{S_{\max}(NM)}, \text{ (раз)}.$$

Принимая во внимание, что максимальное снижение временных затрат при использовании NM процессоров не может превышать NM , то есть $DT_{\max}(NM) \leq NM$, а максимальное значение показателя загрузки оборудования $S_{\max}(NM) \leq 1$, соотношение (7) может быть преобразовано к следующему виду:

$$(11) \quad R(P) = \frac{DT(NM)}{NM} \cdot S(NM), \text{ (раз)},$$

$$(12) \quad R(P) = K_T \frac{DT(NM)}{NM} + K_S S(NM), \text{ (раз).}$$

Отметим, что в соотношении (12) K_T и K_S являются весовыми коэффициентами, определяющими «пользовательскую» важность учета в эффективности распараллеливания величины сокращения времени реализации алгоритма ($K_T \leq 1$) и степени загрузки оборудования параллельным алгоритмом ($K_S \leq 1$).

Для параллельных аппаратных средств систем обработки данных и систем управления, работающих в реальном масштабе времени, одним из важнейших требований является требование ввода данных извне и выдачи результатов обработки данных с необходимой тактовой частотой F_3 или с заданным тактом TT_3 . Значения F_3 (или $TT_3 = 1 / F_3$) определяются динамикой реальных физических процессов. Значения тактового интервала (такта) TT для цифрового устройства, содержащего в своем составе компоненты типа $\Theta_\eta \in \Theta$ с известными временными характеристиками

$T^0 = \{t_\eta^0\}$, $\eta = 1..v$, определяется следующим образом:

$$(13) \quad TT = \max_{\eta=1..v} (t_\eta^0) + 2dt^\Phi, \text{ (с),}$$

где $dt^\Phi = \max(t_{3am}^0, t_{um}^0)$, а t_{3am}^0 и t_{um}^0 – соответственно величина цикла записи и величина цикла считывания данных из компонентов памяти, выполняющих роль «фиксаторов» промежуточных результатов и используемых для организации конвейерной и декомпозиционной обработки данных.

При создании систем обработки информации и управления, работающих в реальном масштабе времени, а также при решении сложных научно-технических задач в самых различных прикладных областях в качестве одного из важнейших требований выступает время выполнения отдельных критически сложных алгоритмов или комплексов алгоритмов ИУС. Метод синтеза максимально параллельных моделей (МПМ) обеспечивает синтез временных глобально-параллельных моделей (с детализацией до операций/функций) при отсутствии ограничений на доступный ресурс, имеющих минимальное возможное время выполнения алгоритмов.

Обобщенный алгоритм синтеза временных максимально параллельных моделей алгоритмов, использующих совмещение независимых операций, представлен на рис. 4.

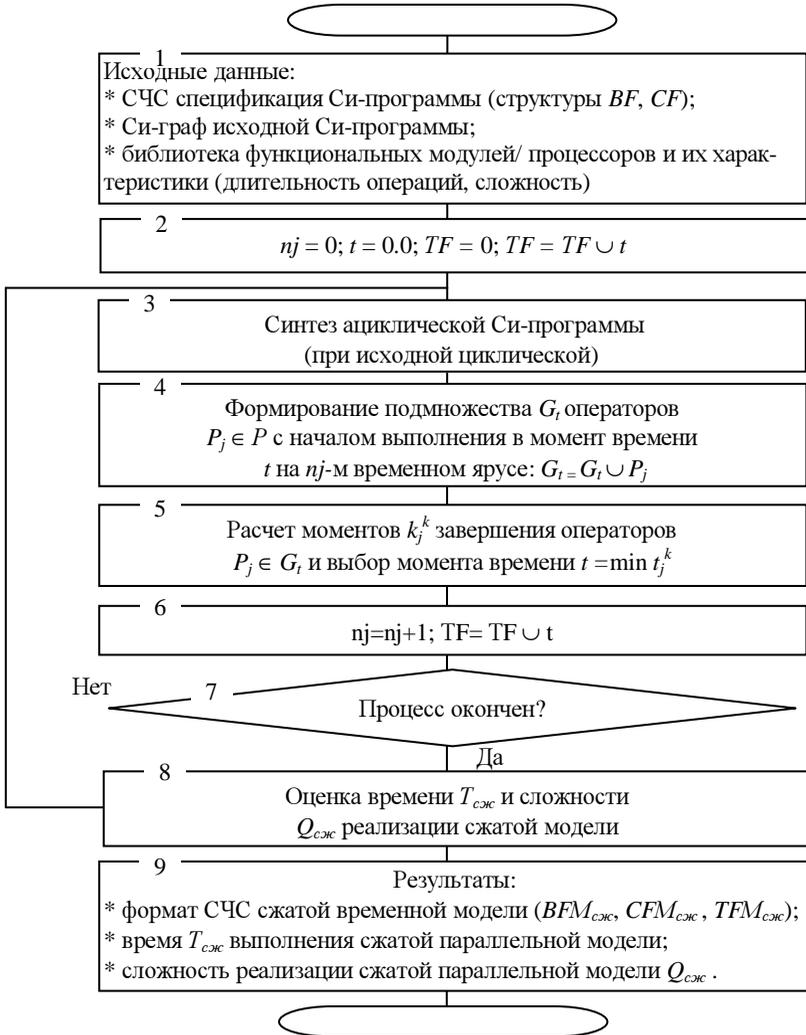


Рис. 4. Этапы формального синтеза максимально параллельных временных моделей алгоритмов

Для иллюстрации результатов синтеза времяпараметризованной максимально параллельной модели используем задачу решения системы двух линейных уравнений по алгоритму Крамера [6]. Исходный текст Си-программы представляет рис. 5, длительности выполнения операций различных типов (здесь и далее – в количестве тактов процессора *R10000 MIPS SUN Microsystems* представляет табл. 1. Результаты синтеза Си-графа исходной Си-программы показаны на рис. 6.

```

#include <stdio.h>
void main(void )
{
    int a,b,c,d,e,f;
    int m,x,y;
    scanf("%d %d %d\n",&a,&b,&c);
    scanf("%d %d %d\n",&d,&e,&f);
    m = (b * d) - (a * e);
    x = ((b * f) - (c * e)) / m;
    printf("%3d",x);
    y = ((a * f) - (c * d)) / m;
    printf("%3d",y);
}
    
```

Рис. 5. Исходная Си-программа решения системы двух линейных уравнений

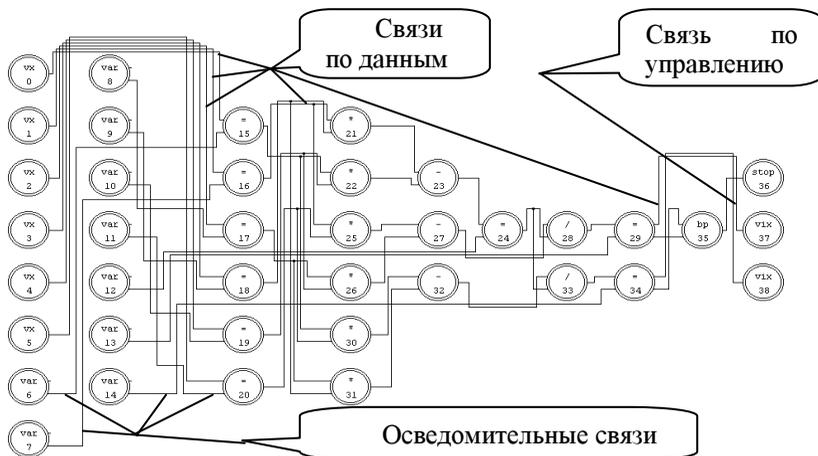


Рис. 6. Си-граф программы решения системы двух линейных уравнений

ТАБЛИЦА 1. Длительность t_j^0 выполнения операций различных типов (такты)

typ	vx	+, -	=	*	%	/	vix	bp	stop	cont
t_j^0	1.00	1.00	2.00	10.00	35.00	35.00	1.00	1.00	1.00	1.00

Максимально параллельная (сжатая) временная модель, использующая метод совмещения независимых операций, представлена на рис. 7. Отметим, что особенностью сжатой временной параллельной модели (рис. 7) является расположение на каждом временном ярусе с номерами $n_j = 0, 1, \dots, 8$ всех операторов, для каждого из которых имеются вычисленные (к данному ярусу с конкретным значением $t(n_j)$ ярусного времени) значения всех операндов. Структуры семантико-числовой спецификации состава (BF) и связей операторов (CF) Си-программы и ее максимально параллельной временной модели представлены табл. 2, 3 и структура TF времен t_j^H начала выполнения операторов (табл. 4).

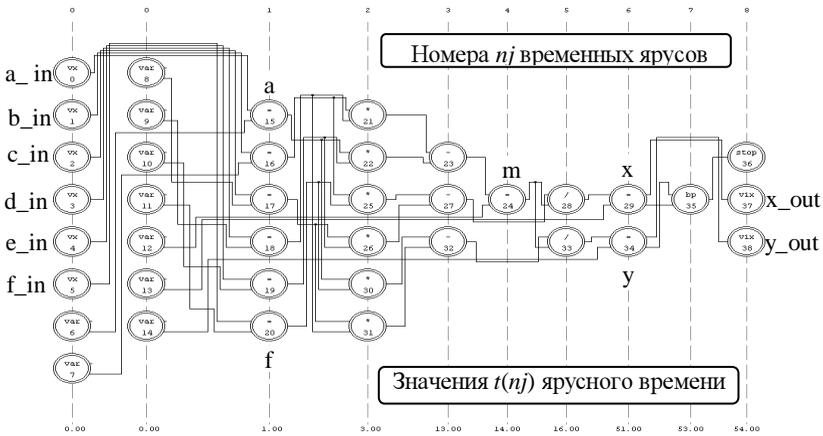


Рис. 7. Временная максимально параллельная (сжатая) модель алгоритма решения системы двух линейных уравнений (совмещение операций)

ТАБЛИЦА 2. Базовая структура BF операторов Си-программы

N	МЕТ	ТУР	NSJ	SJD	BJ	NWJ	WJD	MP1	MP2	VH	VIH	RES
0	0	58	-1	0	0	0	1	0	0	0	1	a_in
1	0	58	-1	0	0	1	1	0	0	0	1	b_in
2	0	58	-1	0	0	2	1	0	0	0	1	c_in
3	0	58	-1	0	0	3	1	0	0	0	1	d_in
4	0	58	-1	0	0	4	1	0	0	0	1	e_in
5	0	58	-1	0	0	5	1	0	0	0	1	f_in
6	0	47	-1	0	0	6	1	0	0	0	2	a
7	0	47	-1	0	0	7	1	0	0	0	2	b
8	0	47	-1	0	0	8	1	0	0	0	2	c
9	0	47	-1	0	0	9	1	0	0	0	2	d

N	МЕТ	ТУР	NSJ	SJD	BJ	NWJ	WJD	MP1	MP2	VH	VH	RES
10	0	47	-1	0	0	10	1	0	0	0	2	e
11	0	47	-1	0	0	11	1	0	0	0	2	f
12	0	47	-1	0	0	12	1	0	0	0	2	m
13	0	47	-1	0	0	13	1	0	0	0	2	x
14	0	47	-1	0	0	14	1	0	0	0	2	y
15	0	12	0	2	0	15	2	0	0	2	1	=
16	0	12	2	2	0	17	2	0	0	2	1	=
17	0	12	4	2	0	19	2	0	0	2	1	=
18	0	12	6	2	0	21	2	0	0	2	1	=
19	0	12	8	2	0	23	2	0	0	2	1	=
20	0	12	10	2	0	25	2	0	0	2	1	=
21	0	3	12	2	0	27	1	0	0	2	1	*
22	0	3	14	2	0	28	1	0	0	2	1	*
23	0	2	16	2	0	29	1	0	0	2	1	-
24	0	12	18	2	0	30	2	0	0	2	1	=
25	0	3	20	2	0	32	1	0	0	2	1	*
26	0	3	22	2	0	33	1	0	0	2	1	*
27	0	2	24	2	0	34	1	0	0	2	1	-
28	0	4	26	2	0	35	1	0	0	2	1	/
29	0	12	28	2	0	36	2	0	0	2	2	=
30	0	3	30	2	0	38	1	0	0	2	1	*
31	0	3	32	2	0	39	1	0	0	2	1	*
32	0	2	34	2	0	40	1	0	0	2	1	-
33	0	4	36	2	0	41	1	0	0	2	1	/
34	0	12	38	2	0	42	2	0	0	2	2	=
35	0	50	40	2	0	44	1	80	0	2	1	bp
36	980	49	42	1	1	-1	0	0	0	1	0	stop
37	0	48	43	1	1	-1	0	0	0	1	0	x_out
38	0	48	44	1	1	-1	0	0	0	1	0	y_out

ТАБЛИЦА 3. Структура CF связей операторов Си-программы

N	JSD	SPJD	SNWH	SNWHO	JWD	WPJD	WNWHO	WNWH
0	1	0	0	0	-1	15	0	0
1	-1	6	1	1	-1	16	0	0
2	3	1	0	0	-1	17	0	0
3	-1	7	1	1	-1	18	0	0
4	5	2	0	0	-1	19	0	0
5	-1	8	1	1	-1	20	0	0
6	7	3	0	0	-1	15	1	1
7	-1	9	1	1	-1	16	1	1
8	9	4	0	0	-1	17	1	1
9	-1	10	1	1	-1	18	1	1

Z	JSD	SPJD	SNWIH	SNWHO	JWD	WPJD	WNWHO	WNWIH
10	11	5	0	0	-1	19	1	1
11	-1	11	1	1	-1	20	1	1
12	13	16	0	0	-1	24	1	1
13	-1	18	0	1	-1	29	1	1
14	15	15	0	0	-1	34	1	1
15	-1	19	0	1	16	22	0	0
16	17	21	0	0	-1	30	0	0
17	-1	22	0	1	18	21	0	0
18	19	12	1	1	-1	25	0	0
19	-1	23	0	0	20	26	0	0
20	21	16	0	0	-1	31	0	0
21	-1	20	0	1	22	21	1	0
22	23	17	0	0	-1	31	1	0
23	-1	19	0	1	24	22	1	0
24	25	25	0	0	-1	26	1	0
25	-1	26	0	1	26	25	1	0
26	27	27	0	0	-1	30	1	0
27	-1	24	0	1	-1	23	0	0
28	29	13	1	1	-1	23	1	0
29	-1	28	0	0	-1	24	0	0
30	31	15	0	0	31	28	1	0
31	-1	20	0	1	-1	33	1	0
32	33	17	0	0	-1	27	0	0
33	-1	18	0	1	-1	27	1	0
34	35	30	0	0	-1	28	0	0
35	-1	31	0	1	-1	29	0	0
36	37	32	0	0	37	35	1	1
37	-1	24	0	1	-1	37	0	0
38	39	14	1	1	-1	32	0	0
39	-1	33	0	0	-1	32	1	0
40	41	34	1	0	-1	33	0	0
41	-1	29	1	1	-1	34	0	0
42	-1	35	0	0	43	35	0	1
43	-1	29	0	0	-1	38	0	0
44	-1	34	0	0	-1	36	0	0

ТАБЛИЦА 4. Структура TF времен t_{jn} начала выполнения операторов P_j «жато» временной параллельной модели задачи

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
t_j^n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
j	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	
t_j^n	1	3	3	13	14	3	3	13	16	51	3	3	13	16	51	53	54	54	54	

Во многих системах обработки информации и управления, работающих в реальном масштабе времени, в качестве существенного параметра (наряду со временем) может выступать сложность/стоимость ап-

паратной или программной реализации параллельной модели. Технология ПМР поддерживает синтез временных глобально-параллельных моделей алгоритмов при наличии ограничений на доступный ресурс. Обобщенный алгоритм формального синтеза временных глобально-параллельных моделей задач при ограничении доступного ресурса, использующих совмещение независимых операций, представляет рис. 8.

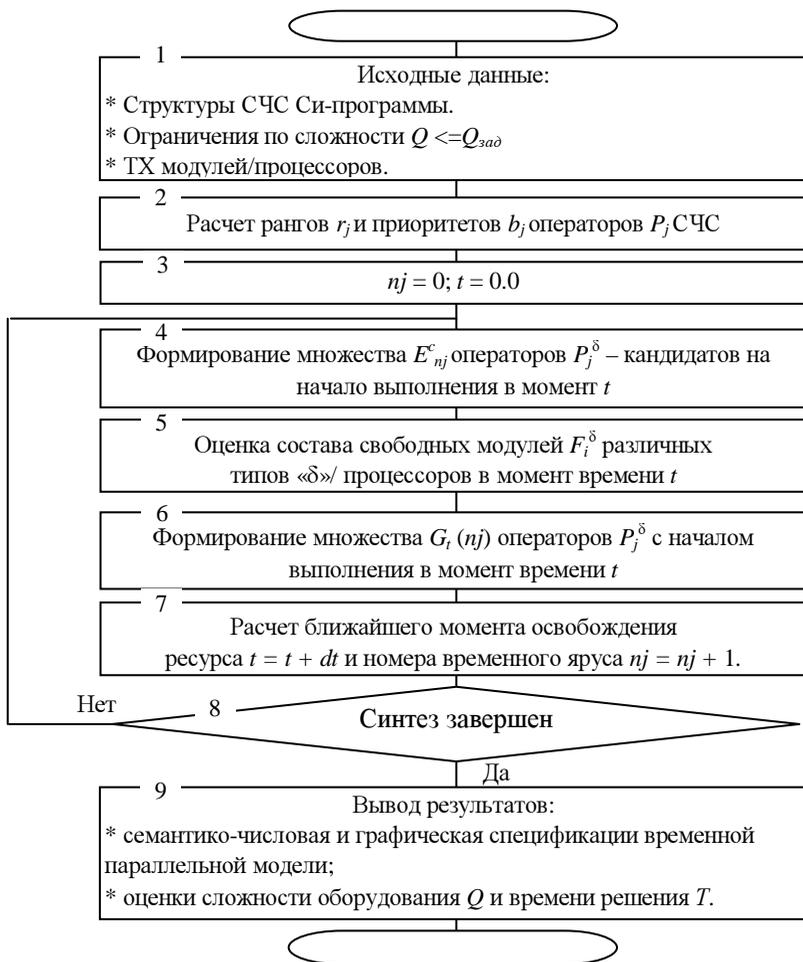


Рис. 8. Этапы формального синтеза временных глобально-параллельных моделей задач при ограничении доступного ресурса

Используем для иллюстрации результатов синтеза задачу решения системы двух линейных уравнений по алгоритму Крамера. Исходный текст Си-программы представляет рис. 5, длительности выполнения операций различных типов представляет табл. 1. Си-граф Си-программы показан на рис. 6. Структуры СЧС представляют табл. 2, табл. 3. Результат автоматического синтеза временной параллельной модели задачи при использовании двух процессоров представлен на рис. 9.

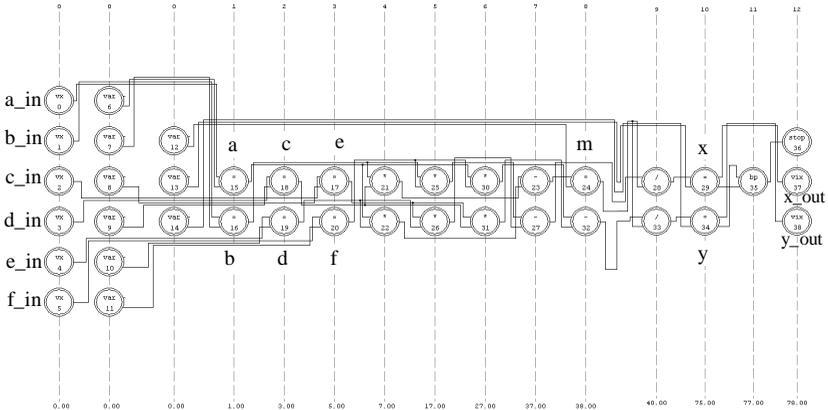


Рис. 9. Временная параллельная модель решения системы линейных уравнений (совмещение операций, $NM = 2$)

Для оценки показателей эффективности параллельных моделей с различной шириной параллельного процесса (зависимостей времени решения $T = f_1(NM)$ от NM и загрузки $S = f_2(NM)$ оборудования) синтезированы модели при изменении количества процессоров в диапазоне $NM = 1, 2, \dots, 6$. Зависимости времени $T = f_1(NM)$ решения и загрузки оборудования $S = f_2(NM)$ от NM представляют рис. 10 и рис. 11.

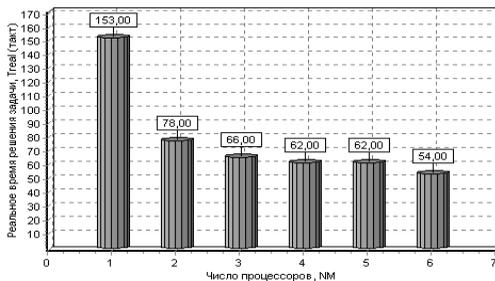
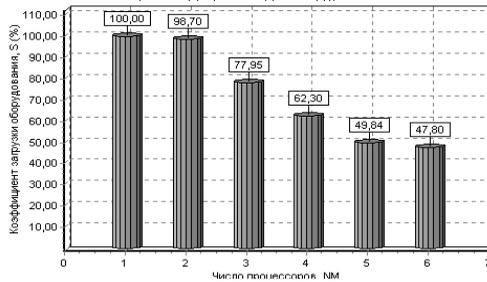


Рис. 10. Зависимости времени $T = f_1(NM)$ решения от NM

Рис. 11. Зависимости загрузки $S=f_2(NM)$ оборудования от NM

Выводы

- (1) Адаптивная технология автоматического синтеза архитектурно- и проблемно-ориентированных мультипараллельных времяпараметризованных моделей задач является первым этапом синтеза параллельных аппаратно-программных средств, удовлетворяющих заданным требованиям.
- (2) Методы формального синтеза времяпараметризованных мультипараллельных моделей процессов решения задач обеспечивают учет следующих групп факторов, оказывающих влияние на эффективность программных и аппаратных средств ВС: особенности различных классов задач; возможность различной прикладной направленности (синтез параллельных программ, синтез параллельных аппаратных средств); состав методов параллельной обработки данных (метод совмещения независимых операций, метод конвейерной обработки, кодово-матричный метод, декомпозиционный метод и метод смеси алгоритмов); возможные принципы управления параллельными вычислительными процессами (управление потоком команд; управление потоком данных; временное управление на каждом такте); особенности принципов построения, архитектуры, конфигурации и временных характеристик различных типов процессоров и известных параллельных ВС, а также перспективных АСВС; возможность семантико-числовой обработки данных с учетом единиц измерения физических величин; необходимость удовлетворения систем требований и ограничений пользователей (время выполнения программы, тактовая частота обработки данных, производительность, сложность/стоимость и др.).
- (3) Конкретные примеры синтеза мультипараллельных моделей процессов и программные реализации основных этапов подтверждают работоспособность и эффективность разработанных методов. Следует отметить, что в доступной научно-технической литературе аналогичные результаты авторами не обнаружены.

Список литературы

- [1] Воеводин В. В., Воеводин Вл. В. *Параллельные вычисления* // СПб.: БХВ Петербург, 2002. — 608 с.
- [2] Лацис А. *Как построить и использовать суперкомпьютер* // М.: Бестселлер, 2003. — 240 с.
- [3] Немнюгин С. А., Стесик О. Л. *Параллельное программирование для многопроцессорных вычислительных систем* // СПб.: БХВ Петербург, 2002. — 400 с.
- [4] Поляков Г. А., Скляр В. В., Толстолужский Д. А. и др. *Проблемы многоверсионного проектирования высоконадежных параллельных программных средств для систем управления критическими технологиями и объектами* // *Радиоелектронні і комп'ютерні системи*, 2006. №7(19). с. 7–16.
- [5] Поляков Г. А., Шматков С. И., Толстолужская Е. Г., Толстолужский Д. А. *Синтез и анализ параллельных процессов в адаптивных времяпараметризованных вычислительных системах*. — Харьков: изд-во ХНУ им. В.Н. Каразина, 2012. — 672 с.
- [6] Г. Корн, Т. Корн *Справочник по математике (для научных работников и инженеров)* // пер. с англ. И. Г. Арамановича, А. М. Березмана, И. А. Вайнштейна и др. под ред. И.Г. Арамановича / М.: Наука, 1974. — 831 с.



Геннадий Алексеевич Поляков

Доктор технических наук, профессор, академик академии наук прикладной радиоэлектроники, место работы — профессор кафедры математического и программного обеспечения информационных систем факультета компьютерных наук и телекоммуникаций Белгородского государственного национального исследовательского университета

e-mail:

tda_ua@pochtamt.ru



Елена Геннадиевна Толстолужская

Кандидат технических наук, старший научный сотрудник, докторант факультета компьютерных наук Харьковского национального университета имени В.Н. Каразина, место работы — доцент кафедры теоретической и прикладной системотехники факультета компьютерных наук Харьковского национального университета имени В.Н. Каразина

e-mail: i-ua@i.ua

Образец ссылки на публикацию:

Г. А. Поляков, Е. Г. Толстолужская. Адаптивная технология автоматического синтеза архитектурно- и проблемно-ориентированных мультимедийных систем // *Радиоелектронні і комп'ютерні системи*, 2006. №7(19). с. 7–16.

типараллельных времяпараметризованных моделей задач // Программные системы: теория и приложения: электрон. научн. журн. 2013. Т. 4, № 3(17), с. ??-??.

URL:

<http://psta.pstiras.ru/read/???>

G.A. Polyakov, E.G. Tolstoluzhskaya. Adaptive technology of automatic synthesis of the architecture- and problem- oriented multiparallel timeparameterized models of tasks.

ABSTRACT. Adaptive technology of automatic synthesis of multiparallel time parameterized models of tasks is described in the article, the basic formal synthesis stages of maximally parallel temporal algorithm models and time parameterized global-parallel tasks models at limitation of accessible resource are brought. The main stages are illustrated on example of time parameterized parallel resolving of two linear equations system on the algorithm of Cramer task models synthesis.

Key Words and Phrases: time parameterized parallel model, numeric specification of program, graph of program, semantic-numeric specification, C-graph.