

УДК 681.31

Г.А. ПОЛЯКОВ¹, Е.Г. ТОЛСТОЛУЖСКАЯ², Д.А. ТОЛСТОЛУЖСКИЙ³¹Академия наук прикладной радиоэлектроники, Украина²Харьковский университет Воздушных Сил, Украина³Харьковский национальный университет им. В.Н. Каразина, Украина

ОЦЕНКА ДИАПАЗОНА ИЗМЕНЕНИЯ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНЫХ ВРЕМЕННЫХ МОДЕЛЕЙ АЛГОРИТМОВ

Раскрываются этапы решения задачи оперативной оценки диапазонов изменения эффективности моделей алгоритмов, заключающиеся в синтезе моделей с предельно минимальным временем выполнения алгоритмов, синтезе моделей с предельно минимальным значением сложности аппаратного/программного ресурса и расчете диапазонов изменения эффективности моделей алгоритмов. Приводится иллюстрирующий пример.

параллельный алгоритм, временные модели алгоритмов, временное моделирование, оценка времени реализации алгоритма

Введение

Постановка проблемы. При проектировании параллельных аппаратных средств и параллельных программ встает вопрос о потенциальной распараллеливаемости алгоритмов, определяемой только составом операторов и связями между ними по данным и по управлению. Актуальность вопроса обусловлена тем, что решение многих задач может выполняться на основе применения различных численных методов и соответствующих алгоритмов, имеющих различные потенциальные возможности распараллеливания [1 – 4].

В то же время существует достаточно широкий класс систем обработки информации и управления, предъявляющих жесткие требования к количеству оборудования (сложности) применяемых вычислительных средств. При проектировании для таких систем аппаратных или программно управляемых средств возникает необходимость в синтезе последовательных временных моделей алгоритмов, аппаратная или программная реализация которых обеспечивает удовлетворение требований по сложности.

Это делает крайне актуальной разработку мето-

дов и средств оперативной автоматической оценки диапазона изменения эффективности моделей алгоритмов с потенциально минимальным временем выполнения (и оценкой максимально необходимой сложности аппаратного/программного ресурса) и/или с минимальной сложностью (и оценкой достижимого при этом времени реализации алгоритмов).

Анализ литературы. Решение широкого круга прикладных задач, связанных с проектированием аппаратно-программного обеспечения систем обработки информации и управления, требует оперативного прогнозирования предельных возможностей по минимизации времени реализации алгоритмов или минимизации сложности их аппаратных/программных реализаций. Вопросы временного моделирования и построения временных моделей работы конкретных устройств при проектировании рассматриваются в ряде работ, посвященных САПР [5 – 7]. Вопросы оценки времени реализации проектов и проведения экспериментов изложены в работах [8, 9]. Анализ показывает, что известные решения не обеспечивают оперативности оценки диапазона изменения эффективности параллельных вре-

менных моделей алгоритмов. Это обусловлено большой длительностью временного моделирования, ориентацией на оценку временных параметров конкретных разработанных человеком цифровых устройств, сетевых графиков и статистических экспериментов, невозможностью учета различных методов параллелизма.

Цели статьи. Целью статьи является описание средств оперативной оценки диапазонов изменения эффективности моделей алгоритмов. Решение этой задачи включает:

- синтез моделей с предельно минимальным временем выполнения алгоритмов и оценкой максимально необходимой сложности аппаратного/программного ресурса;
- синтез моделей с предельно минимальным значением сложности аппаратного/программного ресурса и оценкой достижимого при этом времени реализации алгоритмов;
- расчет диапазонов изменения эффективности моделей алгоритмов.

Исследования и результаты

1. Синтез максимально параллельных (сжатых) временных моделей алгоритмов.

Исходные данные задачи синтеза сжатой временной модели алгоритма:

- Си – программа, представляющая формальное описание рассматриваемого алгоритма;
- числовая спецификация Си – программы в формате СВМ (представленная базовым файлом ВФ и файлом связей СФ на операционном уровне и соответствующими файлами ВФЕ и СФЕ на уровне естественных частей Си – программы) [10];
- значения времени выполнения операций/функций языка Си, используемых в Си-программе.

Требуется:

- синтезировать временную модель, соответствующую максимально параллельной реализации алгоритма (представленную базовым файлом ВФ, файлом связей СФ и временным файлом ТФ) на операционном уровне;

- получить оценки времени выполнения $T_{сж}$ сжатой временной модели алгоритма.

Решение задачи синтеза сжатой временной модели алгоритма включает следующие этапы:

- формирование множественных временных операторов $G_t = P(n_j, t_j^H)$ операторов $P_j \in P$, выполнение которых начинается на временном ярусе с номером n_j ($n_j = 0, 1, 2, \dots, k_j$; k_j – количество временных ярусов процесса реализации сжатой модели алгоритма) в момент t дискретного модельного времени при отсутствии ограничений на количество оборудования;
- проверку окончания распараллеливания алгоритма;
- оценку минимально возможного времени реализации алгоритма $T_{сж}$.

Рассмотрим семантику и формализацию основных этапов решения рассматриваемой задачи.

Этап 1 обеспечивает формирование множества G_t операторов $P_j \in P$ формата СВМ Си – программы, выполнение которых компонентами проектируемого цифрового устройства (при аппаратных приложениях модели) или процессорами (при программных приложениях модели) начинается на временном ярусе с номером n_j в момент времени t (с учетом отсутствия ограничений на количество оборудования).

Множество G_t при $n_j = 0$ и $t = 0,0$ состоит из операторов – исходных данных, операторов – входов данных и операторов – констант, а при $n_j > 0$ и $t >= 1,0$ G_t включает внутренние и выходные операторы алгоритма, для каждого из которых выполняются следующие условия:

а) все сопряженные (входные) для P_j операторы $P_i \in S_j$ начинают выполняться в предшествующие моменты времени $t_i^H < t$;

б) реализация всех сопряженных для P_j операторов $P_i \in S_j$ заканчивается к рассматриваемому моменту времени t (то есть $t_i^k \leq t$, напомним, что $t_i^k = t_i^H + t_i^0$).

Отметим, что значения времени t_j и t_i численно равны времени реализации соответствующих операций/функций формата СВМ алгоритма, выраженному во временных единицах измерения (с, мс, мкс или нс – при синтезе параллельных моделей в интересах проектирования аппаратных средств) или в количестве процессорных тактов (при синтезе параллельных моделей алгоритмов в интересах проектирования параллельных программ).

Логика формирования множества G_t определяется следующими соотношениями (1) – (3):

$$G_t = \begin{cases} \bigcup_j P_j & \text{для } S(P_j)=0 \text{ (при } nj=0); \\ \bigcup_j P_j & \text{для } P_j \text{ (при } nj > 0) \text{ и истинности (2,3),} \end{cases} \quad (1)$$

$$P_j \in \bigcup_{P_\xi \in G_{nj-1}^1} W(P_\xi) \text{ при } S(P_j) \subseteq G_{nj-1}^1; \quad (2)$$

$$t_\xi^H + t_\xi^0 \leq t(nj) \text{ для всех } P_\xi \in S(P_j). \quad (3)$$

Здесь G_{nj-1}^1 – множество операторов P_j , реализация которых была начата на ярусах с номерами от 1 до $nj - 1$.

Этап 2 проверяет завершение процесса синтеза сжатой параллельной модели алгоритма. Признаком завершения является назначение на выполнение всех операторов P_j формата СВМ Си – программы алгоритма. Этому событию соответствует выполнение следующего условия:

$$\left(\bigcup_{nj=0}^{kj} G_t \right) \setminus P = \emptyset. \quad (4)$$

Этап 3 фиксирует значение времени $T_{сж}$ выпол-

нения сжатой параллельной модели алгоритма в качестве потенциального минимума времени параллельной реализации алгоритма $T_{сж} = t_{n-1}^H + t_{n-1}^0$, $n - 1$ определяемого моментом окончания выполнения его выходного оператора P_{n-1} .

II. Синтез последовательных временных моделей алгоритмов.

Исходные данные автоматического проектирования последовательных временных моделей алгоритмов:

- Си – программа, представляющая формальное описание рассматриваемого алгоритма;
- числовая спецификация Си – программы в формате СВМ (представленная базовым файлом ВФ и файлом связей СФ на операционном уровне);
- библиотека ЛВ функциональных модулей F_i (при проектировании аппаратно реализуемого приложения) или библиотека характеристик процессоров (при проектировании программно реализуемого приложения);
- значения времени t_j^0 выполнения операций/функций языка Си, используемых в формате СВМ Си - программы: во временных единицах измерения (с, мс, мкс или нс) – при синтезе моделей в интересах проектирования аппаратно реализуемого приложения, в количестве процессорных тактов - при синтезе моделей алгоритмов в интересах проектирования программно реализуемого приложения;
- задано требование минимальной сложности Q аппаратной/программной реализации алгоритма.

Требуется синтезировать последовательную временную модель реализации алгоритма, обеспечивающую удовлетворение требования минимальной Q сложности аппаратной/ программной реализации алгоритма.

Основные этапы решения задачи представлены на рис. 1.

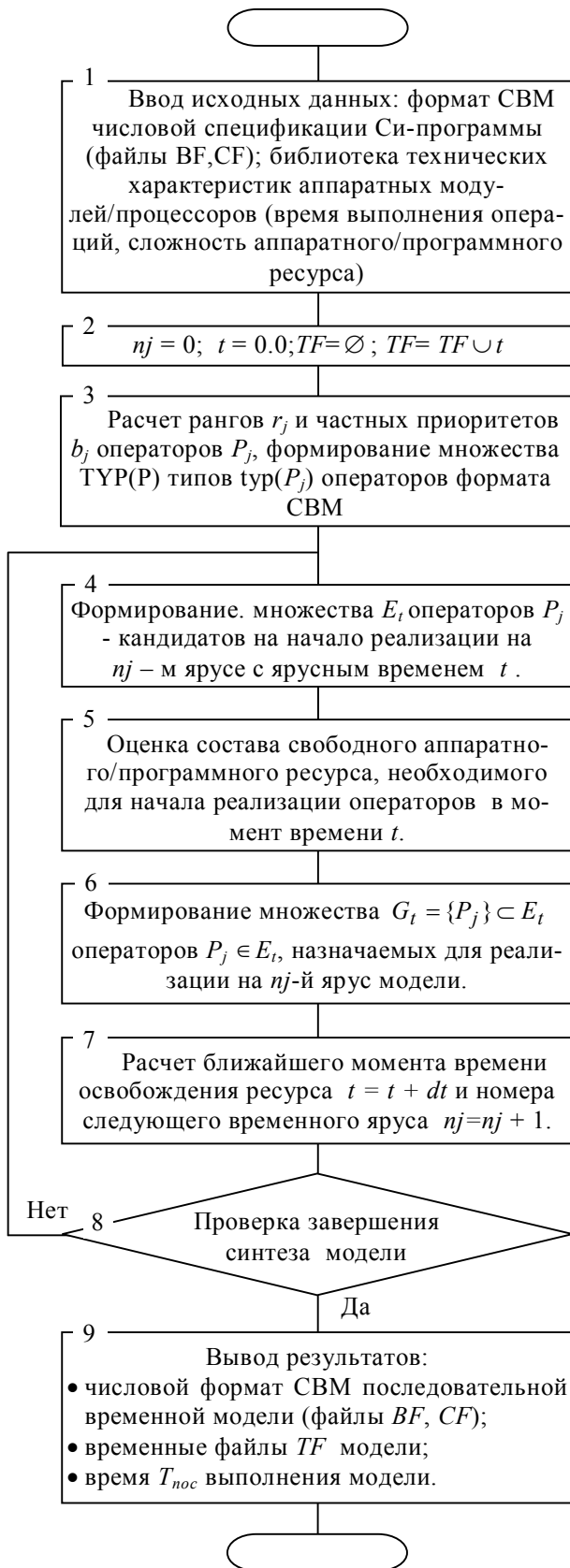


Рис. 1. Алгоритм автоматического проектирования последовательных временных моделей с минимальной сложностью аппаратной/программной реализации

Рассмотрим формализацию основных этапов проектирования.

Процедура 1 (символ 3) выполняет расчет значений рангов r_j операторов P_j и частных приоритетов b_j , определяющих важность оператора P_j по отношению к другим операторам P_i из множества P (и тем самым очередность его рассмотрения при решении задачи выделения аппаратного/ программно ресурса для его реализации) [10].

Процедура 2 (символ 4). Формирует множество $E_{t(nj)}$ операторов-кандидатов P_j формата СВМ Си – программы, реализация которых могла бы быть начата на временном ярусе с номером nj в момент времени t при отсутствии ограничений на сложность аппаратного/программного ресурса. Логика формирования множества $E_{t(nj)}$ определяется следующими соотношениями

$$E_{t(nj)} = R_{nj-1} \cup K_{nj}, \quad (5)$$

где R_{nj-1} – множество операторов P_j , реализация которых не была начата на $(nj - 1)$ -м ярусе в связи с отсутствием необходимых свободных компонентов; K_{nj} – множество операторов, выполнение которых может быть начато на nj -м ярусе в связи с наличием всех необходимых для них данных (т.е. с учетом ранее выполненных операторов и информационно-управляющих связей между операторами $P_j \in P$).

$$R_{nj-1} = \begin{cases} 0 & \text{при } nj = 0; \\ E_{t(nj-1)} \setminus G_{nj-1} & \text{при } nj > 0, \end{cases} \quad (6)$$

$$K_{nj} = \begin{cases} \bigcup_j P_j & \text{для } S(P_j) \neq 0 \text{ (при } nj = 0); \\ \bigcup_j P_j & \text{для } P_j \text{ (при } nj > 0) \text{ и истинности (8, 9),} \end{cases} \quad (7)$$

$$P_j \in \bigcup_{P_\xi \in G_{nj-1}^1} W(P_\xi) \text{ при } S(P_j) \subseteq G_{nj-1}^1; \quad (8)$$

$$t_\xi^n + t_\xi^0 \leq t(nj) \text{ для всех } P_\xi \in S(P_j). \quad (9)$$

Здесь G_{nj-1}^1 – множество операторов P_j , реали-

зация которых была начата на ярусах $n_j = 1 \dots n_j - 1$.

Процедура 3 (символ 3). Выполняет оценку состава свободного (доступного) в момент времени t аппаратного/программного ресурса, на который могут назначаться для реализации операторы $P_j \in E_{t(n_j)}$. Признаком недоступности ρ -го компонента (функционального модуля) является выполнение условия $pz_\rho = 1$. Признаком недоступности процессора (напомним, что программная реализация последовательной модели поддерживается одним процессором, $NM = 1$) является выполнение условия $pz = 1$.

Процедура 4 (символ 4). Назначением процедуры является:

- при программной ориентации модели – выбор оператора $P_j \in E_{t(n_j)}$ с максимальным приоритетом (минимальным численным значением приоритета b_j), проверка признака $pz = 0$ доступности процессора и “закрепление” оператора P_j для реализации за процессором (таким образом, в момент времени t начинается выполнение одного оператора ($G_t = P_j$), который занимает процессор на время t_j^0 ;
- при аппаратной ориентации модели – фиксация статических параметров (тип оператора, значения разрядности входных и выходных данных) операторов $P_j \in E_{t(n_j)}$, определение состава доступных функциональных модулей соответствующих типов (имеющих $pz_\rho = 0$) и “закрепление” всех или части операторов множества $E_{t(n_j)}$ за функциональными модулями соответствующих типов для реализации. При отсутствии необходимого для выполнения оператора P_j свободного функционального модуля соответствующего типа такой оператор включается в состав кандидатов на начало реализации в очередной момент времени путем его включения в множество R_t .

Таким образом, в рассматриваемом случае формирование множества G_t операторов $P_j \in E_{t(n_j)}$, назначаемых на реализацию в момент времени t , производится в соответствии со следующими соот-

ношениями:

$$G_t = \bigcup_{\xi=1}^{\mu} G_t^\xi, \quad (10)$$

где G_t^ξ – оператор $P_j \in E_{t(n_j)}^\xi = E_{t(n_j)} \setminus G_t(\xi-1)$, имеющий высший приоритет (этому соответствует минимальное численное значение приоритета) среди операторов $P_\delta \in E_{t(n_j)}^\xi$,

$$b_j = \min_{P_\delta \in E_{t(n_j)}^\xi} b_\delta, \quad (11)$$

где $G_t(\xi-1)$ – множество операторов $P_j \in E_{t(n_j)}$, включенных в G_t за $\xi-1$ предыдущих шагов формирования G_t , $G_t(0) = \emptyset$, при этом в составе ресурса проектирования цифрового устройства имеется свободный функциональный модуль FM_ρ типа $\Theta_\rho = \text{typ}[P_j]$ (напомним, что в состав аппаратного ресурса каждый из модулей необходимого типа входит в единственном количестве). Исходные значения признаков занятости, соответствующие моменту начала синтеза последовательной модели, определяются соотношениями $pz_\rho = 0$, $pz = 0$.

Процедура 5 (символ 6). Обеспечивает формирование значения признака занятости конкретного аппаратного/программного ресурса (функционального модуля / процессора), выделенного для выполнения оператора P_j , $pz_\rho = 1$, $pz = 1$, выполняет расчет момента времени $t^k[j]$ завершения реализации оператора P_j :

$$t^k[j] = t + t_j^0, \quad (12)$$

где t_j^0 – максимальное значение времени выполнения модулем FM_ρ операции/функции типа $\Theta_\rho = \text{typ}[P_j]$, и рассчитывает момент времени t_{min} наступления ближайшего события “появление свободного ресурса” следующим образом:

- при программной ориентации модели
- при аппаратной ориентации модели

$$t_{min} = t^k[j];$$

$$t_{\min} = \min_{FM_p \in F_1^t} t^k[\rho], \quad (13)$$

где $F_1^t \subseteq F$ – множество функциональных модулей FM_p , имеющих значение признака занятости $pz_p = 1$ (т.е. занятых в момент t выполнением ранее назначенных на реализацию или начинающих выполняться в момент времени t операторов $P_i \in P$), F – исходный ресурс функциональных модулей необходимых типов.

Процедура 6 (символ 8). Проверяет факт завершения синтеза последовательной модели. Признаком завершения является назначение на выполнение всех операторов P_j формата СВМ Си – программы. Этому событию соответствует выполнение следующих условий: первое условие (14) соответствует проектированию модели для аппаратных приложений, второе условие (15) – проектированию модели для программных приложений:

$$\bigcup_{nj=0}^{kj} (G_t \cap P_j) = P; \quad (14)$$

$$\left(\bigcup_{nj=0}^{kj} G_t \right) \setminus P = \emptyset, \quad (15)$$

где kj – максимальный номер временного яруса модели.

III. Расчет диапазонов изменения эффективности моделей выполнения алгоритмов.

Для оценки эффективности используются следующие показатели:

Среднее время реализации множества P операторов произвольного алгоритма определяется выражением (16) [10]:

$$T(P) = \sum_{\xi=1}^W p_{\xi} T_{\xi}, \quad (16)$$

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 (17):

$$T_{\xi} = \max_{P_j \in P(\xi)} (t_j^H + t_j^0), \quad (17)$$

где $P(\xi)$ – множество операторов ξ -й ветви, t_j^H и t_j^0 – момент начала и относительная временная глубина оператора $P_j \in P(\xi)$.

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

Оценка диапазонов изменения времени реализации сжатой и последовательной моделей обеспечивается показателем DT :

$$DT = \frac{T_{noc}(P)}{T_{сж}(P)}. \quad (18)$$

В соотношении (18) $T_{noc}(P)$ и $T_{сж}(P)$ – среднее время соответственно последовательной и сжатой реализации алгоритма (или максимальные времена реализации алгоритма).

Эквивалентная вентиляльная сложность аппаратных реализаций моделей определяется соотношением (19):

$$Q = \sum_{\eta=1}^v q_{\eta} \cdot n_{\eta}, \quad (19)$$

где v – количество различных типов компонентов (выполняющих каждый конкретную операцию над данными фиксированной разрядности); q_{η} – вентиляльная сложность компонента типа η ; n_{η} – количество компонентов типа η в аппаратной реализации модели.

Рассмотрим пример оценки диапазонов изменения эффективности моделей алгоритма расчета корней системы двух линейных уравнений.

Исходные данные для проектирования.

Список операций и типов данных, времен выполнения операций и список обозначений аппаратных функциональных модулей приведен в табл. 1.

Исходная Си-программа представлена текстом 1 на рис. 2.

На рис. 3 представлен пример проектирования параллельной (сжатой) временной модели алгоритма расчета корней системы двух линейных уравнений с минимальной сложностью аппаратной/ программной реализации.

```

Си-программа задачи
#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;
    y = ( a * f - c * d ) / m;
}
    
```

Рис. 2. Текст 1 Си-программы

Временная параллельная граф – схема (рис. 2) сжатой параллельной модели отображает состав операторов Си-программы алгоритма, их связи по данным и осведомительные связи в статике и динамике процесса максимально параллельного выполнения алгоритма (в реальном масштабе времени) для случая программной ориентации модели.

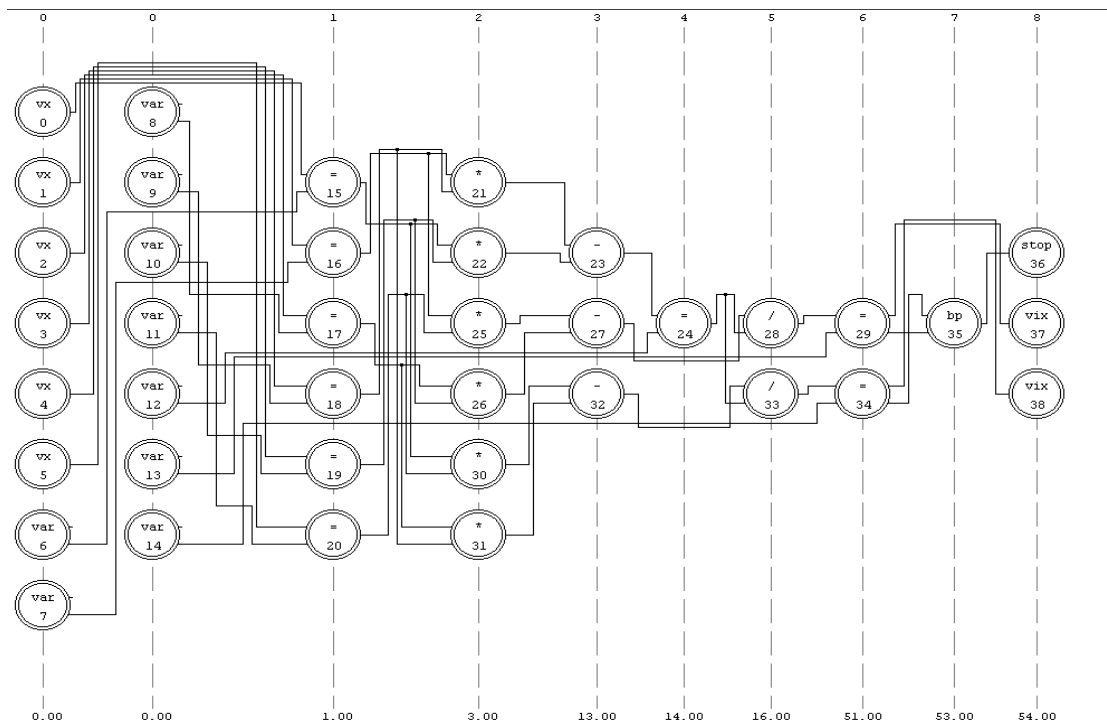


Рис. 3. Максимально параллельная (сжатая) ВПГС

На рис. 4 и 5 представлены результаты проектирования последовательной временной модели алгоритма расчета корней системы двух линейных уравнений с минимальной сложностью аппаратной/ программной реализации.

Таблица 1

Списки операций и типов данных, времен выполнения операций и обозначений аппаратных функциональных модулей

№№	Типы данных и операции	Обозначение аппаратного модуля	Относительное время выполнения операций
1	con	RG con	1.0
2	var	var	1.0
3	+	ADD	1.0
4	*	MUL	10.0
5	/	DIV	35.0
6	-	SUB	1.0
7	=	RG var	2.0
8	==	EQU	1.0
9	if	upl	1.0
10	vh	vx	1.0
11	&	&	1.0
12			1.0
13	&&	&&	1.0
14			1.0
15	goto	bp	1.0
16	vix	vix	1.0
17	stop	stop	1.0

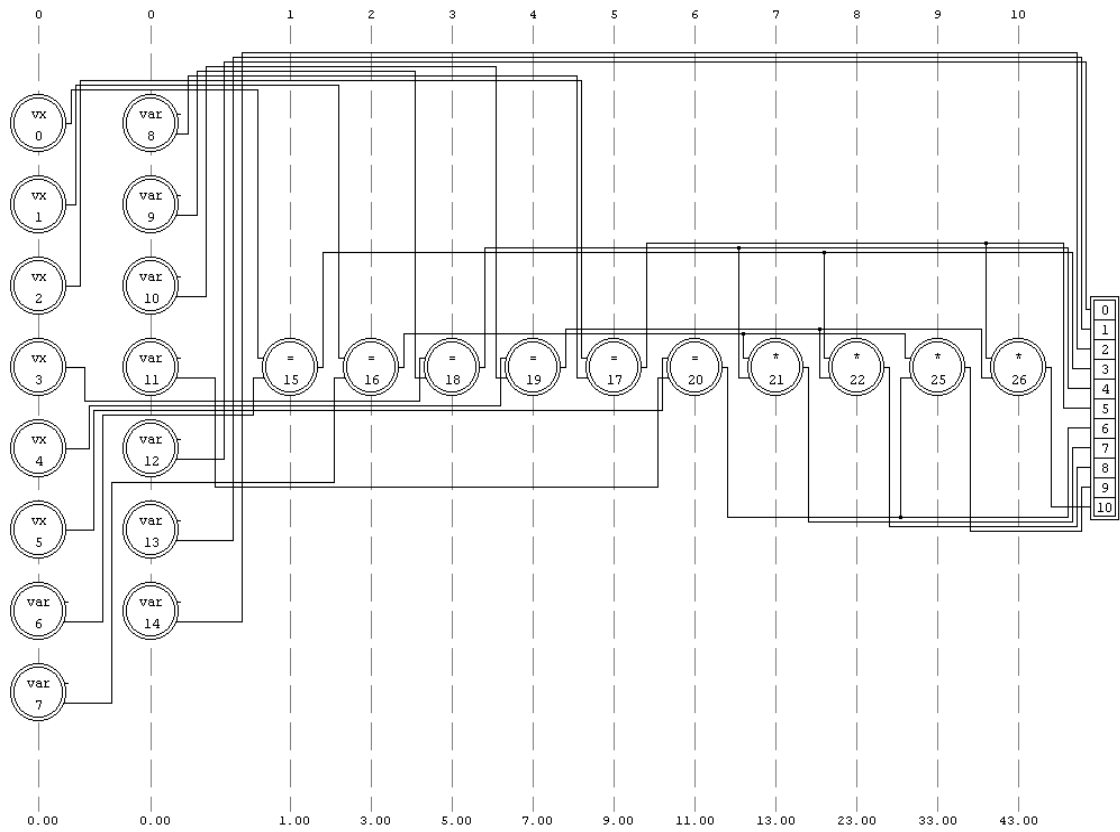


Рис. 4. ВПГС послідовальної моделі алгоритма розрахунку коренів системи двох лінійних рівнянь з двома невідомими (текст 1) для програмних додатків (початок)

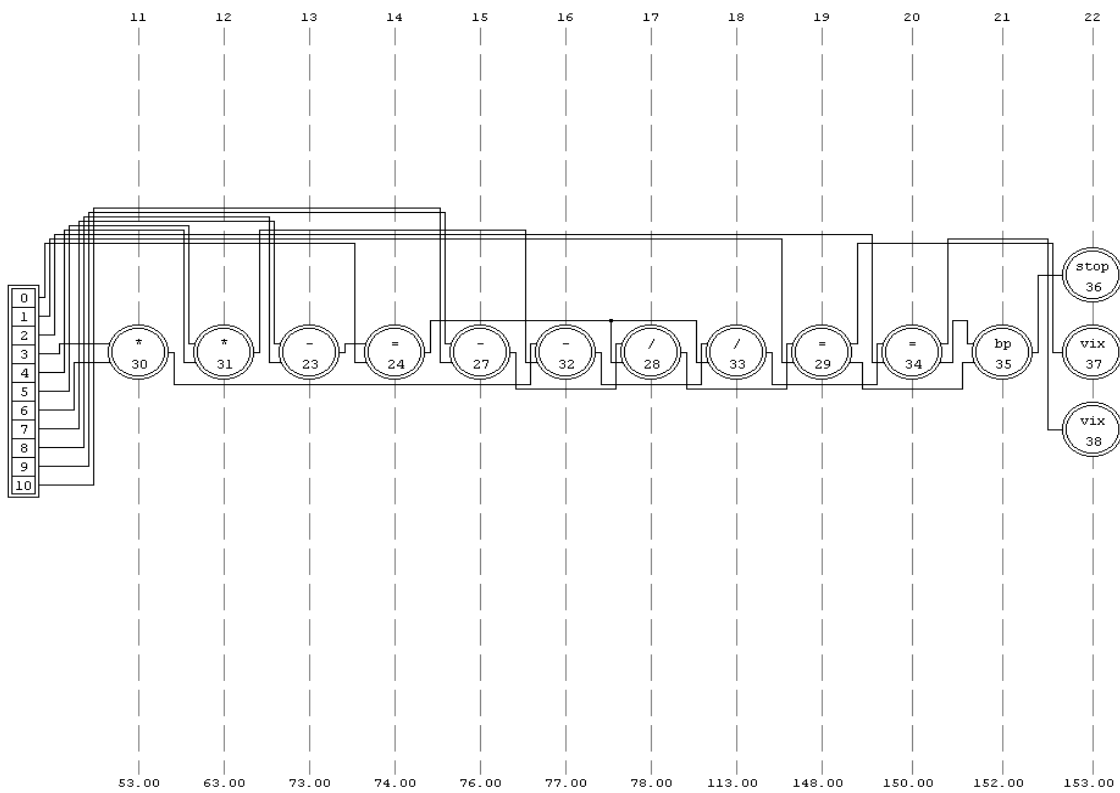


Рис. 5. ВПГС послідовальної моделі алгоритма розрахунку коренів системи двох лінійних рівнянь

с двумя неизвестными (текст 1) для программных приложений (окончание)

Для иллюстрационного примера оценка диапазона изменения времени выполнения включает:

– оценки времени выполнения моделей:

$$T_{сж}(P) = 55,0 \text{ (такт)}, T_{нос}(P) = 153,0 \text{ (такт)};$$

– оценку выигрыша во времени реализации модели при переходе от последовательного выполнения к максимально параллельному $DT = 2,78$ (раз);

– фиксацию диапазона изменения времени

$$TT = \{55,0 \dots 153,0\}.$$

Для оценки диапазона изменения сложности зададимся условной вентиляльной сложностью функциональных модулей $q_* = 150000$, $q_- = 20000$, $q_+ = 30000$, $q_0 = 48$. Тогда:

$$Q_{сж} = 6 q_* + 3 q_- + 2 q_+ + 8 q_0 = 1000584 \text{ экв.вент};$$

$$Q_{нос} = q_* + q_- + q_+ + 8 q_0 = 200384 \text{ экв. вент.}$$

Оценка проигрыша в сложности при переходе от последовательной реализации моделей к максимально параллельной составляет $DQ = Q_{сж} / Q_{нос} \approx 5$ раз, а диапазон изменения сложности

$$QQ = \{1000584 \dots 200384\}.$$

Выводы

1. Изложенные в статье результаты обеспечивают формальный синтез моделей с потенциально минимальным временем выполнения (и максимально необходимой сложностью аппаратного/программного ресурса) и/или с минимальной сложностью (и достижимым при этом временем реализации алгоритмов) и оценку диапазонов изменения эффективности моделей.

2. Результаты синтеза моделей позволяют рассчитывать значения показателей эффективности и производить оценку диапазона изменений эффективности моделей алгоритмов (времени и сложности реализации моделей).

3. Программные средства автоматического решения задачи обеспечивают оперативность оценки диапазонов изменения эффективности при проектировании параллельных цифровых

систем с учетом различных требований и ограничений.

Литература

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ – Петербург, 2002. – 608 с.
2. Трахтенгерц Э.А. Введение в теорию анализа и распараллеливания программ ЭВМ в процессе трансляции. – М.: Наука, 1981. – 256 с.
3. Корнеев В.В. Параллельные вычислительные системы. – М.: Нолидж, 1999. – 320 с.
4. Polyakov G. The hard-and-soft Automatic Design of Self-Organizing Adaptive Systems // Радиоэлектроника и информатика. – 2003. – № 3. – С. 149.
5. Норенков И.П. Основы автоматизированного проектирования: Учебник для вузов. – М.: МГТУ им. Н.Э. Баумана, 2002. – 336 с.
6. Бибило П.Н. Синтез логических схем с использованием языка VHDL. – М.: СОЛОН-Р, 2002. – 384 с.
7. Разевиг В.Д. Система проектирования цифровых устройств OrCAD. – М.: СОЛОН-Р, 2000. – 160 с.
8. Кофман А., Дебазей Г. Сетевые методы планирования / Пер. с французского. – М.: Прогресс, 1968. – 184 с.
9. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World. – СПб.: БХВ-Петербург, 2004. – 368 с.
10. Поляков Г.А., Умрихин Ю.Д. Автоматизация проектирования сложных цифровых систем коммутации и управления. – М.: Радио и связь, 1988. – 304 с.

Поступила в редакцию 4.08.2005

Рецензент: д-р техн. наук, проф. Г.Ф. Кривуля, Харьковский национальный университет радиоэлектроники, Харьков.