

УДК 680.3

Г.А. Поляков¹, Е.Г. Толстолужская², Д.А. Толстолужский³¹Академия наук прикладной радиоэлектроники, Москва²Харьковский университет Воздушных Сил им. И. Кожедуба³Харьковский национальный университет им. В.Н. Каразина

МЕТОД СИНТЕЗА ПАРАЛЛЕЛЬНО-ДЕКОМПОЗИЦИОННЫХ МОДЕЛЕЙ АЛГОРИТМОВ ДЛЯ ВЫСОКОДИНАМИЧНЫХ КРИТИЧЕСКИХ СИСТЕМ

Рассматриваются результаты решения задачи синтеза параллельно-декомпозиционных моделей алгоритмов при задании требований к тактовой частоте или величине такта, элементной базе и отсутствии ограничений на сложность аппаратной или программной реализации модели. Описывается методология, использующая при построении параллельных моделей алгоритмов метод совмещения операций, декомпозиционную обработку и кодово-матричный метод, либо некоторую комбинацию этих методов, адекватную системе требований и ограничений и временным параметрам элементной базы.

параллельные модели, декомпозиционная обработка, сопряженно-внешние связи, декомпозиционные фрагменты

Актуальность исследования

Существует большой класс систем обработки информации и управления, в которых вычислительные средства должны выдавать результаты обработки данных в темпе поступления на вход системы новых наборов исходных данных от внешних источников, то есть в реальном масштабе времени (РМВ). К числу систем РМВ относятся системы обработки радиолокационной информации, системы управления полетами аэрокосмических объектов, управления быстротечными технологическими процессами, системы цифрового телевидения, телекоммуникационные системы с цифровым кодированием и декодированием передаваемых данных и т.п. При проектировании аппаратных и программных вычислительных средств для таких систем одним из основных требований является определяемая внешней средой величина тактового интервала ввода в систему очередных наборов данных (тактовое время, ТТ) или тактовая частота $F = 1/ТТ$ обработки данных. Актуальность этой проблемы обуславливает необходимость разработки методов и средств синтеза параллельных моделей алгоритмов с заданной тактовой частотой F_3 (тактом $ТТ_3$).

Рассматриваются результаты решения задачи синтеза параллельно-декомпозиционных моделей алгоритмов при задании требований к тактовой частоте F_3 (величине такта $ТТ_3$), заданной элементной базе и отсутствии ограничений на сложность аппаратной или программной реализации модели. Описываемая методология использует при построении параллельных моделей алгоритмов метод совмещения операций, декомпозиционную обработку и кодово-матричный метод (либо некоторую комбинацию этих методов, адекватную системе требований и ограничений и временным параметрам используемой элементной базы).

Постановка задачи исследования

Исходными данными для синтеза являются:

- Си-программа, представляющая формальное описание рассматриваемого алгоритма;
- спецификация Си-программы в числовом формате (представленная структурами данных состава и связей операторов);
- библиотеки функциональных модулей или процессоров и их технические характеристики: значения t_j^0 времени реализации операций или функций алгоритма (например, с, мс, мкс или нс при синтезе парал-

лельных моделей аппаратных средств) или в тактах – при синтезе параллельных моделей алгоритмов для программных средств, а также вентиляжная сложность функциональных модулей и (или) процессоров;

- требуемое значение тактовой частоты F_3 (тактового интервала TT_3);
- характер прикладной ориентации модели (для аппаратных или для программных средств).

Необходимо синтезировать в числовом формате параллельно-декомпозиционную модель алгоритма, удовлетворяющую заданным требованиям к тактовой частоте F_3 , (величине такта TT_3). При синтезе моделей алгоритма использовать декомпозиционный метод, метод совмещения независимых операций, кодово-матричный метод или их оптимальную комбинацию в зависимости от значений F_3 (TT_3).

Основные этапы синтеза параллельно-декомпозиционных моделей алгоритмов представлены на рис. 1.

На первом этапе (блок 2, рис. 1) для каждого оператора P_j Си-программы алгоритма рассчитывается соответствующее значение коэффициента декомпозиции $K_j^d = t_j^0 / TT_3$, где t_j^0 – время выполнения оператора P_j , TT_3 – требуемое значение тактового интервала (определяемое значением тактовой частоты F_3), $j = 0, n-1$, где n – количество операций/функций в алгоритме.

На втором этапе (блок 3, рис. 1) для рассматриваемого алгоритма определяется максимальное значение коэффициента декомпозиции

$$K_{\max}^d = \max(K_j^d). \quad (1)$$

На третьем этапе (блок 4, рис.2) для каждого оператора P_j формируется “декомпозиционное расширение” $\Delta P_{\mu_j}^d = \{P_{\mu_j}^d\}$ – подмножество ΔP_j^d операторов $P_{\mu_j}^d$ в соответствии со следующими соотношениями:

- тип оператора $P_{\mu_j}^d$ определяется типом оператора P_j $\text{typ}[P_{\mu_j}^d] = \text{typ}[P_j]$;

- время выполнения $t_{\mu_j}^0$ оператором $P_{\mu_j}^d$ соответствующей операции типа $\text{typ}[P_j]$ совпадает со временем t_j^0 оператора P_j $t_{\mu_j}^0 = t_j^0$;

- момент $t_{\mu_j}^H$ начала выполнения оператора $P_{\mu_j}^d$ связан следующим образом с моментом начала t_j^H оператора P_j :

$$t_{\mu_j}^H = t_{\mu_j-1}^H + DT, \quad (2)$$

где $\mu_j = n_j^d + \rho$, $\rho = 1, 2, \dots, K_{\max}^d - 1$; n_j^d –

количество операторов-вершин в расширении формата СВМ к моменту начала формирования множества ΔP_j^d для оператора P_j ;

- структура сопряженных и внешних связей декомпозиционного оператора $P_{\mu_j}^d \in \Delta P_j^d$ идентична структуре сопряженных и внешних связей базового оператора P_j с операторами базового множества P .

На четвертом этапе (блок 5, рис. 1) формируется множество $P^{(D)}$ операторов $P_j^{(D)}$ коммутации временной последовательности входных наборов данных алгоритма на входные операторы $P_{\mu_j}^d \in \Delta P_j^d$, соответствующих (синтезированных в количестве $K_{\max}^d - 1$) копий базисного множества $P = \{P_j\}$.

Количество операторов коммутации (демультиплекторов) определяется мощностью набора входных данных алгоритма, каждый демультиплектор обеспечивает управляемую номером входного набора данных коммутацию конкретного значения данного на соответствующий входной оператор памяти каждой из $(K_{\max}^d - 1)$ копий базисного множества P операторов P_j . Число управляющих входов и выходов данных каждого оператора – демультиплектора P_j^K соответственно равно $\lceil \log_2 K_j^d \rceil + 1$ и K_j^d .

На данном этапе обеспечивается также формирование множества P^M операторов P_j^M мультиплексирования временной последовательности выходных наборов результатов выполнения различных копий базисного множества P операторов P_j . Количество операторов – мультиплекторов определяется мощностью набора выходных данных алгоритма, каждый мультиплектор обеспечивает управляемую номером входного набора данных передачу значения конкретного результата из соответствующего выходного оператора памяти каждой из $K_{\max}^d - 1$ копий базисного множества P операторов P_j на соответствующий выход параллельно-декомпозиционной модели. Число управляющих входов и выходов данных каждого оператора-мультиплектора $P_j^{(M)}$ равно соответственно

$$\lceil \log_2(K_j^d) \rceil + 1 \text{ и } K_j^d.$$

На пятом этапе (блок 6, рис. 1) формируется множество P^c операторов управления P_j^c , обеспечивающих ввод в модель сигналов временной синхронизации clk (оператор P_j^{cl}) и операторов подсчета суммы S_{clk} тактовых сигналов clk ((операторы P_j^{ct} типа «ct»), определения, тем самым, номеров наборов входных данных), а также формирования сигналов «ct» управления операторами-демультиплекторами P_j^D в соответствии с соотношением

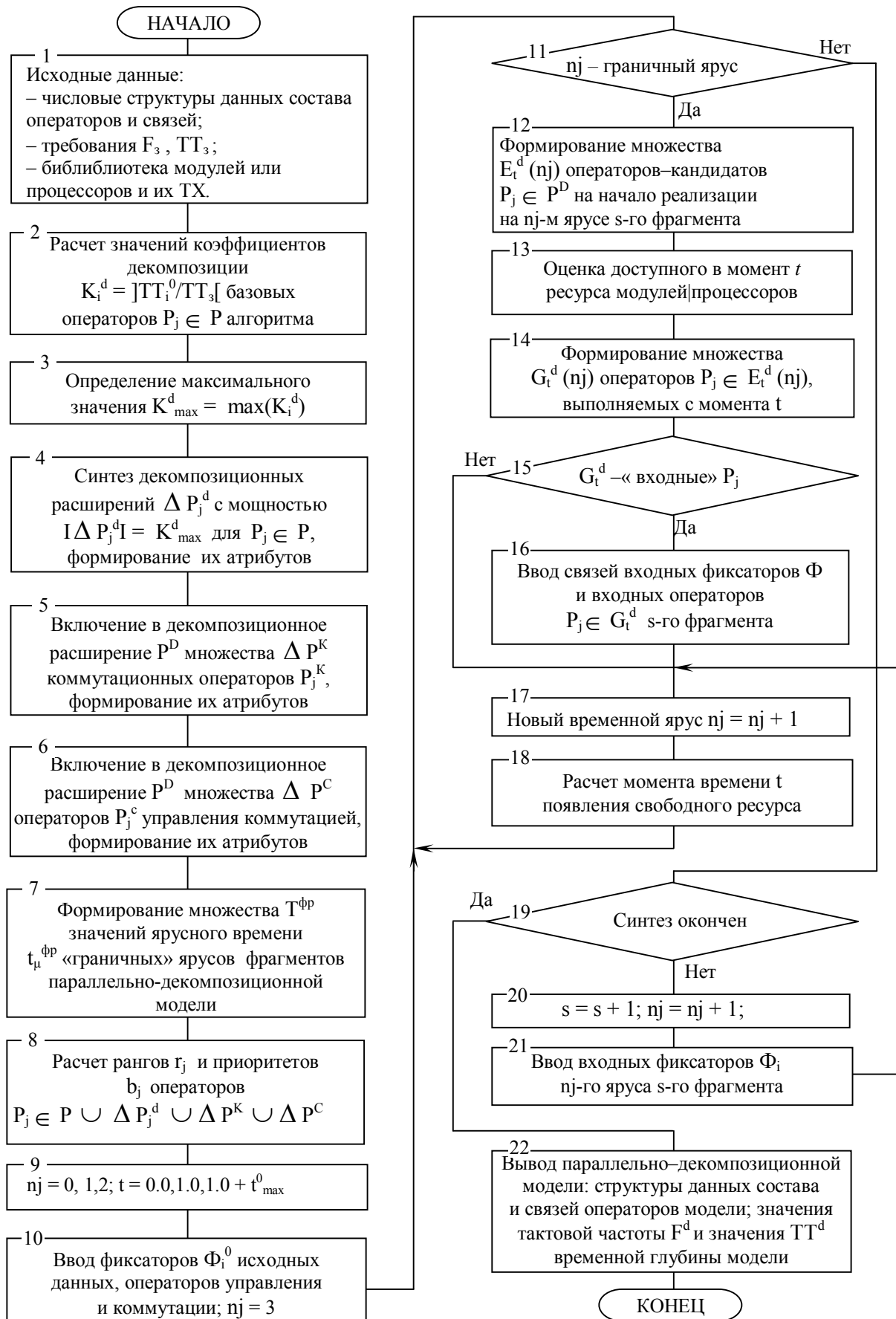


Рис. 1. Этапы синтеза параллельно-декомпозиционных моделей алгоритмов

$$ct_j = (S_{clk}) \bmod K_j^d. \quad (3)$$

Рассмотренные этапы обеспечивают решение первой задачи синтеза параллельно-декомпозиционных временных моделей алгоритмов – расширения базового множества P операторов путем формирования множества базисных копий P^1, P^2, \dots, P^q ($q = K_{max}^d - 1$) с получением расширенного базового множества $P^B = P^1 \cup P^2 \cup \dots \cup P^q$. Си-программы алгоритма, а также путем введения множеств P^D, P^M, P^C операторов демультиплексирования P_j^D , мультиплексирования P_j^M и управления P_j^C с соответствующими сопряженными и внешними связями между ними.

В результате оказывается сформированным расширенное множество $R = P \cup P^D \cup P^M \cup P^C$ параллельно-декомпозиционной модели алгоритма.

Содержанием второй задачи синтеза параллельно-декомпозиционных временных моделей алгоритмов является:

- введение в состав расширенного множества R множества P^Φ операторов-фиксаторов, разделяющих параллельно-декомпозиционные фрагменты синтезируемой временной модели;
- формирование сопряжено-внешних связей между операторами различных групп операторов P_j расширенного множества R ;
- решение задачи временной параметризации операторов $P_j \in P \cup P^D \cup P^M \cup P^C \cup P^\Phi$ (то есть, определение для каждого оператора P_j момента времени t_j^H начала его выполнения).

Решение второй задачи синтеза параллельно-декомпозиционных временных моделей алгоритмов охватывает блоки 7...25 (рис. 1). На шестом этапе (блок 7, рис. 1) формируется множество $T^{FP} = \{t_s^{FP}\}$ значений ярусного времени t_s^{FP} «граничных» временных ярусов (разделяющих параллельно-декомпозиционные фрагменты синтезируемой временной модели) в соответствии со следующими соотношениями: $t_1^{FP} = 1,00$ (для первого, $s = 1$, фрагмента), $t_s^{FP} = t_1^{FP} + (s-1)TT_3$ (для $s = 2, 3, \dots, K_{max}^d$).

На седьмом этапе (блок 8, рис. 1) для операторов $P_j \in R$ рассчитываются значения ранга r_j и приоритета b_j , определяющие очередность рассмотрения операторов при решении задачи назначения на выполнение.

На восьмом этапе (блоки 9, 10, рис. 1) выполняется ввод (при $n_j = 0$) оператора типа «clk» (тактовый сигнал), фиксаторов Φ_i^0 исходных данных синтезируемой модели и операторов управления (при $n_j = 1$), а также операторов коммутации (при $n_j = 2$).

На девятом этапе (блоки 12, 13, 14, рис. 1) осуществляется решение следующих задач: выделение из расширенного множества R подмножества $E_t^d(n_j)$ операторов P_j , для начала выполнения которых на n_j -м ярусе s -го фрагмента готовы все необходимые операнды;

- оценка доступного в текущий момент времени t ресурса функциональных модулей (при аппаратной ориентации приложений) или процессоров (при программной ориентации приложения);
- формирование множества $G_t^d(n_j) \subseteq E_t^d(n_j)$ операторов $P_j \in E_t^d(n_j)$, начинающих выполняться на n_j -м ярусе в момент времени t при наличии необходимого для их выполнения свободного в момент t ресурса.

На десятом этапе (блоки 15, 16, рис. 1) формируются сопряжено-внешние связи входных операторов – фиксаторов Φ_i n_j -го временного яруса s -го декомпозиционного фрагмента и входных базисных операторов $P_j \in G_t^d(n_j)$ этого фрагмента (при выполнении условия, проверяемого символом 15, что, все операторы множества $G_t^d(n_j)$ являются входными для s -го фрагмента). Формирование сопряжено-внешних связей осуществляется на основе операций теории графов.

На одиннадцатом этапе (блоки 17, 18, рис. 1) рассчитывается очередной номер $n_j = n_j + 1$ временного яруса параллельно-декомпозиционной модели и определяется ближайший момент времени t появления свободного ресурса, определяющий значение очередного ярусного времени $t(n_j)$.

На двенадцатом этапе (блоки 19 – 21, рис. 1) проверяется факт окончания синтеза параллельно-декомпозиционной модели (при $s > K_{max}^d$), в случае невыполнения условия формируется номер $s = s + 1$ очередного декомпозиционного фрагмента и очередного номера $n_j = n_j + 1$ временного яруса синтезируемой модели.

Выполнение рассмотренных выше этапов обеспечивает синтез параллельно-декомпозиционных временных моделей алгоритмов, удовлетворяющих заданным требованиям к тактовой частоте F_3 (или к величине такта TT_3).

Поясним на простом примере содержание и особенности основных этапов. Исходные данные: Си-программа алгоритма (рис. 2);

- величина такта TT модели не должна превышать значения $TT_3 \leq 24,00$ нс (что соответствует тактовой частоте $F_3 = 1/TT_3 \geq 41,6$ МГц);
- длительности (в нс) выполнения функциональными модулями различных типов операторов Си- программы (табл. 1).

Требуется: синтезировать параллельно-декомпозиционную модель с использованием метода совмещения независимых операций и декомпозиционной обработки (для аппаратных приложений).

Таблица 1

Значения времени $t^0(\text{typ})$ выполнения функциональными модулями операторов P_j различных типов (typ) параллельно-декомпозиционной модели (нс)

Тип	vx, vix	/	var	=	clk	upl	*	dmx	/o	ctr	mux	*,=	stop
t_j^0	1,0	11,0	1,0	1,50	1,0	0,88	41,29	0,68	1,0	1,0	1,00	1,05	1,0

```
#include <stdio.h>
void main(void)
{
  int a,b;
  int z,s;
  scanf("%d %d %d %d",&a);
  scanf("%d %d %d %d",&b);
  if(a == b)
  { z = a * b ;
    printf("%4d\n",z);
  }
  else
  {
    s = b / a ;
    printf("%4d\n",s);
  }
}
```

Рис. 2. Си-программа разветвляющегося алгоритма

Результаты проектирования параллельно-декомпозиционной модели разветвляющегося алгоритма представляет временная параллельная модель (рис. 3). Модель включает $df = 5$ декомпозиционных фрагментов FR_s с номерами $s = 0, 1, 2, 3, 4$.

В состав различных фрагментов параллельно-декомпозиционной модели входят следующие операторы: в нулевой фрагмент ($s = 0, p_j = 0, 1, 2$) – операторы P_j с номерами: $j = 0, 1, 2, 3, 22, 23, 37, 6, 7, 38, 39, 40$; в первый фрагмент ($s = 1, p_j = 3 \dots 7$) – операторы P_j с номерами $j = 5, 20, 21, 8, 9, 18, 19, 10, 12$; во второй фрагмент ($s = 2, p_j = 8 \dots 12$) – с номерами $j = 4, 13, 25, 26, 27, 44, 28, 29, 35, 36, 30, 32$; в третий фрагмент ($s = 3, p_j = 13, 14$) – с номерами $j = 11, 24, 33, 45, 46, 14$; в четвертый фрагмент ($s = 4, p_j = 15 \dots 18$) – операторы с номерами 31, 47, 34, 41, 42, 43, 15, 16, 17. Это обеспечивает в каждый момент времени реализацию заданного алгоритма по пяти смежным (по времени ввода в модель) наборам исходных данных. Величина такта TT модели удовлетворяет заданному требованию: значения $TT = TT_3 = 24,00$ нс (что соответствует тактовой частоте $F_3 = 1/TT_3 \cong 41,6$ МГц).

Обобщенная структура параллельно-декомпозиционных моделей представлена на рис. 4. Различные элементы этой структуры могут быть конкретизированы с помощью декомпозиционной параллельной временной модели (рис. 3) следующим образом:

– входной ($p_j = 0$) ярус – содержит операторы – входы данных $P_0, P_1(a_{in}, b_{in})$, операторы – имена

исходных данных $P_2, P_3(a, b)$, операторы – имена переменных $P_{22}, P_{23}(z, s)$, а также вводимые в процессе синтеза параллельно-декомпозиционной модели операторы – имена данных P_{18}, P_{19} и оператор – вход P_{29} сигнала «clk» временной синхронизации;

– временные ярусы управления декомпозиционной моделью (ярусы с номерами $p_j = 1, 2$), на которых размещаются операторы приема очередных значений входных наборов данных «a,b» (операторы типа «=» P_6, P_7), оператор подсчета количества введенных в декомпозиционную модель наборов данных (оператор типа «ctr» P_{38}) и операторы коммутации (типа «dmx» P_{39}, P_{40}), обеспечивающие декомпозицию временной последовательности наборов значений входных данных «a,b» на подпоследовательности (с номерами $1, 2, \dots, k_{max}^d$);

– множество L линий L_p (с номерами $p = 1, 2, \dots, 1/2 \cdot k_{max}^d$) «вычислительных» фрагментов FR_s , каждая из которых представляет собой, в общем случае, цепочку фрагментов модели, обеспечивающих решение задачи для конкретного набора входных данных; рассматриваемая модель (рис.3) включает две линии L_1 и L_2 (так как $k_{max}^d = 2$), в состав первой линии входят фрагменты FR_0, FR_1, FR_3, FR_4 , в состав второй линии – фрагмент FR_0, FR_2, FR_3, FR_4 ;

– каждый вычислительный фрагмент FR_s содержит входной временной ярус, на котором размещены входные операторы-фиксаторы (типа «=»), обеспечивающие прием результатов промежуточных вычислений, выполненных предшествующим фрагментом (или несколькими предшествующими фрагментами – в случае наличия в их составе операторов с временем выполнения $t_j^0 > TT$) рассматриваемой линии фрагментов, а также множество временных ярусов (заканчивающихся «выходным ярусом фрагмента»), на которых расположены операторы, выполняющие комбинационные операции (арифметические, логические и т.д.) над данными;

– линии $L_p \in L$ вычислительных фрагментов заканчиваются «выходными временными ярусами» с номерами $p_j = 16, 17$, содержащими операторы P_{41}, P_{42} объединения типа «mux» (арифметического объединения значений одноименных данных, вычисляемых при выполнении различных линий) и оператор P_{43} типа «/o» (логического объединения передач управления от выходных операторов различных линий);

– выходной ярус (с номером $p_j = k_j = 18$) модели содержит операторы P_{16}, P_{17} типа «out» вывода результатов декомпозиционной обработки данных

(z_out, s_out) и оператор P₁₅ «stop» завершения выполнения декомпозиционной модели;

– операторы типа «фиксатор значения» (реализующие операции записи/чтения данных, передаваемых между фрагментами): входные фиксаторы P₆, P₇ нулевого фрагмента, фиксаторы P₂₀, P₂₁ первого фрагмента, фиксаторы P₁₃, P₂₆, P₂₇ второго фрагмента, фиксаторы P₁₁, P₃₃, P₄₅ третьего фрагмента и фиксаторы P₃₁, P₃₇ четвертого фрагмента.

Таким образом, параллельно-декомпозиционная модель, обобщенная структура которой пред-

ставлена на рис. 4, включает следующие группы операторов параллельно-декомпозиционной временной модели:

- группы «базовых» операторов (операторы умножения P₁₀, P₃₀) и операторы деления (P₁₂, P₃₂), получаемой «размножением» операторов Си-программы соответствующих типов с коэффициентом размножения, равным k_{max}^d (в примере k_{max}^d = 2);
- группы операторов – фиксаторов (P₆, P₇, P₂₀, P₂₁, P₁₃, P₂₆, P₂₇, P₁₁, P₃₃, P₄₅, P₃₁, P₄₇);

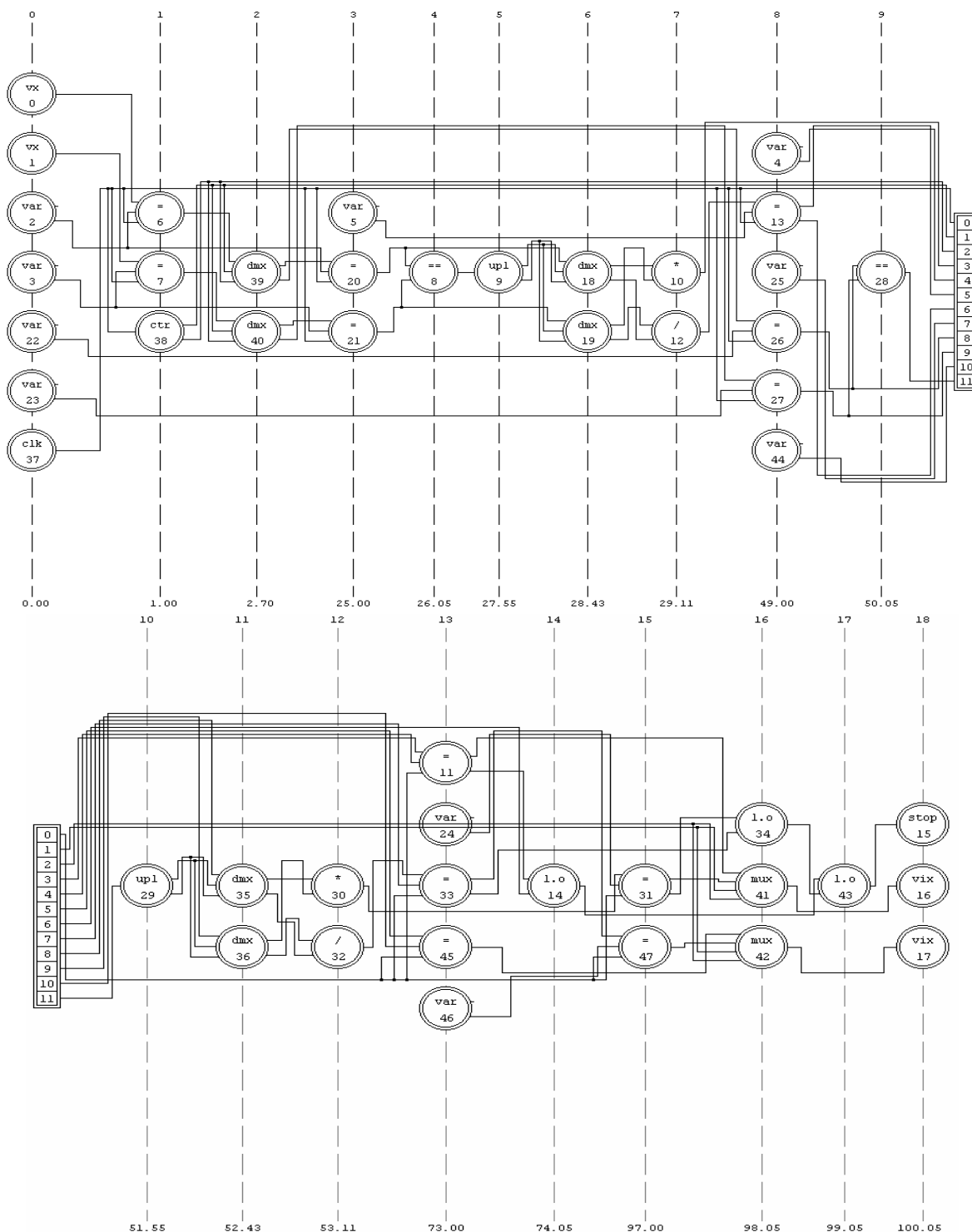


Рис. 3. Параллельно-декомпозиционная модель разветвляющегося алгоритма для аппаратных приложений

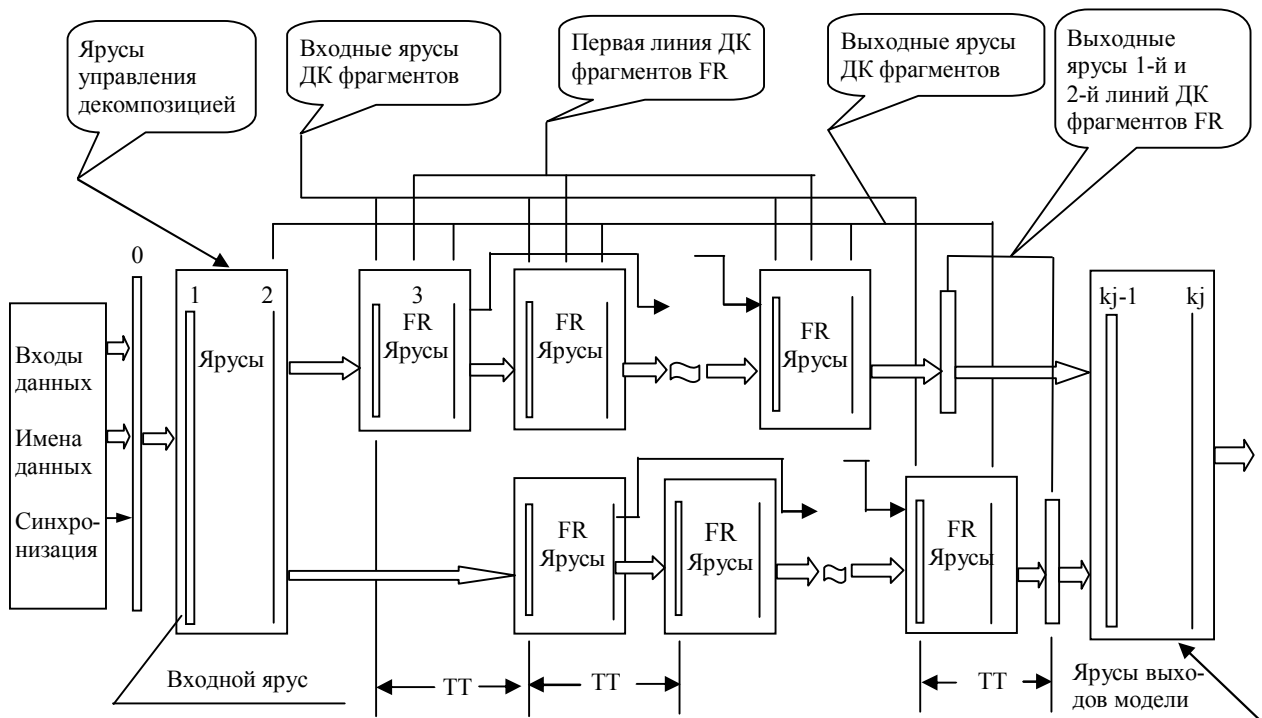


Рис. 4. Обобщенная структура параллельно-декомпозиционных временных моделей алгоритмов

- группы операторов – переименованных переменных ($P_4, P_5, P_{22}, P_{23}, P_4, P_{25}, P_{44}, P_{46}$);
- группы операторов управления синхронизацией (операторы P_{37}, P_{38} типов «clk» и «ctr»);
- группы операторов коммутации данных различных входных наборов между линиями декомпозиционных фрагментов модели (операторы P_{39}, P_{40} типа «dmx»);
- группы операторов коммутации данных ($P_{18}, P_{19}, P_{35}, P_{36}$ типа «dmx») на различные естественные части алгоритма по результатам проверки логических условий операторами типа «upl» (P_9, P_{29});
- группы операторов (P_{41}, P_{42} типа «mix») арифметического объединения значений выходных переменных «z, s», получаемых для различных наборов значений входных данных Си-программы;
- группы операторов (операторы P_{34}, P_{43}) объединения связей по управлению от выходных операторов различных линий фрагментов, обрабатывающих различные входные наборы данных.

Выводы

1. Одним из требований, предъявляемым к высокочастотным критическим системам, является обработка данных в реальном времени с высокой тактовой частотой. Основными путями решения этой проблемы в настоящее время являются разработка и использование новой быстродействующей элементной базы и применение конвейерной обработки данных.

2. В статье изложен метод формального синтеза параллельно-декомпозиционных моделей временных алгоритмов. Метод обеспечивает возможности дальнейшего, по сравнению с конвейерной обработкой (при использовании одной и той же элементной базы), увеличения тактовой частоты и соответствующего увеличения производительности цифровых устройств.

3. Разработанный метод может рассматриваться как основа для разработки инструментальных программных средств автоматического проектирования параллельных цифровых устройств для высокочастотных критических приложений.

Список литературы

1. Поляков Г.А. Адаптивные самоорганизующиеся системы с мультипараллельной обработкой данных - стратегия развития цифровой вычислительной техники в XXI веке? // Сб. науч. тр. 1-й Международный радиоэлектронный Форум «Прикладная радиоэлектроника. Состояние и перспективы развития». – Х.: АНПРЭ, ХНУРЭ. – 2002. – Ч. 2. – С. 237-240.
2. Состояние и основные направления развития высокопроизводительных вычислительных средств / ХВУ; Е.Г. Волокитина, Н.В. Матчина, В.В. Онищенко; Под ред. Г.А.Полякова. – Х.: НЦ РКИ, НТЦ НККУ, 1994. – 306 с.
3. Кривуля Г.Ф., Хаханов В.И. Новые информационные технологии проектирования цифровых систем // Сб. науч. тр. 1-й Международный радиоэлектронный Форум «Прикладная радиоэлектроника. Состояние и перспективы развития» МФ-2002. – Х.: АНПРЭ, ХНУРЭ. – 2002. – Ч. 2. – С. 233-236.

Поступила в редколлегию 12.03.2007

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет "ХАИ", Харьков.