

Статья. Библиографические данные:

Житников А.П.

Параллельные алгоритмы технологических мехатронных систем.

/ 2-я Всероссийская науч.-тех. конф. "Мехатроника, автоматизация, управление:

Сб. трудов. Том 2.

– Уфа: УГАТУ, 2005. С. 155 – 160.

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ТЕХНОЛОГИЧЕСКИХ МЕХАТРОННЫХ СИСТЕМ

А.П. Житников

Россия, г. Уфа, УГАТУ

E-mail: paralg@mail.rb.ru

(новый адрес e-mail: paralg@mail.ru)

Излагаются исходные аспекты техники структурных построений параллельных алгоритмов технологических мехатронных систем.

параллельные алгоритмы, структурные формулы, структурные схемы, временные диаграммы, технологические мнемосхемы

СОДЕРЖАНИЕ

1	ВВЕДЕНИЕ.....	2
	Параллельные процессы и алгоритмы технологических систем.....	2
	Общие концептуальные истоки.....	2
	Специфика автоматизированных технологических систем.....	2
	Средства структурной теории параллельных алгоритмов.....	2
2	АНАЛИЗ ПРИМЕРА АЛГОРИТМИЧЕСКОЙ ЗАДАЧИ.....	4
	Многосторонняя обработка корпусной детали.....	4
	Параллельная конъюнкция алгоритмов (#&).....	4
	Параллельная дизъюнкция алгоритмов (#V).....	5
	Многополюсная блокировка.....	6
	Расширения исходных определений.....	7
3	ЗАКЛЮЧЕНИЕ.....	8
	ЛИТЕРАТУРА.....	9

1 ВВЕДЕНИЕ

11

Параллельные процессы и алгоритмы технологических систем

В организации работы технологических мехатронных систем все более существенную роль играют параллельные дискретные процессы [1] (с совмещением выполняемых действий во времени).

Для решения теоретических и прикладных задач управления параллельными процессами привлекаются различные методы и средства *структурной теории параллельных алгоритмов*. Однако существуют проблемы аналитического описания параллельных структур.

Общие концептуальные истоки

Концептуальным источником параллельной алгоритмизации является исторически первичная область приложения параллельных алгоритмов: *параллельные вычисления* и их *параллельное программирование*.

На этой исходной основе формируется во многом сходное, но относительно самостоятельное направление [2]:

параллельные алгоритмы управления техническими системами – приложения алгоритмов для задач аппаратного и программного управления механическими и технологическими объектами, интегрированных систем мехатроники.

В их составе особую группу составляют так называемые *параллельные алгоритмы логического управления* – используются преимущественно логические методы, вычисления возможны, но они не являются определяющими.

Оба направления формируются на общей концептуальной базе. Однако имеют место определенные различия этих двух направлений [2].

Специфика автоматизированных технологических систем

В автоматизации техпроцессов существуют дополнительные различия:

1) Порядок развития алгоритмизации:

- развитие производственной автоматизации в целом идет от жестких последовательных структур алгоритмов к жестким параллельным структурам и далее – к гибким (переключаемым) параллельным структурам (в вычислениях сложилась другая траектория развития);
- объективно проявляется более простая исходная основа для отражения первичной сущности параллелизма (без осложнения переключаемыми аспектами), что актуально в концептуальном и теоретическом отношении.

2) Сложность процессов как объектов алгоритмизации:

- массовый практический эффект дают даже простейшие параллельные алгоритмы (в отличие от задач параллельных вычислений, где распараллеливание простых задач на ЭВМ не имеет практического смысла);
- имеет место простая база первичных приложений параллельных алгоритмов, что существенно в практическом и теоретическом отношении.

Средства структурной теории параллельных алгоритмов

В статье [3] представлена концепция языка полиморфного описания параллельных алгоритмов, включая строго согласованные и взаимно обратимые формы представления алгоритмов:

- *структурные формулы* (основная исходная форма) – *инфиксные, префиксные, постфиксные и комбинированные* синтаксические виды;
- *вербальные тексты* алгоритмов разных синтаксических видов и лексических групп – алгол-подобные, оккам-подобные тексты и т.п.;
- *структурные схемы* – блок-схемы, штрих-схемы, граф-схемы;

- временные *диаграммы* исполнения алгоритмов – *линейные* и *сетевые*, статические и динамические диаграммы (и их расчетная поддержка [3]).

Для технологических приложений параллельных алгоритмов актуальны различные статические и динамические *мнемосхемы*.

Далее представлены простые тестовые примеры полиморфного структурного описания параллельных алгоритмов:

базисные структуры ключевого концептуального значения с прикладной интерпретацией (из комплекта контрольных примеров разной сложности).

Вербальные тексты алгоритмов не приводятся (вопрос требует отдельного изложения, абстрактные тексты примеров приведены в [3]).

2 АНАЛИЗ ПРИМЕРА АЛГОРИТМИЧЕСКОЙ ЗАДАЧИ

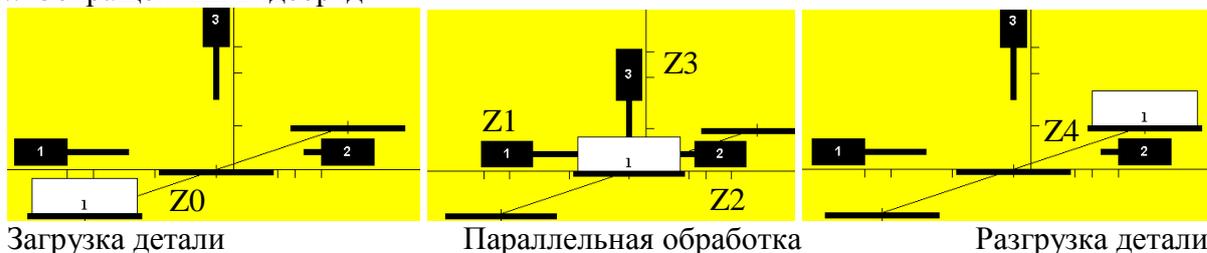
Многосторонняя обработка корпусной детали

Рассматривается обработка корпусных деталей на агрегатном станке. В конфигурацию технологической системы включены три силовые головки для сверления отверстий. Задается операционный цикл (МТС-1):

загрузка детали, параллельная обработка трех отверстий, разгрузка детали.

МТС-1: Мнемосхема технологической системы

// Сокращенный видеоряд



Параллельная конъюнкция алгоритмов (#&)

СКА-1: Система команд алгоритма: // для примера МТС-1

Z0, Z4: Команды (вызова подпрограмм) загрузки и разгрузки детали.

Z1, Z2, Z3: Команды обработки – вызывают подпрограмму типового цикла сверления отверстий: быстрый подвод, рабочая подача, быстрый отвод.

СФА-1: Структурная формула алгоритма / ИнФ: Инфиксная форма

$$A523 = (Z0 - (Z1 \& Z2 \& Z3) - Z4) = Z0 - (Z1 \#\& Z2 \#\& Z3) - Z4$$

$$// \#\& = \&\# \quad Z0 - (Z1 \#\& Z2 \#\& Z3) - Z4 = Z0 - (Z1 \&\# Z2 \&\# Z3) - Z4$$

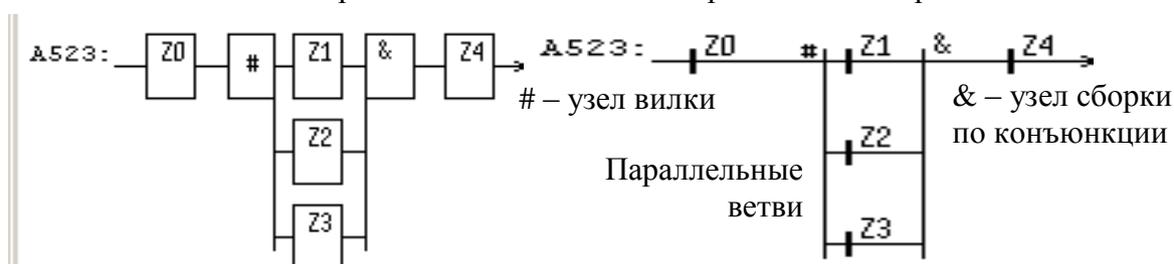
ВИА-1: Вербальная (словесная) интерпретация алгоритма: // по СФА-1

выполнить команду Z0 (загрузка) **затем** (–) выполнить ком. Z1 **и параллельно** (&#) выполнить ком. Z2 **и параллельно** выполнить ком. Z3 (параллельная обработка) **затем** выполнить ком. Z4 (разгрузка детали).

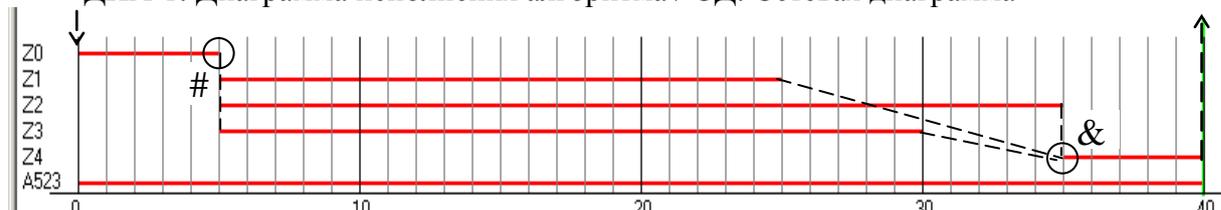
ССА-1: Структурная схема алгоритма. Варианты:

БСА-1: Блок-схема алгоритма

ШСА-1: Штрих-схема алгоритма



ДИА-1: Диаграмма исполнения алгоритма / СД: Сетевая диаграмма



Обеспечивается структурный изоморфизм и взаимная обратимость всех форм представления алгоритмов. Рассматриваются дополнительные аспекты:

СФА-2 / ИнПрПоФ: Инфиксно-префиксно-постфиксная форма
 $A523 = Z0 - (Z1 \# \& Z2 \# \& Z3) - Z4 = Z0 - \#(Z1, Z2, Z3) \& - Z4 =$

СФА-3 / ИнПрПоФ: Инфиксно-префиксная форма // полная форма
 $= Z0 - \# \& (Z1, Z2, Z3) - Z4$

Уточняется структурная семантика алгоритма, определяются разные шаблоны для записи разных видов вербальных текстов алгоритмов [3].

Параллельная дизъюнкция алгоритмов (#V)

Допустим, существует ограничение по степени параллелизма процесса (например, ограничение мощности гидростанции привода силовых головок в системе с мощными режимами резания): возможна работа не более двух силовых головок одновременно – реализуется частичный параллелизм.

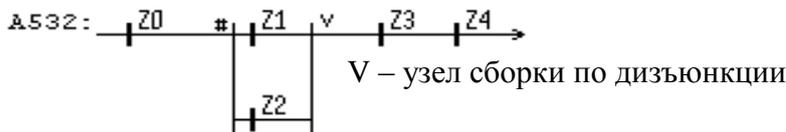
Вариант частичного параллелизма с завершением по конъюнкции (&):

$A522 = (Z0 - (Z1 \& Z2) - Z3 - Z4) = (Z0 - (Z1 \# \& Z2) - Z3 - Z4)$

Более гибкий вариант параллелизма с завершением по дизъюнкции (V):

$A532 = (Z0 - (Z1 V Z2) - Z3 - Z4) = (Z0 - (Z1 \# V Z2) - Z3 - Z4)$

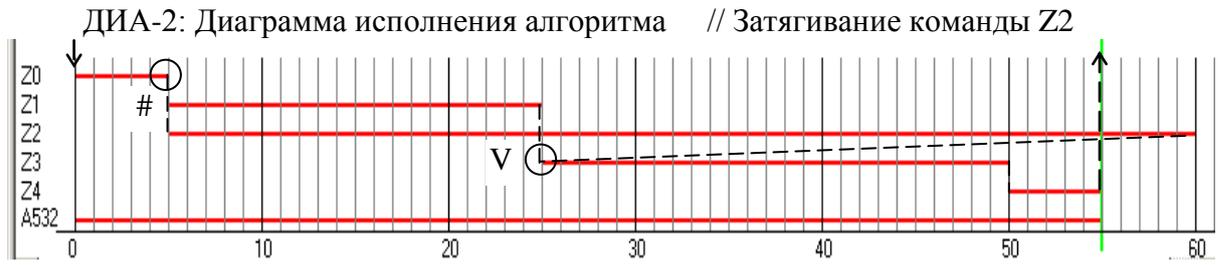
ШСА-2: Штрих-схема алгоритма // Параллельная дизъюнкция



Выполнение команды Z3 (ДИА-2) начинается после завершения более короткой команды Z1, но до окончания параллельной ей команды Z2: простой заменой операций (& на V) достигается экономия времени.

Многополюсная блокировка

Вариант А532 алгоритма повышает эффективность цикла, но возможна неправильная работа системы при затягивании во времени команды Z2:



Разгрузка детали (отработка команды Z4) начинается до окончания обработки детали (команда Z2) – недопустимая аварийная ситуация.

Вводится блокировка для контроля завершения отработки циклов Z1, Z2, Z3 всех силовых головок (СФА-5, ШСА-3):

задаются дополнительные стрелочные связи блокировки – запуск команды Z4 возможен только после отработки всех трех команд Z1, Z2, Z3 управления циклами силовых головок.

Возникает многополюсная структура (с особыми связями), которая не может быть отображена на обычные структурные формулы булевого типа. Используются дополнительные стрелочные обозначения.

СФА-5: Структурная формула алгоритма – многополюсная структура:

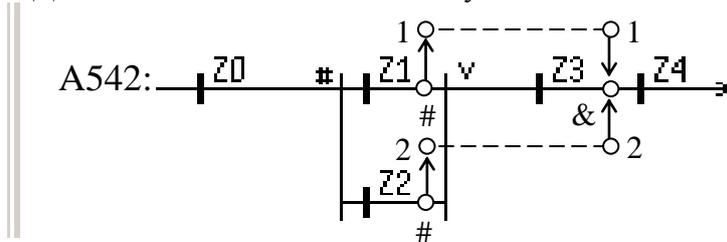
$$A542 = (Z0 - ((Z1 \uparrow^1) \vee (Z2 \uparrow^2)) - Z3 - \downarrow_{\&1,2} - Z4) =$$

$$= Z_0(Z_1 \uparrow^1 \vee Z_2 \uparrow^2) Z_3 \downarrow_{\&1,2} Z_4$$

где $\uparrow^i = \#(\uparrow^i, \rightarrow) = (\uparrow^i \# \rightarrow)$ $\downarrow_{\& i,k} = (\rightarrow, \downarrow^i, \downarrow^k) \& = (\rightarrow \& \downarrow^i \& \downarrow^k)$ – особые дополнительные выходные и входные связи на базе узлов вилки и сборки.

ШСА-3: Штрих-схема алгоритма – многополюсная структура:

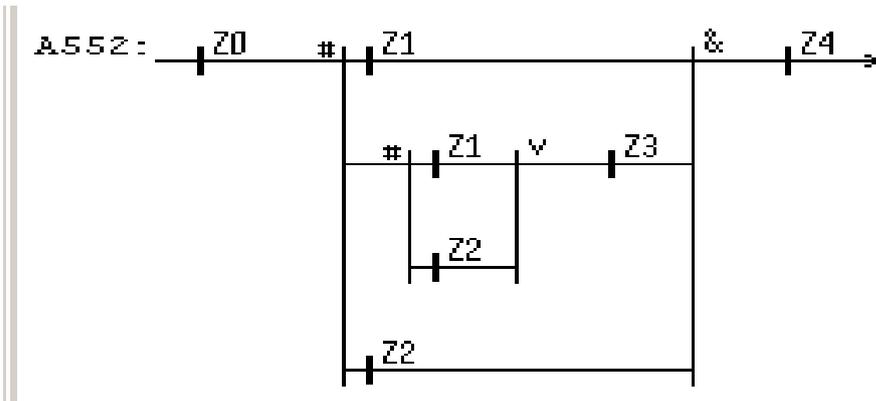
// Дополнительные особые связи указаны тонкими линиями



Эквивалентная двухполюсная структура:

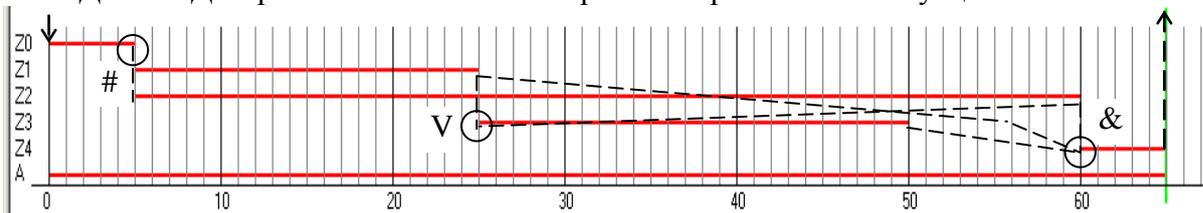
// расщепление цепей

$$A552 = (Z0 - (Z1 \& ((Z1 \vee Z2) - Z3) \& Z2) - Z4)$$



Многополюсные структуры сводятся к эквивалентным двухполюсным структурам, но при этом обычно снижается компактность схем.

ДИА-3: Диаграмма исполнения алгоритма – критическая ситуация:



Имеет место критическая ситуация – затягивание во времени исполнения команды Z2, но обеспечивается нормальное завершение цикла.

Расширения исходных определений

В работе [3] кратко представлена концепция полиморфного синтаксиса базисных параллельных структур: ациклические структуры параллельных алгоритмов жесткого (не переключаемого) типа.

На этой базе возможно определение гибких (переключаемых) структур на основе символики *логических схем алгоритмов* (ЛСА) [4]:

$$I_{pi}(A_k) = I_{pi} A_k = p_i \uparrow_j A_k \downarrow_j = p_i \uparrow_j A_k \downarrow_v^j \quad - \text{структуры типа } \underline{\text{If}} \dots \text{then};$$

$$I_{pi}(A_k, A_r) = p_i \uparrow_j A_k \omega \uparrow_m \downarrow_j A_r \downarrow_m = p_i \uparrow_j A_k \uparrow_m \downarrow_j A_r \downarrow_v^m \quad - \text{типа } \underline{\text{If}} \dots \text{then} \dots \text{else};$$

$$W_{pi} A_k = \downarrow_m p_i \uparrow_j A_k \omega \uparrow_m \downarrow_j = \downarrow_v^m p_i \uparrow_j A_k \uparrow_m \downarrow_j \quad - \text{циклы типа } \underline{\text{While}};$$

$$T_{pi} = \downarrow_j p_i \uparrow_j = \downarrow_v^j p_i \uparrow_j \quad - \text{циклы типа } \underline{\text{Wait}}: \text{ждать (сигнал } p_i) \text{ и т.п.}$$

В работе [3] обозначаются также структуры типа For, For All, For Any (компактные продольные и поперечные свертки однородных структур):

$$A_{i1} - A_{i2} - \dots - A_{in} = L_{k(k \in S)} A_k \quad // S = \{i1, i2, \dots, in\}$$

$$A_{i1} \& A_{i2} \& \dots \& A_{in} = \forall k(k \in S) A_k \quad A_{i1} \vee A_{i2} \vee \dots \vee A_{in} = \exists k(k \in S) A_k$$

3 ЗАКЛЮЧЕНИЕ

В статье отражены исходные аспекты применения языка структурных формул параллельных алгоритмов во взаимосвязи с другими формами представления алгоритмов.

Данный язык поэтапно формируется на основе совмещения и совместной доработки структурных средств двух типов:

- средства типа *системы алгоритмических алгебр* (САА) [5] – формулы алгоритмов аналогичны булевым формулам (но есть специфика): обеспечивается логически наглядная структурная запись алгоритмов, но существуют ограничения применимости выражений булевого типа только для двухполюсных структур параллельных алгоритмов;
- средства типа *параллельных ЛСА* (ПЛСА) [4] – особые стрелочные формулы алгоритмов (с вертикальными стрелками особых связей): обеспечивается описание любых структур, но сложные формулы без явных булевых выражений достаточно громоздки, теряют структурную наглядность и скрывают явную структурную логику.

Совместное применение средств указанных двух типов и их развитие (в целевом приложении) существенно повышает эффективность структурного описания параллельных алгоритмов. На этой основе формируется техника полиморфных структурных построений, представленная здесь в исходных аспектах на простых примерах.

Уже на уровне анализа простых примеров (базовой концептуальной значимости) выявляются нестандартные аспекты параллелизма с завершением по дизъюнкции. Параллельная дизъюнкция алгоритмов теоретически возможна и требует всестороннего освоения. Данный вопрос все чаще обсуждается разными авторами (например, в работе [2]).

Специальной кодировкой узлов структурных схем алгоритмов возможен переход к языку параллельных алгоритмов логического управления (ПРАЛУ) [2]. При этом обеспечивается привязка к специальным сетям Петри, параллельным конечно-автоматным представлениям и общей алгоритмической методологии работы [2].

ЛИТЕРАТУРА

Статьи [1,3] доступны на сайте:

<http://paralg.ucoz.com>: Параллельные алгоритмы и логика

1. Житников А.П. Алгоритмизация производственных информационных технологий. // Управление в сложных системах: Межвуз. научн. сб. – Уфа: УГАТУ, 1996. С. 71 – 80.
2. Закревский А.Д. Параллельные алгоритмы логического управления. – М.: Едиториал УРСС, 2003. – 200 с.
3. Житников А.П. Житников В.П., Шерыхалина Н.М. [Программно-методический комплекс "Параллельные алгоритмы"](#). // Материалы научно-технического совещания "Высокопроизводительные вычислительные ресурсы России: состояние и перспективы развития". – Уфа: УГАТУ, 2003. С. 151 - 161.
4. Лазарев В. Г., Пийль Е. И. Синтез управляющих автоматов. – М.: Энергоатомиздат, 1989. 328 с.
5. Ющенко Е.Л. и др. Многоуровневое структурное проектирование программ. – М.: Финансы и статистика, 1989. 208 с.