

Статья. Библиографические данные:

Житников А.П.
Параллельные и конвейерные алгоритмы робототехнологических модулей.
// Proceedings of the 2nd International Conference
“Intelligent Technologies for Information Processing and Management”, Volume 2, November 10-12,
Ufa, Russia, 2014,
pp. 159-165

ПАРАЛЛЕЛЬНЫЕ И КОНВЕЙЕРНЫЕ АЛГОРИТМЫ РОБОТОТЕХНИЧЕСКИХ МОДУЛЕЙ

Параллельные и конвейерные алгоритмы робототехнологических модулей

А.П. Житников

Уфимский государственный авиационный технический университет

Уфа, Россия

e-mail: paralg@mail.ru

Аннотация¹

Ставится задача массового освоения теории параллельных алгоритмов для решения задач алгоритмизации робототехнологических комплексов. За основу принимается компактная модификация структурных формул и схем алгоритмов типа логических схем алгоритмов.

1. Введение

Работа гибких производственных модулей (ГПМ) и систем (ГПС) и, в частности, робототехнологических комплексов (РТК) представляет собой иерархическую систему дискретных процессов двух типов:

- процессы продвижения, хранения и обработки дискретных потоков продукции (поштучно и партиями) и прочих объектов: деталей, узлов, материалов, инструмента и т.п.;
- процессы управления материальными потоками.

При этом первичную основу составляют последовательные дискретные процессы, то есть последовательности действий во времени. Однако в производстве решающую роль играют *параллельные дискретные процессы* – с совмещением выполняемых действий во времени (в разных видах совмещения). Большое значение имеют межоперационные и внутриоперационные *конвейерные циклические процессы* – с пересечением интервалов смежных итераций циклов во времени (особая форма параллелизма).

С точки зрения организации управления указанные *автоматизированные системы* представляют собой *системы управления*, выполняющие общий *процесс управления* и включающие в себя две подсистемы, которые выполняют два составляющих процесса этого общего процесса управления (рис. 1):

- *управляющая система* или субъект управления (в составе системы управления): выполняется дискретный *управляющий процесс*;

Труды второй международной конференции "Интеллектуальные технологии обработки информации и управления", 10 - 12 ноября, Уфа, Россия, 2014

- *управляемая система* или объект управления (в составе системы управления): выполняется дискретный *управляемый процесс*.

Соответственно этому существуют задачи описания, организации и моделирования дискретных процессов функционирования таких систем с целью их изучения (по документации), исследования и оптимальной (рациональной) их реализации на стадиях проектирования и эксплуатации и т.п.

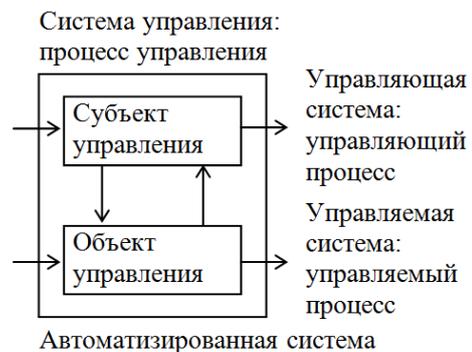


Рис. 1. Управление в автоматизированной системе

2. Проблемы описания работы технологических систем автоматизации

В *традиционной автоматизации* технологических процессов на основе *специальных технологических автоматов и автоматических линий* описание их работы осуществляется применением традиционных временных, путевых и позиционных диаграмм (циклограмм) разного типа и сложности. Для отражения процессов управления используются релейно-контактные и бесконтактные схемы управления, таблицы и диаграммы включений конкретной аппаратуры управления. В целом, так или иначе, решаются вопросы отображения параллелизма процессов и их *синхронной* конвейерной организации, но, в основном, на интуитивном уровне представления и реализации разных факторов параллелизма. К более поздним средствам относятся *дискретные модели* разных типов, в частности:

1) *Модели логического управления* – автоматы комбинационного и последовательностного типа. К последнему типу относятся *конечные автоматы*, представленные обобщенными системами состояний

и переходов. Но они изначально ориентированы только на последовательные дискретные процессы.

2) Их расширение – **параллельные конечные автоматы** (коллективы автоматов). Это достаточно специфичные средства отображения взаимосвязи всевозможных комбинаций состояний параллельных процессов, проблематичные пока для массового практического освоения и применения.

3) **Сетевые методы**, в частности сети Петри. Они изначально ориентированы на моделирование параллельных дискретных процессов. Но их общим недостатком, как и всех дискретных моделей, является необходимость оперирования громоздкими матрицами смежности, переходов т.п., причем, как правило, слабо заполненными. Это средства в основном теоретического значения и анализа разных критических проблем, но не как основной рабочий аппарат описания процессов систем автоматизации.

Гибкие и, в частности, **роботизированные производственные модули и системы** komponуются на основе оборудования универсального типа с программным управлением. Их проектирование и практическое применение связано с необходимостью отражения циклов обработки дискретных потоков продукции, переменных в разных отношениях. При этом неприемлемы жесткие синхронные конвейерные схемы и необходимы гибкие **асинхронные** конвейерные режимы. Применение указанных выше средств осложняется, вплоть до того, что компоновки таких систем конструктивно отображаются чертежами и схемами, а описание их работы выполняется **в словесной форме** [1]. На это

существуют свои причины (подлежащие анализу). Задачи управления универсальным оборудованием реализуются комплектами управляющих программ. Отдельные управляющие программы обеспечивают управление отдельными единицами оборудования с обменом командами между ними (и, возможно, данными) на этапах их взаимодействия. Но проблемой является **связное отображение** общих процессов работы **комплексов взаимодействующего оборудования**, на что также есть свои причины (также подлежащие анализу).

3. Краткая постановка задачи

Управляющие программы являются машинными **алгоритмами** управления оборудованием. В целом автоматизированная система – это **алгоритмическая система управления** дискретными процессами. Для гибких и, в частности, роботизированных производственных модулей и систем актуально **массовое освоение** методов и средств **теории алгоритмов**. Необходимо обеспечение доступности основ теории параллельных (и последовательных) алгоритмов для их массового разработчика и еще более массового пользователя. При этом:

Во-первых, теория алгоритмов весьма обширная и разнообразная (рис.2).

Во-вторых, указанные выше дискретные модели по традиции также относятся к теории алгоритмов, но обладают вполне определенной, хотя и не очень ясной пока отличительной спецификой.

Появляется проблемная задача:



Рис.2. Теория алгоритмов. Два основных направления теории

- анализа общего состояния теории алгоритмов и локализации необходимых методов и средств ее применения;
- выявления соотношения теории алгоритмов с указанными выше дискретными моделями;
- разработка технологий алгоритмизации описания ГПМ, ГПС и, в частности, РТК.

4. Теория алгоритмов

На данное время единой связной общей теории алгоритмов не существует, и она представлена множеством разных частных теорий, направлений, задач, подходов, алгоритмических языков и т.п. Это обстоятельство, его причины и последствия требуют отдельного изложения. Но в данном случае достаточно констатировать, что теория алгоритмов разделена на два направления, ориентированные на принципиально разные цели, задачи, методы, средства и области приложений (рис. 2):

1. **Классическая фундаментальная теория алгоритмов**, именуемая также как **теория вычислимости** – это исторически первая теория, ориентированная на решение фундаментальных проблем математики. По устаревшей традиции она обычно именуется общим термином "теория алгоритмов" (что сбивает общую ориентировку в организации изучения теории алгоритмов):

сейчас (с 2000-х гг.) она повсеместно изучается в рамках учебных дисциплин по дискретной математике, математической логике и теории алгоритмов на первых-вторых курсах обучения.

2. **Неклассическая прикладная теория** – это, по своему существу, **структурная теория алгоритмов** (что очевидно по комментариям на рис. 2), хотя она так реже именуется:

изучается она фрагментарно и, обычно, не в курсах теории алгоритмов, а в рамках разных специальных дисциплин, и, как правило, на более поздних курсах обучения (хотя целесообразно ее раннее изучение).

Ее структурный характер определяют, в частности, следующие ключевые структурные аспекты:

- первичное ее структурное разделение на теорию традиционных – **последовательных** алгоритмов и теорию **параллельных** алгоритмов с тенденцией их объединения в обобщенную теорию параллельных (и, в частности, последовательных) алгоритмов;
- существование ключевых исходных структурных понятий **потока управления** и **потока данных** алгоритма (рис. 3), которые впервые были выявлены в программах.

Дискретные модели имеют следующее соотношение с алгоритмами и теорией алгоритмов:

Во-первых, **дискретные модели** возникли в теории комбинационных и последовательных

логических схем (автоматов), в принципе **не основаны** на понятии алгоритма, и в их составе это понятие конструктивно **не работает**.

Во-вторых, дискретные модели могут использоваться **для моделирования** потока данных и потока управления алгоритма – по отдельности и в их взаимосвязи, но сами дискретные модели такой структуры и таких понятий **не имеют**. При этом модель всегда в чем-то отличается от моделируемой сущности, и следует четко выяснять, какие разные аспекты алгоритмов моделируют разные дискретные модели (что, в целом, пока не очень ясно).

В-третьих, **алгоритмы** могут использоваться для **программной реализации дискретных моделей**.

Такие дискретные модели целесообразно рассматривать как модельный инструментальный прикладной теории алгоритмов (рис. 2) наряду, например, с формальными грамматиками (которые сами используют автоматные модели).

5. Поток управления и поток данных

Алгоритм – это предписание для некоторой алгоритмической системы управления, которая, так или иначе, отображается в составе текстов и контекстов алгоритмов. Алгоритмы содержат две структуры, именуемые как потоки (рис. 3) [2 – 4]:

- **поток управления** алгоритма – это управляющая структура алгоритма (отражение управляющей подсистемы рис. 2 алгоритмической системы управления):

точнее, представлен (так или иначе отражается) некоторый **канал управления** (с потоком команд управления в его составе), в котором организуется управление последовательным или параллельным во времени порядком выполнения (и параметрами) заданных комплексов действий;

- **поток данных** алгоритма – это управляемая структура алгоритма (отражение управляемой подсистемы рис. 2 общей алгоритмической системы управления):

точнее, представлен (так или иначе отражается) некоторый **канал данных** (с потоком данных в его составе), где выполняется размещение, хранение, продвижение и обработка данных.



Рис. 3. Структура управления алгоритма

При этом в технических приложениях алгоритмов понятия данных и их потоков требует обобщения на **информационные** и **материальные** данные и потоки данных (заготовки, детали, узлы, инструмент и т.п.).

6. Базовые алгоритмические средства

За основу алгоритмического описания параллельных (и последовательных) дискретных процессов ГПМ, ГПС и РТК целесообразно принять собственные средства прикладной теории алгоритмов, а не дискретные модели, которые могут привлекаться в случае дополнительной необходимости. Все основные частные теории алгоритмов в принципе **эквивалентны между собой** и различаются своими достоинствами и недостатками в разных условиях применения, полнотой, специализацией на удобство отражения разных алгоритмических аспектов и т.п.

Первоосновой всех таких средств по ряду причин следует считать так называемые **логические схемы алгоритмов** (ЛСА), которые появились в программировании еще в начале 50-х гг. (операторный метод Ляпунова А.А.), получили технические приложения, и, позднее, были дополнены так называемыми **параллельными логическими схемами алгоритмов** (ПЛСА) [5]. На самом деле это особые (структурные) формулы алгоритмов стрелочного типа – с вертикальными стрелками условных и безусловных переходов.

ЛСА и ПЛСА первоначально – это **высокоуровневый** алгоритмический язык описания программ в области **низкоуровневого** системного программирования и микропрограммирования. Их достоинством является возможность точного отображения низкоуровневых структур программного управления. Недостатком их является громоздкость описания алгоритмов как следствие явной опоры на низкоуровневые механизмы передачи управления и параллелизма операторов алгоритма.

Но целесообразно не отказываться от ЛСА и ПЛСА, а вводить на их фундаментальной основе компактные высокоуровневые средства описания алгоритмов и совместно их использовать при необходимости [6].

Например, для ветвлений последовательных процессов возможна бесстрелочная запись ЛСА типа:

$$p \uparrow^1 A \downarrow^1 = IpTA \quad // \text{ If } p \text{ Then } A;$$

$$p \uparrow^1 A_1 \omega \uparrow^2 \downarrow^1 A_2 \downarrow^2 = IpTA_1EA_2 \quad // \text{ If } p \text{ Then } A_1 \text{ Else } A_2.$$

Для параллельного выполнения операторов возможна компактная запись ПЛСА типа (рис. 3):

$$A2 = A_2 = R \uparrow^1 \uparrow^2 \uparrow^3 \downarrow^1 Z_1 \omega \uparrow^4 \downarrow^2 Z_1 \omega \uparrow^5 \downarrow^3 Z_1 \omega \uparrow^6 \downarrow^4 \downarrow^5 \downarrow^6 S = \\ = (Z_1 \parallel Z_2 \parallel Z_3) = Z_1 \parallel Z_2 \parallel Z_3.$$

Эти аспекты уточняются далее на примерах.

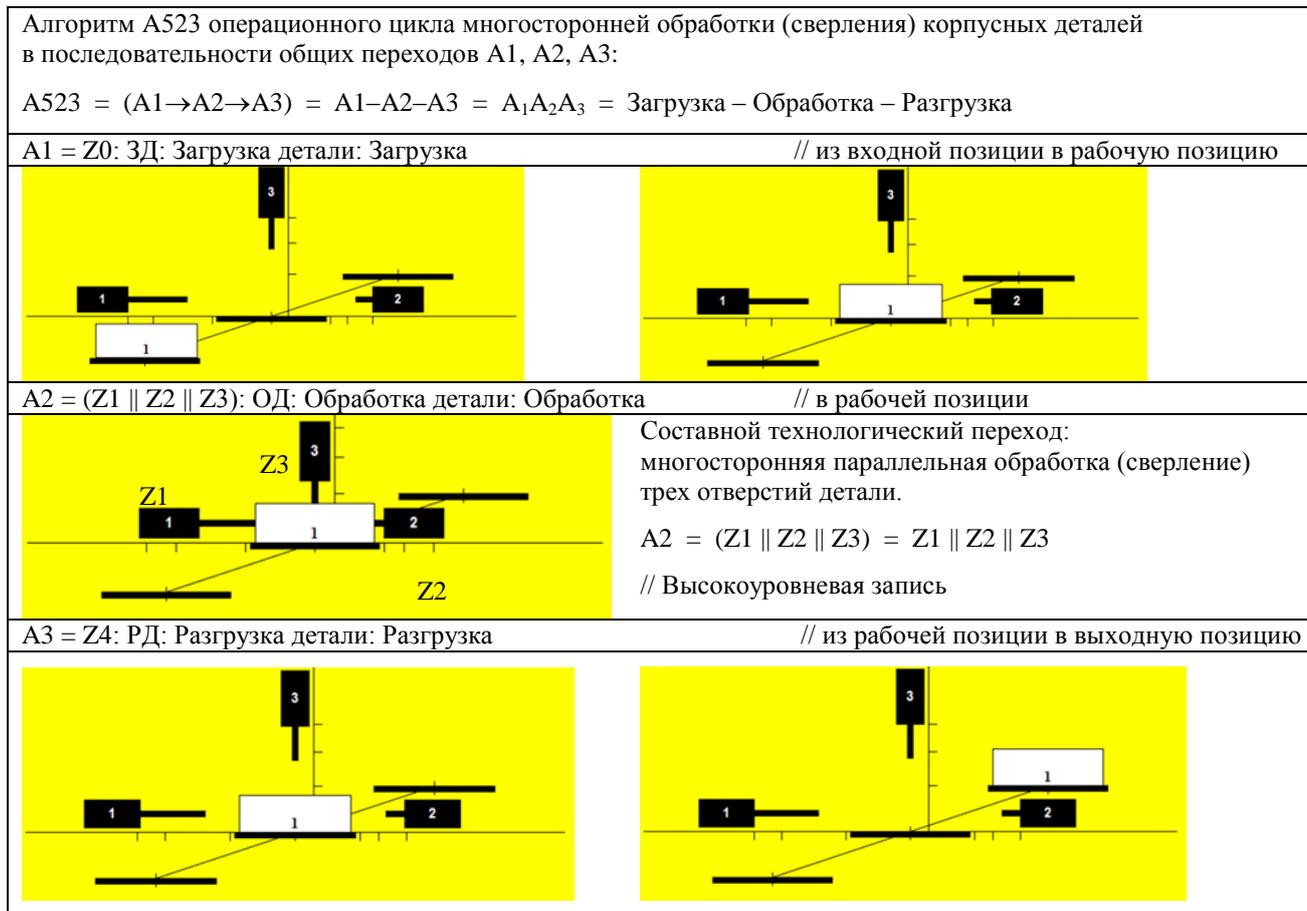


Рис. 4. Мнемосхемы автоматизированной технологической системы. Операционный цикл

Табл. 1. Аналитическое описание структуры алгоритма: высокоуровневая модификация ПЛСА

СФА: Структурная формула алгоритма / ИнФ: Инфиксная форма
$A523 = (A1-A2-A3) = (Z0-(Z1 \parallel Z2 \parallel Z3)-Z4) = (Z0-(Z1 \text{ RS } Z2 \text{ RS } Z3)-Z4) = (Z0-(Z1 \# \& Z2 \# \& Z3)-Z4)$
СФА: Структурная формула алгоритма / ИнПрПоФ: Инфиксно-префиксно-постфиксная форма
$A523 = (Z0-(Z1 \# \& Z2 \# \& Z3)-Z4) \Rightarrow Z0 \# (Z1, Z2, Z3) \& -Z4 = Z0 \# (Z1, Z2, Z3) \& -Z4$
<p>$- = \rightarrow$: служебный оператор секвенции – последовательной связи (простых и составных) рабочих операторов</p> <p>$\parallel = \text{RS} = \# \&$: парный служебный оператор параллели – параллельной связи рабочих операторов</p> <p>$\# = R \sim R \uparrow^1 \uparrow^2 \uparrow^3$: служебный оператор разделения (дивергенции) потоков – вилка потоков (fork)</p> <p>$\& = S \sim \downarrow^4 \downarrow^5 \downarrow^6 S$: служебный оператор соединения (конвергенции) потоков – сборка потоков (join): применяется сборка S по конъюнкции $S = \&$ (выход на оператор Z4 при завершении всех потоков Z1 и Z2 и Z3)</p>

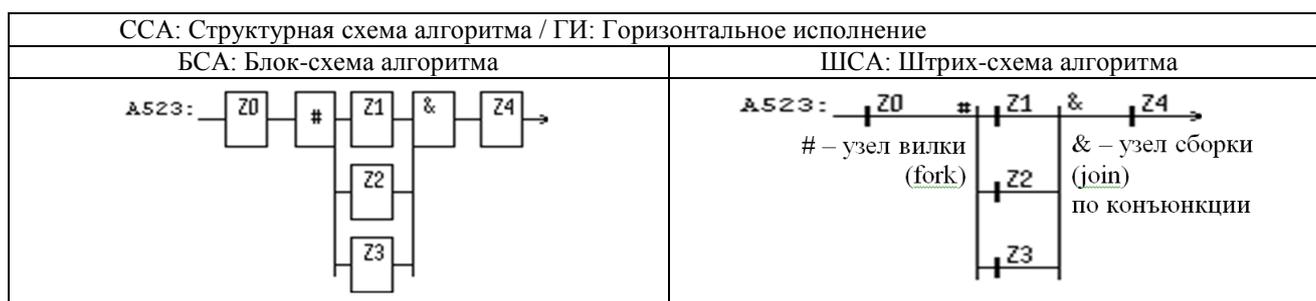


Рис. 5. Схемное описание структуры алгоритма

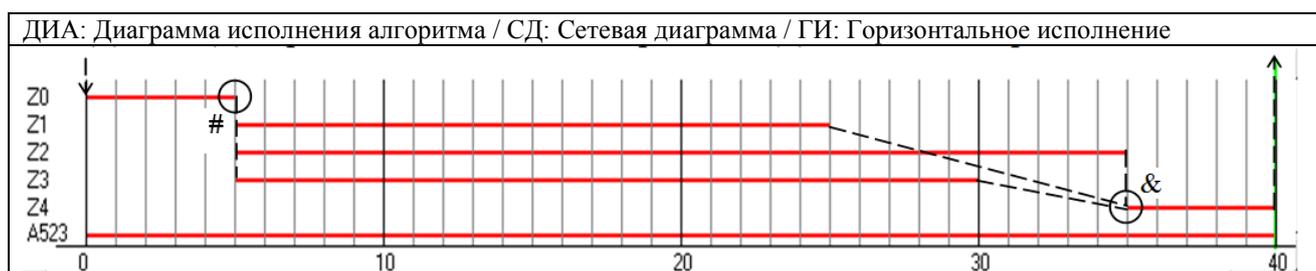


Рис. 6. Временная диаграмма операционного цикла

7. Пример 1: РТК обработки резанием

На рис. 4 представлена мнемосхема (видеоряд) операционного цикла для автоматизированной технологической системы, как подготовка формирования РТК. Далее приводится аналитическое и графическое описание структуры алгоритма управления (табл. 1, рис.5) и диаграмма параллельного процесса его исполнения (рис.6). Для принятых параметров длительности всех действий обеспечивается коэффициент ускорения (по соотношению длительностей последовательной T1 и параллельно T3 обработки детали): $k = T1 / T3 =$

$$= (5+20+30+25+5) / (5+\max(20, 30, 25)+5) = 2,125.$$

Это **внутрицикловое ускорение** операционной обработки одной детали. Возможно существенное **дополнительное ускорение** обработки потока деталей путем допустимого перекрытия во времени интервалов смежных итераций операционного цикла. Это требует организации **внутриоперационного конвейерного режима** обработки потока деталей:

загрузка очередной детали начинается до завершения разгрузки и даже обработки текущей в потоке детали. Автоматически реализуется **последовательно-параллельная обработка** потока деталей на автоматизированном рабочем месте (типа рис. 7).

Применение **обслуживающих роботов** на переходах загрузки и разгрузки деталей обеспечивает такую возможность, причем в **гибком** конвейерном режиме. Но при этом конвейерный режим требует обеспечения блокировок аварийного доступа роботов в зону рабочей позиции технологической системы.

Организация конвейерного режима целесообразна для относительно коротких по длительности переходов обработки по отношению к переходам загрузки-разгрузки (например, при сверлении малых приборных корпусов из легких сплавов).

Предельным случаем является организация конвейерного режима РТК для операции штамповки с очень коротким переходом обработки (один двойной ход ползуна). Далее приводится пример такого РТК.

8. Пример 2: Штамповочный РТК

В статье [7] (и других связанных с ней статьях) достаточно подробно излагается постановка задачи и два метода синтеза асинхронного конвейерного алгоритма штамповочного РТК – с реализацией максимальной пропускной способности дискретного (позиционного) технологического канала. На рис. 9 представлена предельно упрощенная (и развернутая на плоскости) концептуальная динамическая мнемосхема РТК (выборочный видеоряд).

На рис. 7 представлена обобщенная линейная диаграмма исполнения конвейерного алгоритма без детализации состава действий и их причинно-следственных связей. Точнее, это обобщение детальной сетевой диаграммы (здесь не приводится).

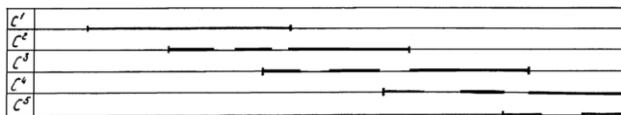


Рис. 7. Линейная временная диаграмма

Наглядно показано перекрытие интервалов времени исполнения пяти первых смежных итераций ($C^1..C^5$) общего операционного цикла обработки потока деталей для некоторой партии деталей.

Итерация C^1 первой детали потока проходит без прерываний. В последующих итерациях появляются прерывания их исполнения, которые нарастают по длительности и стабилизируются к 5-й итерации C^5 .

Прерывания создаются блокировками аварийных ситуаций – столкновений рук роботов с ползуном прессы и между собой. Это задача разделения времени доступа к общим ресурсам процессов – приостановки движения (схватов) рук роботов около рабочей зоны прессы, если она занята, до ее освобождения.

На рис. 8 показана схема конвейерного алгоритма и его структурная формула типа ПЛСА (в некоторой модификации). Последняя детальная запись – это низкоуровневый вариант ПЛСА. Это точная формула, но, очевидно, очень *громоздкая* и практически

нечитабельная для стороннего технического читателя (не ее автора). Наличие схемы улучшает ее понимание, но актуально *повышение смыслового уровня* описания конвейерного алгоритма.

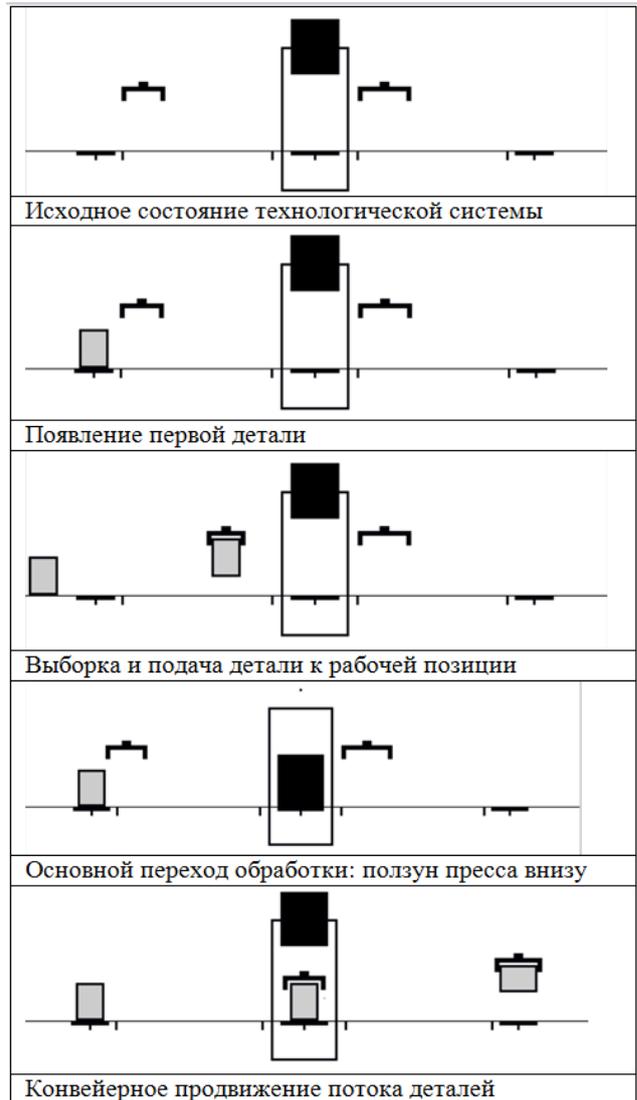


Рис. 9. Мнемосхема штамповочного РТК

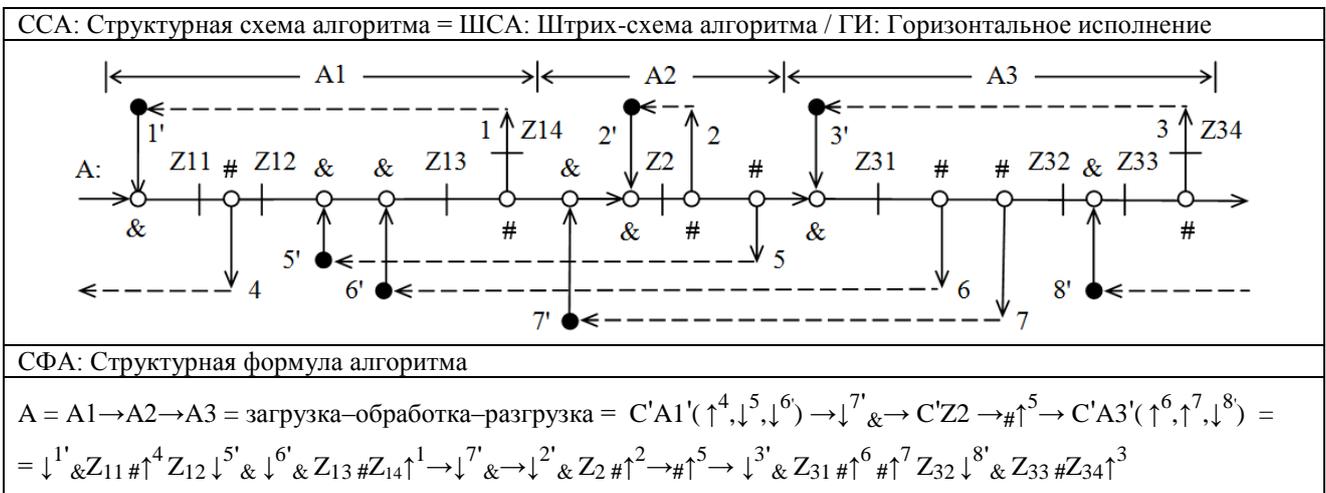


Рис. 8. Асинхронный конвейерный алгоритм

Специфика конвейерного алгоритма

Особенностью конвейерной схемы является наличие системы зацепленных контуров с замыканием по конъюнкции (с единичными начальными условиями связи замыкания – отмечены темными кружками и штрихами номеров входных связей). Они реализуют сходные блокировки двух типов по назначению:

- исключение повторного запуска для частных циклов оборудования до их завершения:

это три верхних контура на схеме, которые представлены *компактной* записью типа $C'A_i'$, где C' – оператор специального цикла, A_i' – тело цикла;

- исключение начала действий предыдущего оборудования в опасной зоне до завершения действий последующего оборудования в этой зоне (по ходу операционного процесса):

это полные и неполные нижние контуры, которые представлены в структурной формуле исходящими и входящими нумерованными стрелками.

Это *более компактное* совместное применение первичных и вторичных средств ЛСА (ПЛСА). Далее вертикальные стрелочные элементы могут заменяться переменными событиями потока управления с операторами порождения и обнаружения событий (по аналогии с событийно-ориентированными методами высокоуровневого программирования), привлекаться высокоуровневые средства других теорий и т.п.

9. Заключение

1. Показана большая актуальность массового освоения методов и средств теории алгоритмов для описания управления параллельными (и последовательными) дискретными процессами гибких производственных модулей и систем и, в частности, робототехнологических комплексов.
2. Приводится общий состав теории алгоритмов. Задается приоритет освоения неклассической прикладной (структурной) теории алгоритмов (относительно классической теории алгоритмов, то есть теории вычислимости).
3. В составе неклассической прикладной теории алгоритмов за первичную основу принимаются структурные формулы и схемы типа последовательных (ЛСА) и параллельных (ПЛСА) логических схем алгоритмов. Выделено их фундаментальное достоинство – отражение базовых низкоуровневых структур условных и безусловных переходов и параллелизма потока управления алгоритма. С этим связывается и их недостаток – громоздкое описание структуры потоков управления алгоритмов (примерно аналогично применению множеств семафоров Дейкстры для низкоуровневого описания синхронизации многопоточных программ).
4. Поставлена задача и демонстрируются реальные возможности преодоления этого недостатка путем

введения производных высокоуровневых средств описания структуры алгоритмов и их использования совместно с первичными структурными средствами при необходимости.

5. Кратко представлены примеры параллельных и конвейерных алгоритмов робототехнологических комплексов типа гибких производственных модулей. Эти подходы применимы и для роботизированных комплексов на уровне гибких производственных систем и, в частности, гибких автоматических линий и складских роботов.
6. На этих примерах четко отражается специфика алгоритмов управления материальными объектами и потоками:
 - явное алгоритмическое описание потока управления алгоритма при косвенном отражении потока (материальных) данных в технической документации и мнемосхемах (а также циклограммах их работы);
 - для описания потока данных могут применяться, в частности, дискретные модели, но осмотрительно (с обоснованием необходимости этого) – без чрезмерного теоретизирования практических задач.

Список используемых источников

1. Козырев Ю.Г. Гибкие производственные системы. Справочник: справочное издание. – М: КНОРУС, 2015. – 364 с.
2. Житников А.П. Потоки управления и потоки данных параллельных (и последовательных) алгоритмов. // Сбор. матер. XIV Междунар. науч.-практ. конф. «Информ. и коммуникац. технологии в образовании». – Борисоглебск: БГПИ, 2013. – С. 45-61.
3. Житников А.П. Структурный алгоритмический анализ простой вычислительной задачи. // Там же. – С. 62-77.
4. Житников А.П. Структурная формализация потоков управления и потоков данных параллельных (и последовательных) алгоритмов. // Там же. С. 78-100.
5. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. – М.: Энергоиздат, 1989. – 328 с.
6. Житников А.П. Базовая структурная семантика параллельных алгоритмов. // Принятие решений в условиях неопределенности: Межвуз. научн. сб. – Уфа: УГАТУ, 2008. – С. 41-50.
7. Житников А.П. Синтез асинхронного конвейерного алгоритма робототехнологического модуля. // 2-я междунар. конф. "Инф. технологии интеллект. поддержки принятия решений". Т.3: Междунар. науч. изд. – Уфа: УГАТУ, 2014. – С. 265-273.