

Статья. Библиографические данные:

Житников А. П.
Синтез асинхронного конвейерного алгоритма робототехнологического модуля.
/ Материалы II Международной конференции "Информационные технологии интеллектуальной поддержки принятия решений" (ITIDS'2014). Т. 3.
– Уфа: УГАТУ, 2014. С. 265 – 273.

Приводится копия статьи без дополнительной обработки.
К данной копии статьи добавлены следующие графические приложения
(указаны гиперссылки):

Приложение 1.

Модельная программа РТК: Роботизированный технологический комплекс.

Приложение 2.

Режим одиночных циклов (не конвейерный режим):
цикл продвижения и обработки одной детали.

Приложение 3.

Конвейерный режим: обработка потока деталей.

Синтез асинхронного конвейерного алгоритма робототехнологического модуля

А.П. Житников

Уфимский государственный авиационный технический университет

Уфа, Россия

e-mail: paralg@mail.ru

Аннотация¹

Кратко излагается метод синтеза конвейерного алгоритма управления обработкой потоков деталей для типовой конфигурации гибкого производственного модуля с двумя обслуживающими промышленными роботами на загрузке и разгрузке деталей.

1. Введение

Проектирование и исследование конфигураций роботизированных технологических комплексов на уровне гибких производственных модулей и систем связано с необходимостью моделирования циклов обработки дискретных потоков продукции (поштучно и партиями деталей). В его основе лежат некоторые *параллельные*, в общем случае, *алгоритмы* действия и взаимодействия основного и вспомогательного технологического оборудования, их агрегатов, технологической оснастки и т.п.

Исходными объектами моделирования являются гибкие производственные модули – технологические автоматизированные рабочие места, выполняющие отдельные технологические операции. На этой основе возможно моделирование многомодульных гибких производственных систем с учетом технологических маршрутов, межоперационных передач, накопителей и т.п. Могут быть разные конфигурации применения внутриоперационных, межоперационных и общих (совмещенных) обслуживающих роботов, что определяется разными критериями их выбора. При этом ведущим исходным параметром является *соотношение длительности* основных циклов обработки деталей и вспомогательных циклов обслуживания обработки (загрузки, разгрузки и т.п.).

В данной статье рассматривается ситуация: относительно *короткие основные циклы* обработки при относительно *длительных вспомогательных*

циклах загрузки и разгрузки деталей, например, в автоматизации штамповки (рис. 1, рис. 8).

При этом целесообразно совмещение (перекрывание) смежных операционных циклов с организацией внутриоперационных *конвейерных режимов*. В массовом производстве широко используются конвейерные процессы с выравниванием интервалов совмещаемых этапов во времени, что обеспечивает жесткую *синхронную конвейеризацию* традиционными техническими средствами. Для гибких модулей и систем с переменными параметрами продукции и ее обработки это, в общем случае, невозможно, и необходимы *асинхронные конвейерные режимы и алгоритмы*.

2. Краткая постановка задачи

Принимается алгоритмическая задача моделирования работы гибкого производственного модуля *базовой типовой конфигурации* (рис. 1) с организацией внутриоперационного *конвейерного режима обработки* внутриоперационного потока деталей.

2.1. Объект алгоритмизации

Базовая конфигурация модуля (рис. 1) обобщенно представляет собой широкий класс конкретных геометрических и конструктивных схем компоновок гибкого производственного модуля. Она включает в себя следующее технологическое оборудование:

1) Два обслуживающих промышленных робота:

ПР1: Промышленный робот 1 для загрузки рабочей позиции p_2 модуля из входной обменной позиции p_1 модуля (однорукий робот с одним схватом).

ПР2: Промышленный робот 2 для разгрузки рабочей позиции p_2 модуля в выходную обменную позицию p_3 модуля (также однорукий робот с одним схватом).

2) Одна единица основного технологического оборудования *с коротким циклом обработки* деталей, например штамповочным прессом (ШП).

ШП выполняет простые рабочие циклы – двойные хода ползуна (вниз и вверх) с достаточно высокой возможной частотой (порядка, например, 60 двойных

Груды второй международной конференции "Информационные технологии интеллектуальной поддержки принятия решений", 18 - 21 мая, Уфа, Россия, 2014

ходов в минуту, то есть 1-го двойного хода в секунду). Выполняется, в данном случае, обработка деталей давлением в рабочей позиции p_2 модуля.

В целом гибкий технологический модуль имеет **дискретный позиционный технологический канал**, включающий 3 последовательно связанные **основные позиции** для (поштучного) размещения деталей:

$$p_1 \rightarrow p_2 \rightarrow p_3.$$

На входе и выходе технологического канала модуля условно показаны (пунктиром) внешние межоперационные транспортные средства Tr_1 , Tr_2 для (поштучной) загрузки входной обменной позиции p_1 и разгрузки выходной обменной позиции p_3 соответственно. Это могут быть внешние обслуживающие роботы или специальные устройства. При этом рассматривается открытая система основных позиций технологического канала:

$$\rightarrow p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow.$$

Данные позиции p_1 , p_2 , p_3 технологического канала модуля связаны **гибкими внутриоперационными транспортными связями** (рис. 1) – траекториями перемещения деталей в схватах роботов. Траектории деталей в технологическом канале помимо указанных основных позиций p_i , $i = 1..3$ содержат **дополнительные контрольные позиции** до входа p_i' и после выхода p_i'' основных позиций:

$$\rightarrow p_1' \rightarrow p_1 \rightarrow p_1'' \rightarrow \underline{p_2' \rightarrow p_2 \rightarrow p_2''} \rightarrow p_3' \rightarrow p_3 \rightarrow p_3'' \rightarrow.$$

2.2. Режимы обработки потока деталей

При наличии быстродействующих пневматических роботов длительность основного цикла обработки детали в рабочей позиции p_2 существенно меньше длительности вспомогательных циклов обслуживания для перемещения деталей между позициями $p_1 \rightarrow p_2$ и $p_2 \rightarrow p_3$. При использовании электромеханических или гидравлических приводов роботов эта разница, как правило, будет значительно больше.

В данном случае **обычный режим** поштучной обработки потока деталей в технологическом канале модуля (от начала и до конца) будет заведомо нерациональным:

неизбежны большие внутриоперационные простои основного оборудования (ШП) на ожидание циклов загрузки $p_1 \rightarrow p_2$ и разгрузки $p_2 \rightarrow p_3$ деталей (рис. 1).

Целесообразна организация достаточно проблемного **внутриоперационного конвейерного режима** обработки потока деталей с одновременным продвижением в технологическом канале нескольких деталей на разных этапах общего цикла их продвижения. При этом, в принципе, возможно обеспечение **высокого темпа обработки** деталей модулем с **минимизацией простоев** основного оборудования – только на ожидание (рис. 1):

- этапов $p_2' \rightarrow p_2$ укладки деталей в рабочую позицию p_2 (в циклах ее загрузки $p_1 \rightarrow p_2$);
- этапов $p_2 \rightarrow p_2''$ выборки деталей из рабочей позиции p_2 (в циклах ее разгрузки $p_2 \rightarrow p_3$).

Очевидно, что любые производные конфигурации модуля с совмещением двух роботов в один двурукий робот или, тем более, однорукий робот (с одним или двумя схватами) будут только усугублять общую ситуацию и ее проблематику.

Необходим **анализ специфики** внутриоперационного конвейерного режима для базовой конфигурации гибкого модуля и **предложение корректных способов** построения (синтеза) общего асинхронного конвейерного алгоритма управления модулем.

3. Проблемный конвейерный режим

Внутриоперационный конвейерный режим обработки потока деталей в канале модуля означает:

параллельное (во времени) выполнение модулем двух и более смежных итераций операционных циклов продвижения отдельных деталей **со сдвигом** итераций циклов во времени по начальным этапам.

Для асинхронного режима возникает проблема типа: **взаимное исключение** доступа параллельных процессов (итераций циклов) к общим ресурсам.

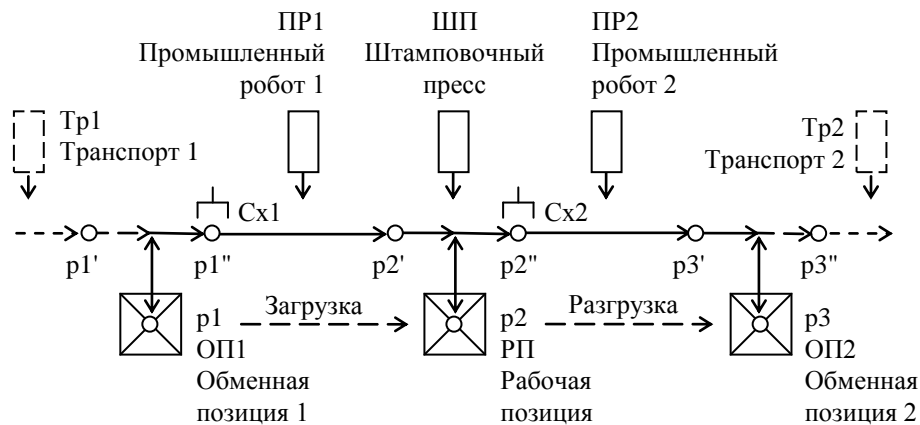


Рис. 1. Базовая конфигурация гибкого роботизированного модуля

В данном случае общими ресурсами параллельных цикловых процессов (или, точнее, выполняющих их объектов) являются **зоны основных позиций** по технологическому каналу гибкого модуля:

ОП1: Входная обменная позиция 1 – исключается одновременный доступ схвата Сх1 загрузочного робота ПР1 и средств внешнего транспорта Тр1.

ОП2: Выходная обменная позиция 2 – исключается одновременный доступ схвата Сх2 разгрузочного робота ПР2 и средств внешнего транспорта Тр2.

РП: Рабочая позиция – попарно исключается одновременный доступ 3-х объектов: схвата Сх1 загрузочного робота ПР1, схвата Сх2 разгрузочного робота ПР2 и ползуна штамповочного прессы ШП.

В классической работе [1] рассматриваются общие принципы организации (параллельно выполняемых) **взаимодействующих последовательных процессов**, включая вопросы их взаимного исключения по общим ресурсам. Рассматриваются так называемые транспортеры или конвейеры (от английского слова convey – передавать, перевозить) – последовательно связанные процессы с передачей данных между ними по мере готовности. Но асинхронный конвейерный параллелизм со взаимным исключением процессов систематически не рассматривается.

В вычислительной технике широко используется конвейерная обработка потоков команд в процессоре с потактовой синхронизацией последовательности его аппаратных компонент (параллельно выполняющих фазы считывания, декодирования, выполнения кодов операций и т.д.). В программировании приводятся разные задачи конвейерной обработки данных в последовательности обрабатывающих программных и аппаратных средств, как правило, на уровне общих положений или их программной реализации конкретными языками низкого или высокого уровня.

Далее этот вопрос излагается в алгоритмической специфике робототехнологической задачи.

4. Прототип излагаемой задачи

В статьях [2, 3] приводится схема конвейерного алгоритма аналогичного модуля (конкретной геометрической конфигурации с поворотными руками роботов) и **графический (схемный) метод** построения структуры конвейерного алгоритма [2], включающий в себя:

1) Подробное **описание конфигурации** модуля в нескольких схемах (с табличными спецификациями графических элементов) и детальную общую позиционную диаграмму модуля, включая:

- **позиционные циклы** состояний (схватов) роботов и ползуна штамповочного прессы;
- **блокировочные связи** взаимного исключения их доступа к общим основным позициям.

2) Построение по исходной диаграмме **отдельных схем алгоритмов** (типа компактных штрих-схем, производных от блок-схем) для циклов работы обслуживающих роботов и ползуна прессы, включая: входы и выходы их внешних связей – блокировок взаимного исключения доступа к общим позициям.

3) **Преобразование контуров** цикловых алгоритмов (относительно входов и выходов внешних связей) и их **последовательное объединение** в общий конвейерный алгоритм с замыканием всех блокировочных связей взаимного исключения.

4) Построение и краткий анализ **сетевой временной диаграммы** исполнения конвейерного алгоритма.

В данной статье в рамках указанного общего метода построений приводится развитие его основного пункта (п. 3). Кратко излагается:

позитивный схемно-аналитический синтез асинхронного конвейерного алгоритма работы модуля методом (восходящей) индукции – на основе исходного обобщенного описания типовой конфигурации модуля (рис. 1).

5. Общий порядок построений

5.1. Условные обозначения

Условные литерные обозначения

Далее (табл. 1) приводятся условные литерные обозначения (УЛО) алгоритмов и их команд, где:

$A(i) = A_i$ – операторы алгоритмов;

$Z(i) = Z_i$ – операторы команд алгоритмов;

i – некоторые индексные выражения (возможно пустые): цифровые, буквенные, буквенно-цифровые, арифметические и т.п.

Дополнительно используются УЛО служебных структурных операций (для комбинирования основных операторов алгоритмов и команд), которые далее поясняются по ходу их применения.

УЛО используются для обозначения графических элементов *структурных схем алгоритмов* (ССА, рис. 2 – рис. 7) и для построения *структурных формул алгоритмов* (СФА, рис. 2 – рис. 7).

Условные графические обозначения

В используемой здесь системе анализа алгоритмов для графического представления алгоритмов в общем случае применяются структурные схемы алгоритмов (ССА) следующих типов:

блок-схемы алгоритмов (ССА = БСА), строго производные от них компактные *штрих-схемы алгоритмов* (ССА = ШСА) и граф-схемы алгоритмов (ССА = ГСА) разных стилей и видов исполнения, в частности: ВИ: *Вертикальное исполнение*, ГИ: *Горизонтальное исполнение* и т.п.

В данной статье достаточным является использование компактных ШСА/ГИ: штрих-схем алгоритмов горизонтального исполнения (рис 2 – рис. 7).

Используемые условные графические обозначения (УГО) для структурных элементов схем алгоритмов являются достаточно очевидными на представленных далее штрих-схемах (и отдельно не приводятся):

- элементарные прямоугольные блоки блок-схем заменяются на штрих-схемах короткими поперечными штрихами;
- элементарные блоки со многими входами или выходами (в частности, для узлов разделения # и соединения $o = \&$ потоков управления) могут иметь продленные линии боковин, которые на штрих-схемах превращаются в (неограниченно) продленные поперечные штрихи (рис. 3/1, рис. 5, 6);
- такие узлы могут для компактности отображаться в производных формах схем малыми светлыми или темными кружками (рис. 3/2, рис. 4 – 7).

Табл. 1. УЛО: Условные литерные обозначения для операторов алгоритмов и команд

Обозначение	Наименование		Примечания (примерное содержание действий)
	Краткое	Полное наименование	
Операторы алгоритмов			
A		Оператор общего алгоритма (итерации) операционного цикла работы модуля	Продвижение и обработка потока деталей в технологическом канале
A1	Загрузка	Оператор циклового алгоритма загрузочного обслуживающего робота ПР1	Загрузка детали в рабочую поз. p2 из входной обменной поз. p1
A2	Обработка	Оператор циклового алгоритма штамповочного пресса ШП	Обработка (штамповка) детали: в рабочей поз. p2
A3	Разгрузка	Операторы циклового алгоритма разгрузочного обслуживающего робота ПР2	Разгрузка детали из рабочей поз. p2 в выходную обменную поз. p3
Операторы команд алгоритмов			
Алгоритм загрузки деталей A1: Обслуживающий робот ПР1			
Z11	Выборка	Переход: Выборка детали из входной обменной позиции ОП1: p1	Ход вперед, ход вниз схвата Сх1, зажим детали, ход вверх, ход назад в поз. p1" с деталью
Z12	Подача	Переход: Подача детали к штамповочному прессу ШП	Перемещение схвата Сх1 с деталью из поз. p1" в поз. p2'
Z13	Укладка	Переход: Укладка (установка) детали в рабочую позицию РП: p2	Ход вперед, ход вниз схвата Сх2 с деталью, разжим детали, ход вверх, ход назад в поз. p2' (без детали)
Z14	Отвод	Переход: Отвод схвата Сх1 робота ПР1 от штамповочного пресса ШП	Возврат схвата Сх1 из поз. p2' в поз. p1" (без детали)
Алгоритм обработки деталей A2			
Z2	Обработка	Переход: Обработка детали в рабочей позиции РП: p2 штамповочного пресса ШП	Двойной ход ползуна: вниз в поз. p2, вверх от поз. p2
Алгоритм разгрузки деталей A3: Обслуживающий робот ПР2			
Z21	Выборка	Переход: Выборка детали из рабочей позиции РП: p2 штамповочного пресса ШП	Ход вперед, ход вниз схвата Сх2, зажим детали, ход вверх, ход назад в поз. p2" с деталью
Z22	Подача	Переход: Подача детали от штамповочного пресса ШП	Перемещение схвата Сх2 с деталью из поз. p2" в поз. p3'
Z23	Укладка	Переход: Укладка (установка) детали в выходную обменную позицию ОП2: p3	Ход вперед, ход вниз схвата Сх2 с деталью, разжим детали, ход вверх, ход назад в поз. p2' (без детали)
Z24	Отвод	Переход: Отвод схвата Сх2 робота ПР2 к штамповочному прессу ШП	Возврат схвата Сх2 из поз. p3' в поз. p2" (без детали)

5.2. Исходный алгоритм

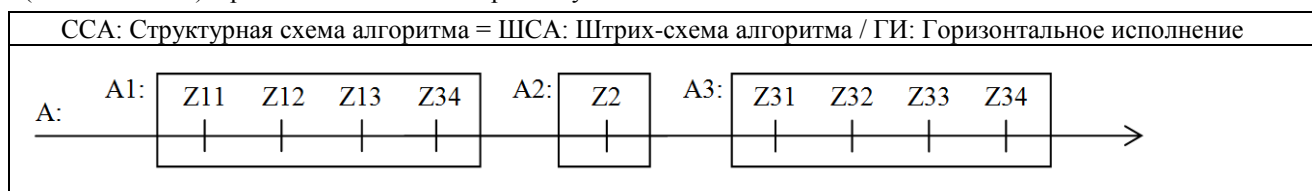
Первичная последовательная структура

На рис. 2 представлена тривиальная структура исходного последовательного алгоритма А обработки отдельных деталей, образующая простую последовательность составляющих алгоритмов A1→A2→A3. Алгоритмы А, А1..А3 представляют собой простую последовательность операторов. Секвенция → (последовательная связь) операторов в явной инфиксной форме (ИнФ) может обозначаться (и именоваться) в разных источниках по-разному:

$A_i \rightarrow A_j = A_i - A_j = A_i * A_j$ ("произведение" операторов).

В так называемых *логических схемах алгоритмов* (ЛСА) и *параллельных логических схемах алгоритмов* (ПЛСА) [4] (особые структурные формулы) секвенция операторов используется в неявной инфиксной форме (СФА/НИнФ, рис.2-7):

$A_i \rightarrow A_j = A_i A_j = A_i A_j$.



СФА: Структурная формула алгоритма	
$A = A1 \rightarrow A2 \rightarrow A3 = (Z11 \rightarrow Z12 \rightarrow Z13 \rightarrow Z14) \rightarrow Z2 \rightarrow (Z11 \rightarrow Z12 \rightarrow Z13 \rightarrow Z14)$	(Явная) ИнФ: Инфиксная форма: относительно секвенции (\rightarrow).
$A = A1A2A3 = Z11Z12Z13Z14 \rightarrow Z2 \rightarrow Z11Z12Z13Z14$	НИнФ: Неявная инфиксная форма.

Рис. 2. Исходный последовательный алгоритм

ССА: Структурная схема алгоритма = ШСА: Штрих-схема алгоритма / ГИ: Горизонтальное исполнение	
1. Исходная детальная схема	
2. Условные упрощения графической записи исходной схемы	
СФА: Структурная формула алгоритма	
$A'' = A1' \rightarrow A2' \rightarrow A3' =$ $= (\rightarrow \& \downarrow^{1'}) \rightarrow Z11 \rightarrow Z12 \rightarrow Z13 \rightarrow (Z14 \uparrow^1 \# \rightarrow) \rightarrow (\rightarrow \& \downarrow^{2'}) \rightarrow Z2 \rightarrow (\uparrow^2 \# \rightarrow) \rightarrow (\rightarrow \& \downarrow^{3'}) \rightarrow Z11 \rightarrow Z12 \rightarrow Z13 \rightarrow (Z14 \uparrow^3 \# \rightarrow) =$ $= \downarrow^{1'} \& Z11 Z12 Z13 \# Z14 \uparrow^1 \rightarrow \downarrow^{2'} \& Z2 \# \uparrow^2 \rightarrow \downarrow^{3'} \& Z11 Z12 Z13 \# Z14 \uparrow^3$	

Рис. 3. Промежуточная базовая структура алгоритма (формирование конъюнктивных циклов с элементами параллелизма)

Промежуточная переходная структура

Первичная последовательная структура алгоритма А (рис. 2) заведомо неэффективна, так как связана с большими простоями пресса на ожидание полных циклов загрузки и разгрузки деталей с возвратом схватов роботов в исходные позиции.

Однако на ее основе формируется промежуточная переходная структура алгоритма А'' (рис. 3) как последовательная связь модификаций А1', А2', А3' составляющих алгоритмов:

1) Они представляю собой особые замкнутые циклы:

- циклы имеют узлы # разделения (дивергенции) потоков и узлы о = & соединения (конвергенции) потоков в общем контуре обратной связи циклов;
- узлы "о" соединения потоков представляют собой алгоритмическую конъюнкцию о = & и реализуют функцию разрешения (по вертикали) запроса (по горизонтали) на выполнение

очередной итерации цикла – только после завершения предыдущей итерации цикла;

- контуры с замыканием по конъюнкции имеют дополнительные входы для задания единичных (1) начальных условий (по дизъюнкции V).

2) Переходы Z14, Z34 возврата схватов Сх1, Сх2 роботов в исходные позиции включены в контуры обратных связей циклов А1', А3'. При этом:

- циклы А1', А2', А3' по-прежнему связаны и выполняются последовательно;
- но они имеют элементы параллелизма – переход Z14 возврата схвата Сх1 может выполняться одновременно с очередным циклом А2' штамповки, а переход Z34 возврата схвата Сх2 может выполняться одновременно с очередным циклом работы (внешнего) транспорта Тр2.

Структурная формула алгоритма (рис. 3, СФА) строится как модификация ЛСА и ПЛСА с

поперечными стрелками (согласно [6,7]) и достаточно очевидна по смыслу (в сопоставлении со схемами).

5.3. Синтез конвейерного алгоритма

Общий принцип синтеза

Приводится порядок синтеза конвейерного алгоритма методом полной (восходящей) индукции (рис. 4-7):

1) Строятся первые несколько частичных структурных решений – прохождение одной, двух, трех и т.д. итераций конвейерного цикла с проработкой блокирующих связей взаимного исключения. Обычно бывает достаточно первых 3-х итераций (рис.4 – 6).

2) Определяется общий вид структуры типовой итерации со всеми входными и выходными связями блокировок взаимного исключения (рис. 7/1).

3) Получается многоконтурное структурное решение конвейерного алгоритма на основе замыкания одноименных связей блокировок (рис7/2).

4) Выполняется проверка правильности полученного конвейерного структурного решения.

5) Проводится программное моделирование (рис. 8).

Одна первая итерация цикла

Строится (вырожденный) конвейерный процесс обработки партии деталей размером $n = 1$ (рис. 4):

$$A(n) = A(1) = A^1.$$

Процесс включает одну единственную первую итерацию A^1 , которая определяется исходя из необходимости правильного прохождения процесса продвижения и обработки только одной (первой) детали – в абстрагировании от всех прочих проблем.

При необходимости возможно замыкание нелинейных ("поперечных") выходов на общий линейный ("продольный") выход (пунктир, рис 4). Но для промежуточных решений это не требуется.

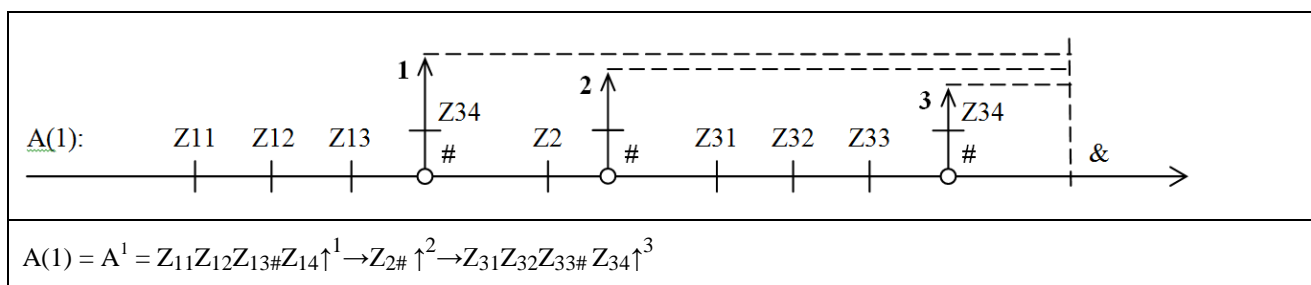


Рис. 4. Одна первая итерация общего цикла

Две первые итерации цикла

Строится конвейерный процесс обработки партии деталей размером $n = 2$ (рис. 5):

$$A(n) = A(2) = (A^1 \parallel A^2) = (A^1 \# \& A^2) \text{ .} \#(A^1, A^2) \&.$$

Процесс включает две первые итерации A^1, A^2 , которые определяются исходя из необходимости правильного прохождения процесса продвижения и обработки только 2-х деталей – также в абстрагировании от всех прочих проблем.

Используется высокоуровневый знак \parallel (параллель) параллельного объединения операторов по аналогии с работой [1]. Для нее принимается [5, 6] высокоуровневая интерпретация $\parallel = \#o = \# \&$ как парной операции, включающей операцию "#" разделения (дивергенции) потоков и операцию "o" соединения (конвергенции) потоков по алгоритмической конъюнкции $o = \&$.

Запись $(A^1 \# \& A^2) \text{ .} \#(A^1, A^2) \&$ означает, что традиционной инфиксной форме (ИнФ) записи операций (между операндами) соответствует префиксно-постфиксная форма (ПрПрФ) записи операций $\#(A^1, A^2) \&$, которая прямо соответствует

общему виду (несущего остова) структурной схемы с двумя параллельными потоками (рис. 5).

Принимается связь данной базовой параллельной структуры с ПЛСА [4] согласно [5, 6]:

$$\#(A1, A2) \& = R \uparrow^1 \uparrow^2 \downarrow^1 A1 \uparrow^3 \downarrow^2 A2 \uparrow^4 \downarrow^3 \downarrow^4 S,$$

где $R \uparrow^1 \uparrow^2$ – оператор распараллеливания ($R \sim \#$);

$\downarrow^3 \downarrow^4 S$ – оператор соединения ($S \sim o = \&$).

ЛСА и ПЛСА были разработаны для задач низкоуровневого программирования (системного программирования и микропрограммных автоматов) и не получили распространения в высокоуровневом прикладном программировании. Но в излагаемой системе анализа алгоритмов им придается фундаментальное базовое значение с последующей высокоуровневой интерпретацией базовых алгоритмических структур и для совместного их применения с дополнительными нелинейными связями потоков управления (рис 5 – 7).

Связи блокировок двух потоков

Выделяются следующие блокировочные нелинейные связи двух параллельных потоков (рис. 5):

- связи 11, 21, 31 – блокируют повторный запуск переходов загрузки, обработки и разгрузки деталей во втором потоке до завершения соответствующих переходов в первом потоке;
- связь 51 блокирует начало нового перехода Z13 укладки детали в рабочую позицию схватом Сх1 во втором потоке до завершения перехода Z2 обработки детали в первом потоке;
- связь 71 блокирует запуск нового перехода Z2 обработки детали во втором потоке, до окончания перехода Z31 выпорки детали из рабочей позиции схватом Сх2 в первом потоке;

- связь 61 блокирует одновременный доступ в рабочую двух схватов Сх1 и Сх2 ().
- аналогично отражается наличие связей блокировок для внешнего транспорта Тр1, Тр2 (для упрощения схемы эти блокировки обозначены в принципе – отрезками связей).

Строго говоря, некоторые блокировочные связи в их общей массе могут быть избыточными. Но изначально они все формально учитываются.

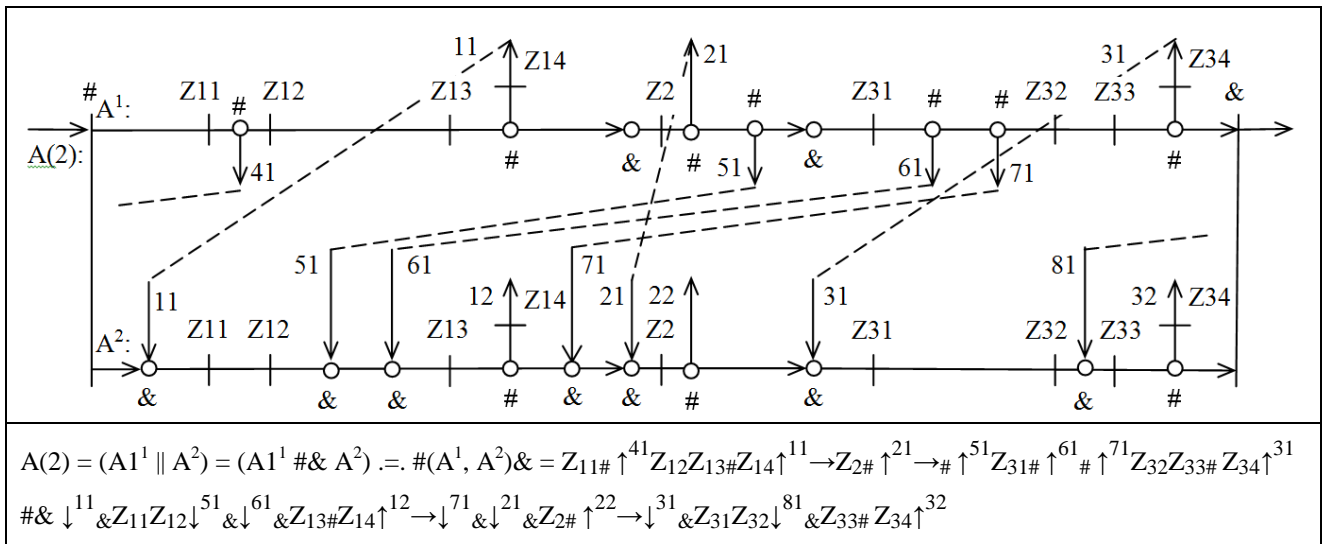


Рис. 5. Первые 2 итерации цикла

Три первые итерации цикла

Аналогично строится конвейерный процесс обработки партии деталей размером $n = 3$:

$$A(n) = A(3) = (A^1 \parallel A^2 \parallel A^3) = (A^1 \# \& A^2 \# \& A^3) \therefore \#(A^1, A^2, A^3) \&$$

Затем полученная схема дорабатывается (рис. 6) на следующем этапе синтеза.

Унификация итераций конвейерного цикла

Вторая (средняя) итерация (рис. 6, рис. 7/1) является типовой основой для общего структурного решения в построении конвейерного алгоритма управления модулем. Однако она должны быть типовой для всех итераций, включая упрощенные итерации первого и последнего прохода детали в обработке (конечной) партии деталей.

Для этого по аналогии со средней (2-й) итерацией (рис. 6, рис. 7/1):

- в последнюю (3-ю) итерацию добавляются недостающие выходные связи;
- в первую итерацию добавляются недостающие входные связи.

При этом в первой итерации на дополнительных конъюнктивных входах вводятся единичные начальные условия (по аналогии с рис. 2).

Построение конвейерного алгоритма

За основу принимается типовая средняя итерация:

- удаляются вспомогательные вторые индексы блокировочных связей;
- соединяются одноименные концы выходных и входных связей блокирующих контуров связей;
- уточняются единичные начальные условия входных связей (темные кружки).

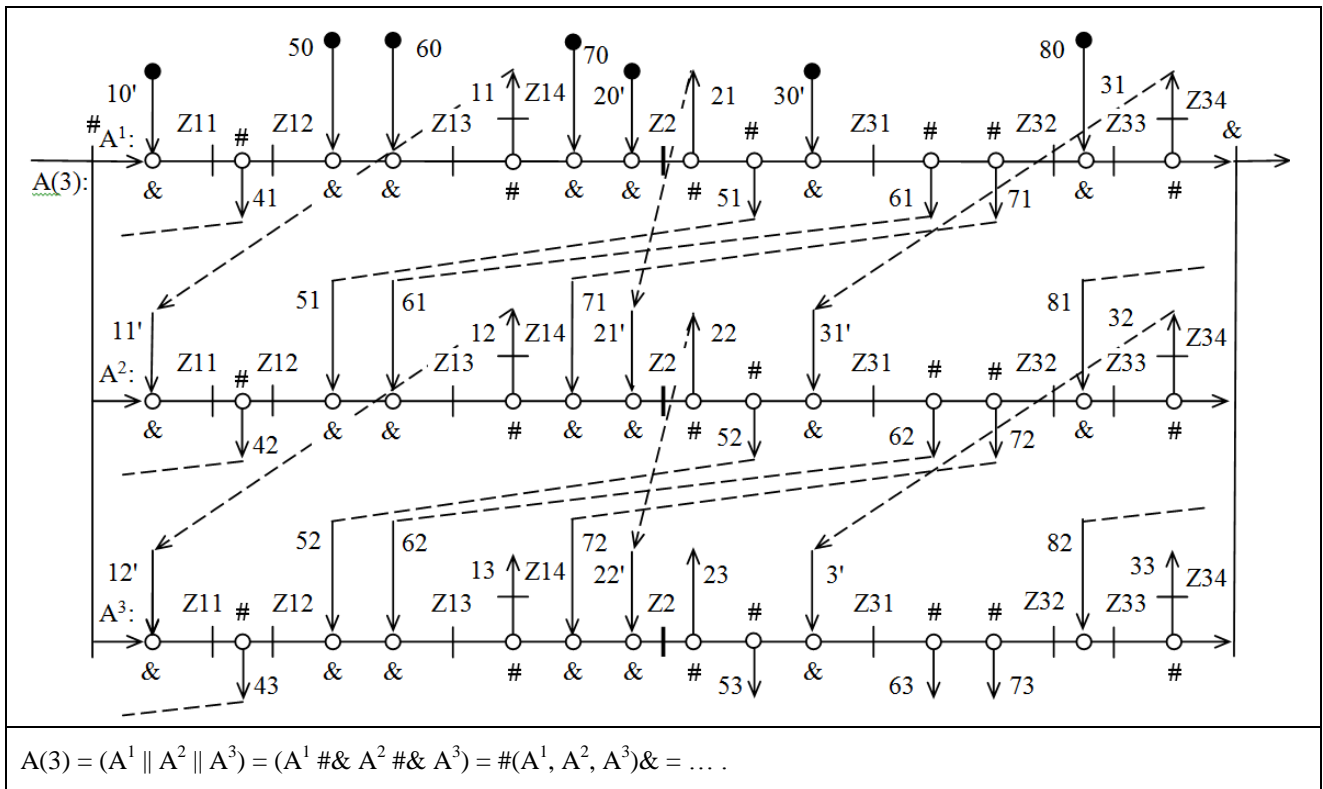


Рис. 6. Первые 3 итерации цикла. Эквивалентная унификация структуры всех итераций цикла

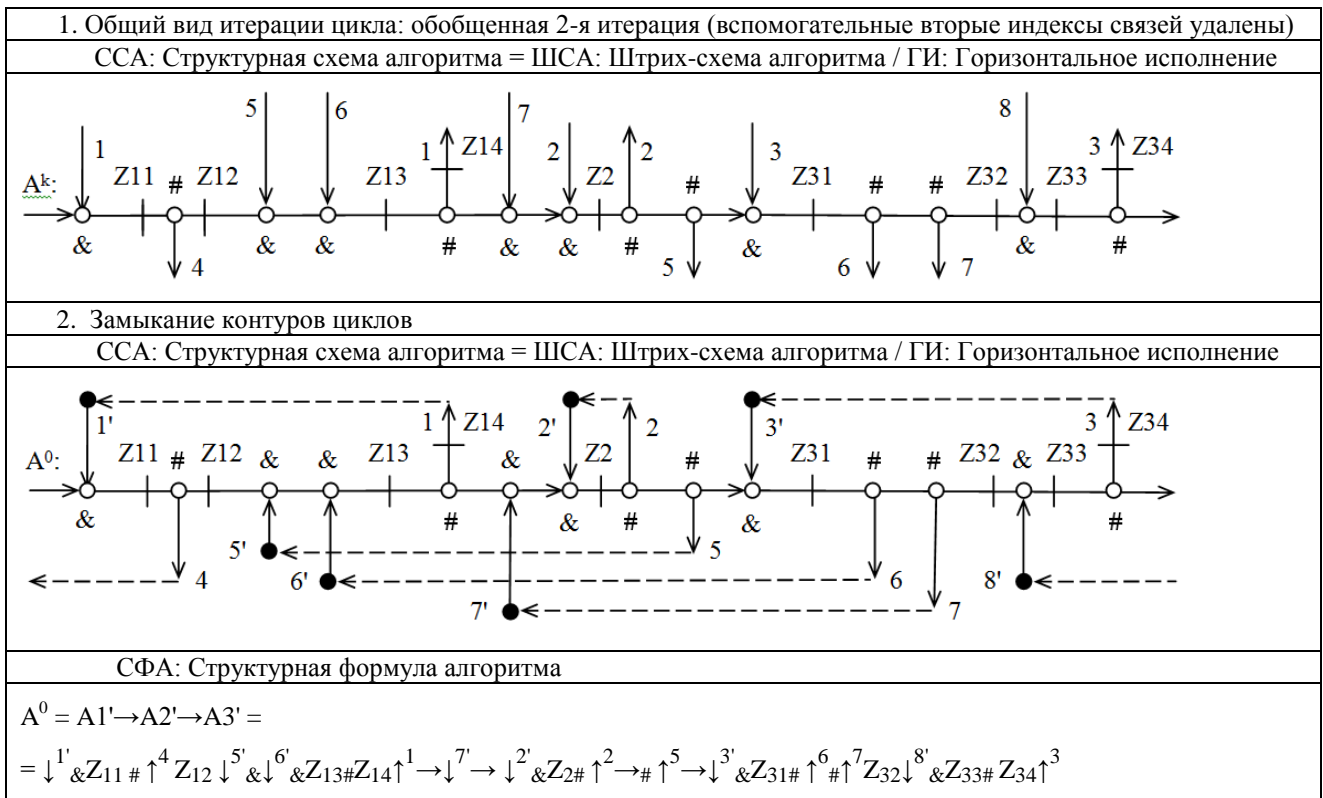


Рис. 7. Свертка циклической конвейерной структуры

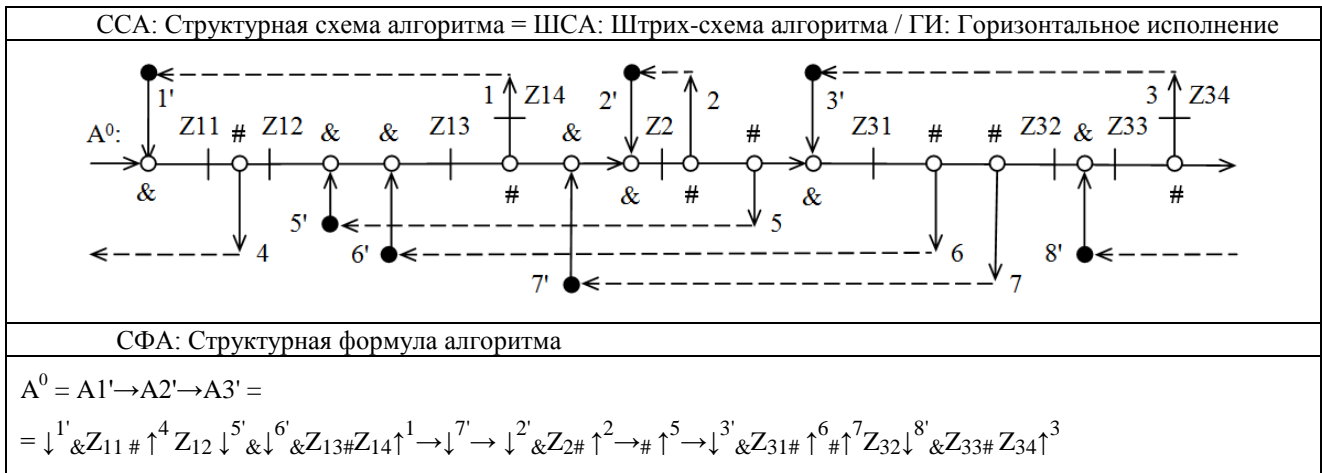


Рис. 7. Свертка циклической конвейерной структур

На последней схеме основные циклы переходов загрузки, обработки и разгрузки деталей развернуты вверх (по аналогии с промежуточной схемой рис. 2), а дополнительные контуры развернуты вниз.

Получается **система зацепленных конъюнктивных циклов**, обеспечивающих заданное взаимное исключение параллельных процессов итераций.

Обоснование полученного решения

Полученная структура является телом цикла конвейерной обработки партии деталей (который оформляется как цикл повторения по числу деталей партии, цикл с предусловием наличия деталей и т.п.).

Возможно построение доказательства правильности полученного алгоритмического решения методом полной математической индукции, включая пункты:

- доказать утверждение: из правильности схемы для некоторого этапа i построений (схема на i итераций потоков) логически следует правильность схемы очередного этапа ($i+1$);

- доказать правильность решения для некоторого начального этапа i_0 (например для $i_0 = 1$);
- получить вывод типа: данная схема решения справедлива для любых значений $i = n \geq i_0$.

Аналогичным образом можно доказывать разные другие свойства данного алгоритмического решения, включая дополнительную специфику программной реализации конвейерного алгоритма.

Программное моделирование конвейерного алгоритма

В конечном счете, полученные теоретические структурные решения необходимо подтвердить практически программным моделированием.

Были разработаны две действующие учебно-исследовательские графические программные модели роботизированного комплекса данного типа, реализованные двумя разными методами (рис. 8):

- многопоточное программирование (язык Python);

- событийно-ориентированное программирование (учебный конструктор игр и презентаций Scratch).

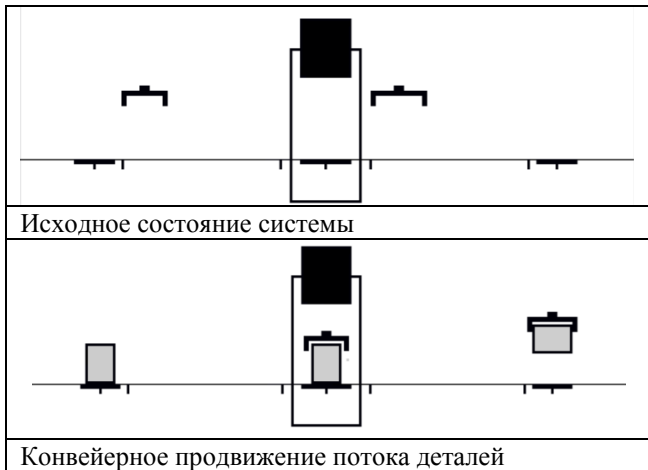


Рис. 8. Визуальная программная модель

В обоих случаях используются программные модели идеализированной базовой модели аппаратной реализации конвейерного алгоритма. При этом изначально точно воспроизводится явная структурная логика параллелизма конвейерного алгоритма.

На этой точной теоретической основе возможно применение и исследование специальных средств многопоточного программирования (низкоуровневых семафоров, высокоуровневых мониторов с автоматической расстановкой системы семафоров и т.д.), событийного программирования и т.п.

6. Заключение

Кратко излагается синтез асинхронных конвейерных алгоритмов для робототехнологических модулей методом полной (восходящей) индукции, включая:

- синтез первых частных решений для цикла конвейерной обработки партии деталей нарастающей численности и сложности, включая одну, две и три первые итерации цикла;

К данной копии статьи добавлены следующие приложения.

Приложение 1.

Модельная программа РТК: Роботизированный технологический комплекс

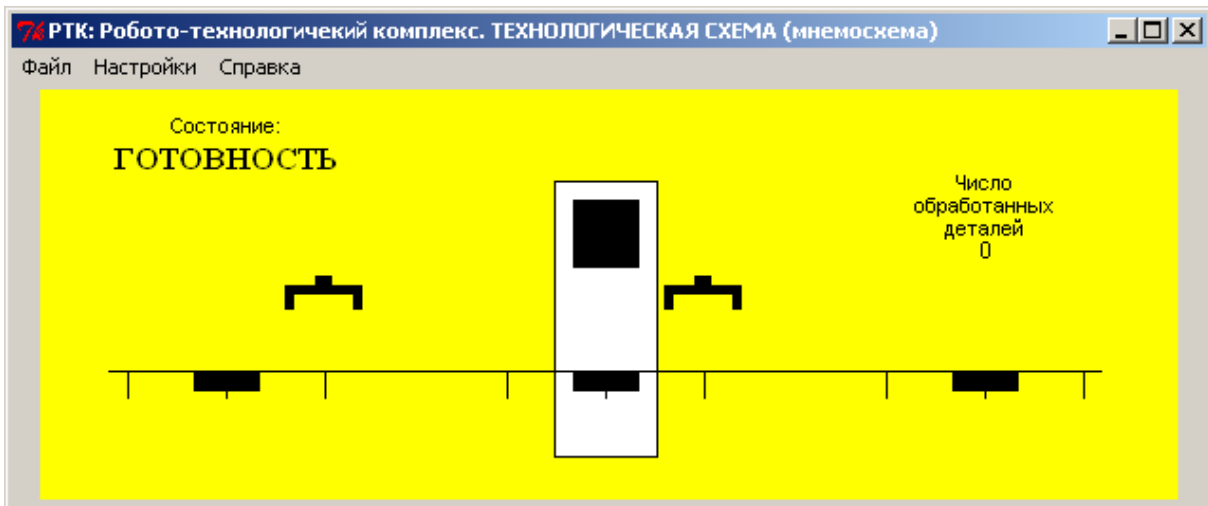
- определение типовой итерации цикла и формирование его конвейерной структуры;
- теоретическое обоснование структурного решения и его программное моделирование.

Список используемых источников

Статьи [2, 3, 5, 6] автора доступны на сайте:

URL: <http://paralg.ucoz.com/>

1. Хоар Ч. Взаимодействующие последовательные процессы. – М.: Мир, 1989. – 264 с.
2. Житников А. П. [Алгоритмизация управления операционными циклами роботизированных технологических комплексов](#). – Уфа, УАИ, 1983. – Деп. НИИМаш, № 236 мш–Д83. – 22 с.
3. Житников А.П. [Алгоритмизация управления роботизированными комплексами](#). / Материалы III Всесоюзной конференции "Роботы и робототехнические системы". – Челябинск: ЧПИ, 1983. – С. 59 – 60.
4. Лазарев В. Г., Пийль Е. И. Синтез управляющих автоматов. – М.: Энергоатомиздат, 1989. – 328 с.
5. Житников А. П., Житников В. П., Шерыхалина Н. М. [Программно-методический комплекс "Параллельные алгоритмы и программы"](#). / Мат. науч.-тех. совещания "Высокопроизводительные вычислительные ресурсы России: состояние и перспективы развития". – Уфа: УГАТУ, 2003. – С. 151 – 161.
6. Житников А. П. [Базовая структурная семантика параллельных алгоритмов](#). / Принятие решений в условиях неопределенности: Межвуз. научн. сб. – Уфа: УГАТУ, 2008. – С. 41-50.



ДМС: Динамическая мнемосхема системы. Исходное состояние системы

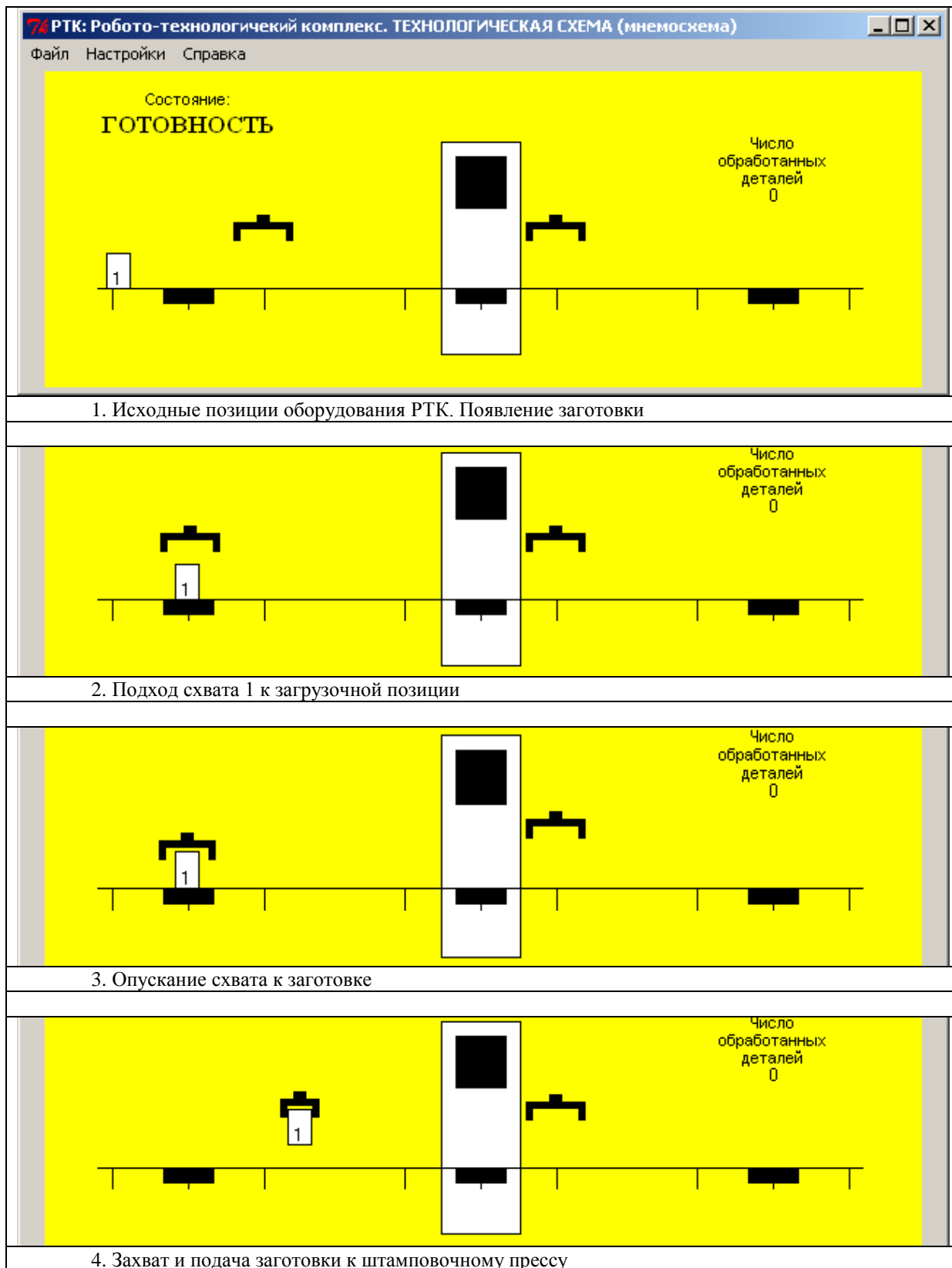


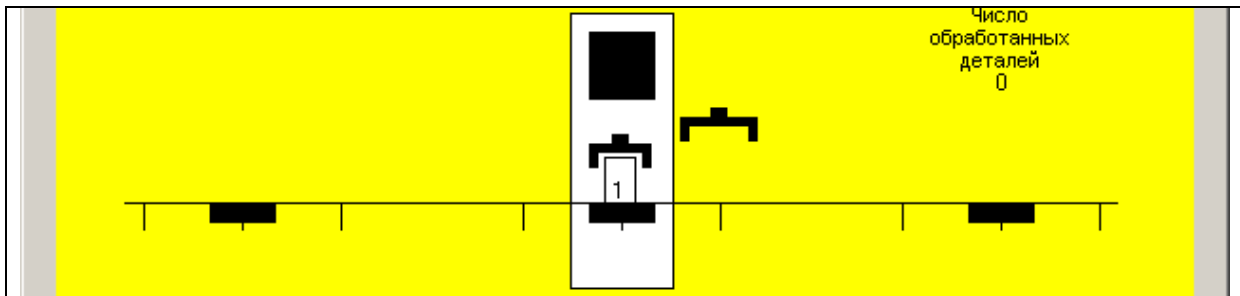
ПУ: Пульт управления (модель)

Приложение 2.

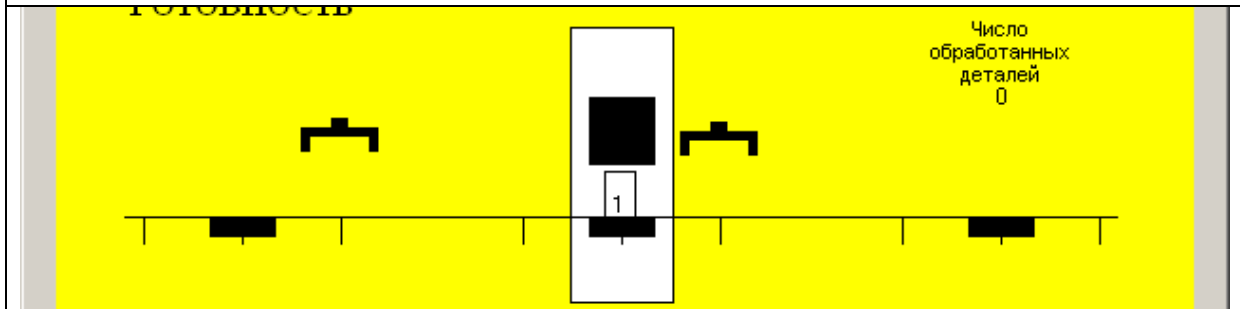
Режим одиночных циклов (не конвейерный режим)

Цикл продвижения и обработки одной детали

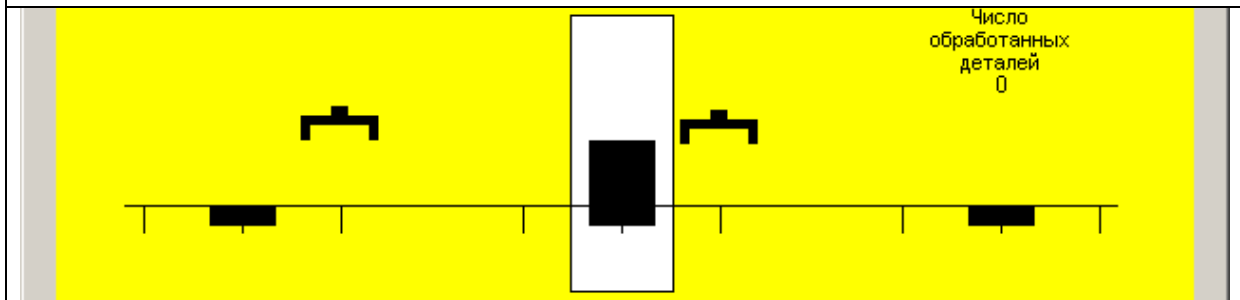




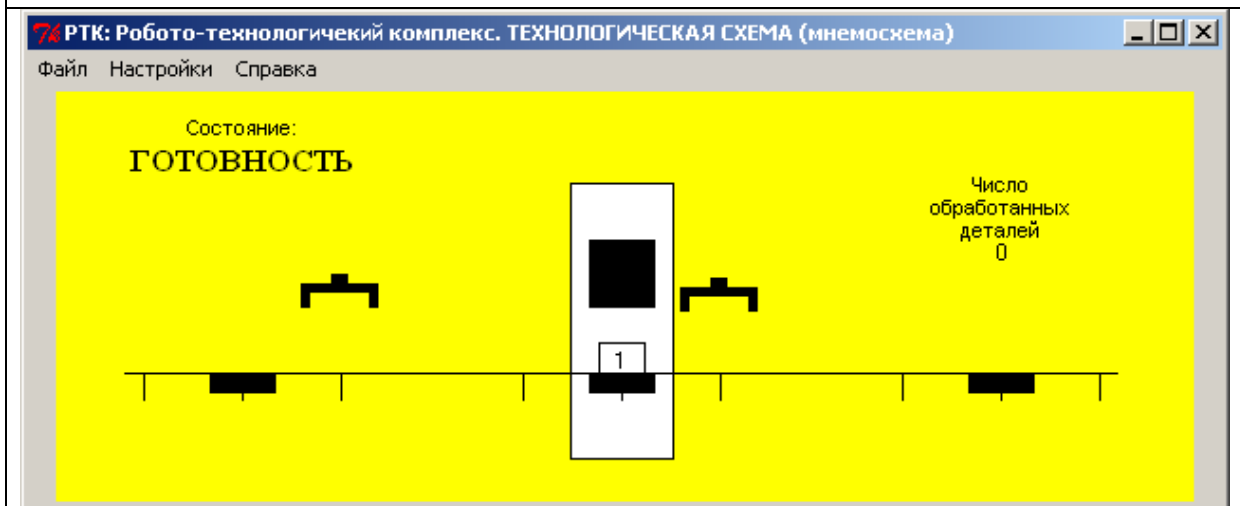
5. Установка заготовки в рабочую позицию – в матрицу штампа



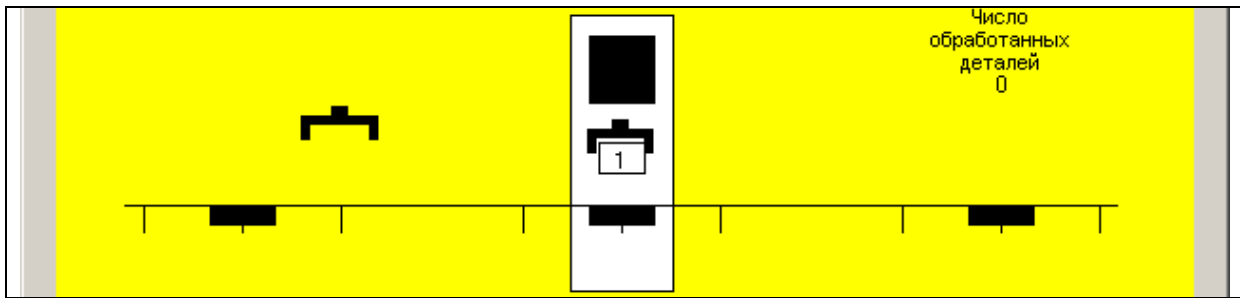
6. Отвод схвата 1 в исходную позицию



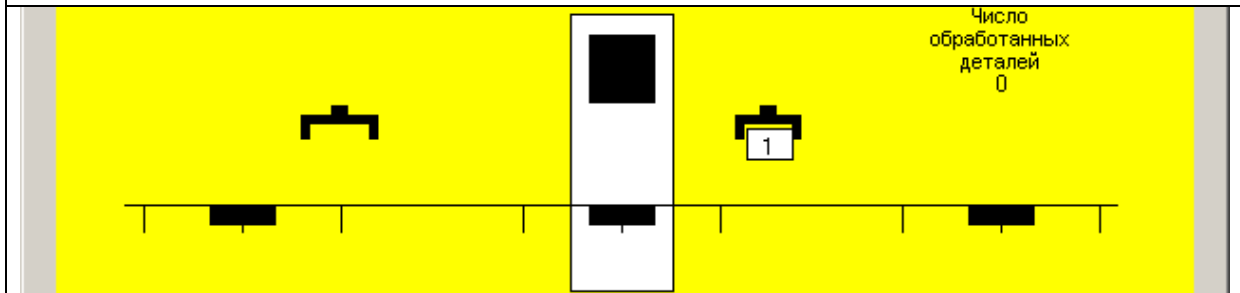
7. Опускание ползуна штамповочного пресса с пуансоном штампа



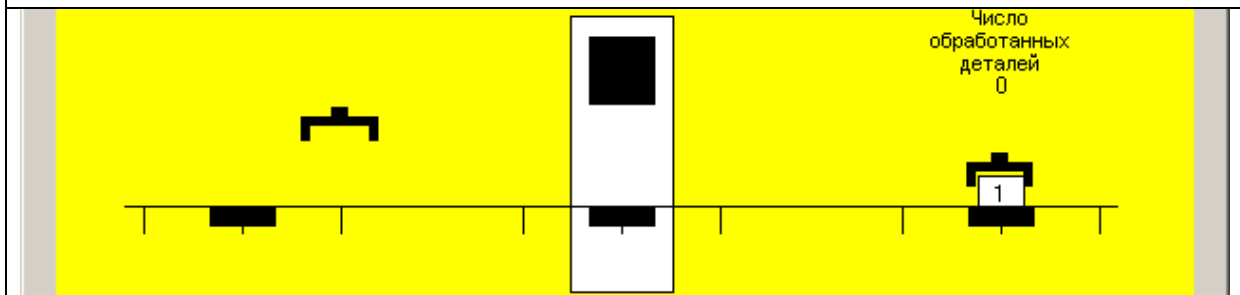
8. Отвод вверх ползуна штамповочного пресса



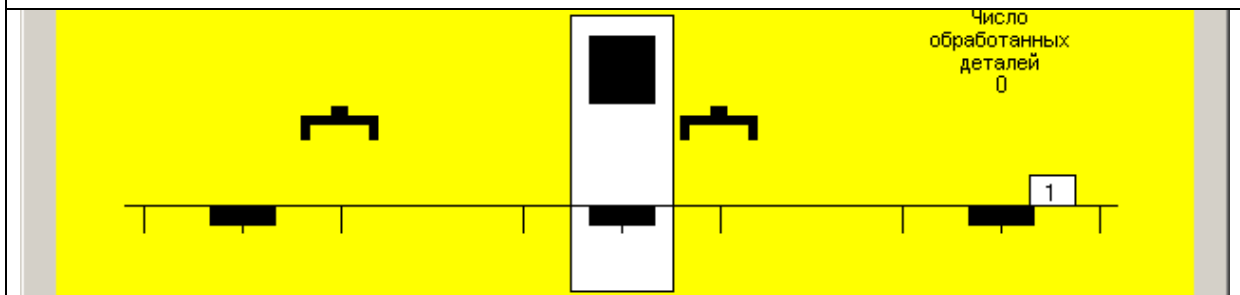
9. Захват штампованной детали схватом 2



10 Подача детали от штамповочного пресса



11. Установка штампованной детали в разгрузочную позицию



12. Возврат схвата 2 в исходную позицию. Удаление детали

Приложение 3.
Конвейерный режим: обработка потока деталей

