Министерство общего и профессионального образования
Российской Федерации
Самарский государственный аэрокосмический университет
имени академика С. П. Королева

А. А. Калентьев

Автоматизированный синтез алгоритмов асинхронного управления техническими системами с множеством дискретных состояний

Самара 1998

УДК 681.3.068:519.7

Калентьев А.А. Автоматизированный синтез алгоритмов асинхронного управления техническими системами с множеством дискретных состояний / Самар. гос. аэрокосм. ун - т. Самара, 1998. 204с.

В монографии рассмотрена методология проектирования алгоритмов асинхронного управления сложными системами. В основу изложения положен принцип создания алгоритма управления под решаемую системой целевую задачу. Изложены все этапы проектирования, начиная от постановки целевой задачи и ее формального описания и кончая текстом алгоритма управления, полученного средствами автоматизированной системы «ГРАФКОНТ».

Книга предназначена для инженерно - технических и научных работников, специализирующихся в области автоматизации технических систем и АСУ реального времени.

Табл. 37. Ил. 19. Библиогр.: 32 назв.

Рецензенты: д - р техн. наук, проф. М.А. Кораблин
д - р техн. наук Ю.Н. Лазарев

ISBN 5-7883-0053-3

ОГЛАВЛЕНИЕ

Предисловие

Важной составляющей при решении проблемы комплексной автоматизации является создание методов и средств управления техническими системами с множеством дискретных состояний, которые занимают значительное место в машиностроении, а также широко используются в химической промышленности, энергетике, металлургии. Такие системы, характерные для многих отраслей промышленности, обладают значительной «сложностью», состоят из большого числа элементов, функционирующих зависимо либо независимо между собой.

Проблема организации управления такими системами заключается в описании их поведения (функционирования) на некотором формализованном языке и последующей реализации, на базе этого описания, алгоритма управления. Для сложных систем решение этой проблемы традиционными методами, в рамках теории дискретных систем и конечных автоматов, оказывается малоэффективным. Это связано с необходимостью иметь формализованное описание алгоритма управления. Отсутствие формальных методов проведения начального этапа проектирования, на котором определяется переход от физической постановки задачи к математической, существенно удлиняет сроки создания и ввода в действие системы. Основной причиной такого положения является недостаточная проработанность вопросов, связанных с поведением системы, взаимодействием ее элементов в процессе функционирования. Это относится, в частности, к изучению систем, содержащих разнотипные элементы, к разработке и исследованию моделей поведения систем, в рамках которых можно описать их взаимодействие, а также к поиску эффективных методов и алгоритмов управления такими системами. Иными словами, традиционные методы управления позволяют построить соответствующий алгоритм по его формализованному описанию безотносительно к поведению системы. Требуется систематизация и формализация знаний о поведении системы. Это позволит перейти к алгоритму, управляющему заданным поведением.

Предлагаемая вниманию читателей книга является одной из попыток решения этой проблемы. В ней рассмотрены модель описания поведения системы с множеством дискретных состояний и методы построения соответствующих алгоритмов управления с использованием разработанного математического аппарата функционального исчисления (алгебра взаимодействия элементов (приборов) системы). Традиционный математический аппарат конечных автоматов и сетей Петри, введенный в шестидесятых годах и нашедший развитие в работах ряда ученых (А.Д. Закревский, Д. Петерсон, В.Е. Котов и др.) требует формализованного описания алгоритма управления. Достоинством предлагаемого аппарата является наглядность и простота формализованного описания поведения системы. В некотором смысле это язык арифметических выражений, каждое из которых описывает взаимодействие двух, трех и т.д. элементов. Такое описание дает возможность проведения всех дальнейших исследований формальными методами, в частности, построение алгоритма асинхронного управления. Применение функционального исчисления для формализованного описания поведения системы и алгоритмов ее управления, по мнению автора, будет представлять интерес для широкого круга специалистов в области автоматизации средств управления.

В книге изложена методология автоматизированного проектирования алгоритмов асинхронного управления техническими системами с множеством дискретных состояний. Изложение ведется последовательно, начиная от построения модели, описывающей поведение системы. Затем строится функциональное исчисление, являющееся теоретической основой его формализованного описания. В следующей главе строится функция управления системой, которая, в свою очередь, является формализованной основой асинхронного алгоритма управления. Заканчивается работа созданием текста алгоритма на некотором внутреннем языке автоматизированной системы.

Благодарности - профессору, ректору Самарского государственного аэрокосмического университета В.А. Сойферу за поддержку и доверие; профессору, начальнику отделения управления ГНПРКЦ «ЦСКБ - Прогресс» Ю.Г. Антонову за постановку задачи и приглашение к ее решению; профессору, начальнику отдела ГНПРКЦ «ЦСКБ - Прогресс» Я.А. Мостовому за сотрудничество и критику; ассистенту кафедры технической кибернетики А.А. Тюгашеву за верность идее, а также сотрудникам кафедры технической кибернетики СГАУ и инженерам 1402 отдела ГНПРКЦ «ЦСКБ - Прогресс» за участие в вычислительных экспериментах.

1. СИСТЕМЫ С МНОЖЕСТВОМ ДИСКРЕТНЫХ СОСТОЯНИЙ

1.1 Основные понятия

Рассматривается автономная техническая система, состоящая из n разнотипных элементов и процессора, выполняющего функции управления, чтения, записи и преобразования данных от элементов системы. Элементы связаны с процессором по управляющим и информационным каналам. Между собой они связаны по каналам передачи данных и под управлением процессора могут осуществлять эти передачи.

По информационным каналам процессор может передавать данные элементам системы либо принимать их после завершения своей работы, принимать признак успешного и/или неуспешного завершения работы элемента системы. По управляющим каналам процессор может передавать элементам команды управления: включать и/или отключать элемент и т.д. Будем считать, что длительность работы элементов системы значительно превышает длительность выполнения команд управления, команд приема и передачи данных, команд проверки истинности логических условий. Элемент может находиться в состоянии «включен» либо «отключен». Переход из одного состояния в другое определяется значением логического условия, которое, в свою очередь, может определяться значениями данных, поступающих на вход элемента. Элементы относительно друг друга могут включаться в неравномерные или асинхронные моменты. Состоянием системы назовем состояние ее элементов.

Системы, функционирование которых есть процесс смены их состояний, назовем системами с множеством дискретных состояний. Моменты смены состояний будут называться асинхронной последовательностью, а механизм, отвечающий за согласованную работу элементов, назовём алгоритмом асинхронного управления.

Функционирование системы определяется взаимодействием ее элементов и направлено на выполнение некоторой заданной целевой задачи. Для ее решения требуется согласование работы элементов по данным, поступающим с выхода одного элемента на вход другого, по времени. Учитывая, что система содержит большое число элементов, задача согласования носит довольно трудоемкий характер.

Проблема организации управления такими системами заключается в описании их поведения (функционирования) на некотором формализованном языке и последующей реализации, на базе этого описания, алгоритма управления.

Ее решение сводится к построению закона функционирования системы, поиску математических средств описания целевой задачи и на ее основе алгоритма асинхронного управления. Все это и составляет комплексную проблему согласованной работы приборов и устройств автономной технической системы.

Ставится задача разработки теоретических основ данной проблемы и на ее основе создания информационной технологии формирования алгоритмов асинхронного управления (ААУ).

1.2 Схема функционирования системы

Обозначим $\Phi = \{\Phi_1, \Phi_2, \ldots \Phi_n\}$ - множество элементов системы. Зададим множество переменных, с которыми работают элементы системы $I= \{ x, y, z, \ldots\}$. Для каждого элемента системы Φ_i определены упорядоченные множества $in(\Phi_i)\subseteq I$, $out(\Phi_i)\subseteq I$ соответственно входных и выходных переменных. Пару $< \Phi, I >$ назовем функциональным базисом системы.

На множестве элементов Φ можно ввести отношение связности: $r(\Phi_i, \Phi_j) = (in(\Phi_i) \& out(\Phi_j)) \cup (in(\Phi_j) \& out(\Phi_i)) \cup (out(\Phi_i) \& out(\Phi_j))$, обладающее свойством $r(\Phi_i, \Phi_j) \neq \varnothing$. Оно означает, что элементы Φ_i, Φ_j могут иметь общие данные, следовательно, могут влиять друг на друга. Результат функционирования системы будет зависеть от порядка включения этих элементов.

Каждому элементу Φ_i системы поставим в соответствие символ инициализации или включения $\Phi_i^в$ и непустое множество символов завершения работы

$$Z(\Phi_i) = \{ \Phi_i^1, \Phi_i^2, \ldots, \Phi_i^k \}$$

Здесь символ Φ_i^k есть завершение работы элементом Φ_i с выбором k-й альтернативы.

Схемой функционирования P системы над функциональным базисом $< \Phi, I >$ называется конечная последовательность

$$P= (P_0, P_1, \ldots, P_n),$$

где: P_0- включение начального (нулевого) элемента системы $\Phi_0^в$;

P_i ($i=1, 2, \ldots, n-1$) есть либо символ включения $\Phi_j^в$ элемента Φ_j в работу, либо символ завершения его работы Φ_j^2 с выбором, например, 2-й альтернативы. Символ включения на P_i-м шаге может быть условным либо безусловным. Безусловное включение означает выполнение пункта i схемы функционирования Р. Условное включение означает выполнение пункта i схемы Р при истинности условия и невыполнение пункта i схемы Р при его ложности. Происходит переход к следующему i+1-му пункту.

P_n - завершение работы начального элемента системы $Ф_0{}^1$.

Включение нулевого элемента $Ф_0$ есть включение системы на холостом ходу: подача питания, включение процессора, инициализация констант и т.д.. Завершение работы нулевого элемента $Ф_0$ есть завершение работы системы: отключение процессора и питания.

Исходя из физического смысла задачи, для каждой последовательности Р можно установить следующие аксиомы.

1.Для любого элемента $Ф_i$ в схеме Р одинаковым должно быть число включений и завершений его работы. Эта аксиома выражает условие нормальной работы системы.

2.Если схема Р содержит завершение элемента $Ф_i{}^1$ на l-м месте, то она обязательно содержит включение элемента $Ф_i{}^a$ на j-м месте и j<l. Эта аксиома выражает требование, чтобы отключение элемента на l-м шаге происходило не раньше его включения.

3.Перед повторным включением элемента $Ф_i$ обязательно должно быть его отключение, т.е. запрещается повторное включение при включенном элементе.

4.Выполнение отношения связности.

Выполнение всех аксиом означает выбор правильной последовательности.

Схему функционирования удобно представлять в виде временных циклограмм. Пусть, например, I= { x,y,z },

Φ={ Φ_1, Φ_2, Φ_3 },

$in(\Phi_1)=\{x\}$, $in(\Phi_2)=\varnothing$, $in(\Phi_3)=\{x,y\}$;

$out(\Phi_1)=\{y\}$, $out(\Phi_2)=\{x\}$, $out(\Phi_3)=\{z\}$,

$Z(\Phi_1) = \{ \Phi^1{}_1, \Phi^2{}_1 \}$,

$Z(\Phi_2) = \{ \Phi^1{}_2 \}$,

$Z(\Phi_3) = \{ \Phi^1{}_3, \Phi^2{}_3 \}$.

Тогда схемами являются:

$\Phi_0{}^в, \Phi_2{}^в, \Phi_2{}^1, \Phi_1{}^в, \Phi_1{}^1, \Phi_3{}^в, \Phi_3{}^2, \Phi_0{}^1,$

$\Phi_0{}^в, \Phi_1{}^в, \Phi_2{}^в, \Phi_2{}^1, \Phi_3{}^в, \Phi_3{}^2, \Phi_1{}^2, \Phi_0{}^1.$

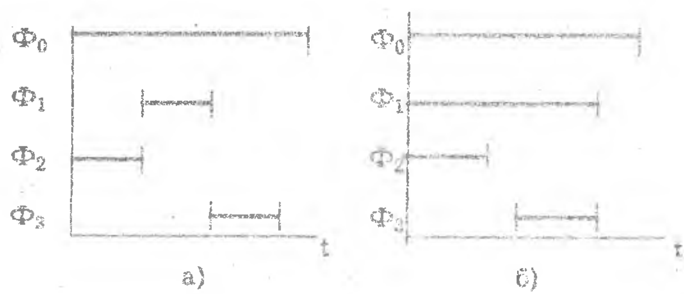

Рис.1.1.Схемы функционирования

Как видно из рис.1.1,а и б, первая и вторая схемы функционируют по разному. Отметим, что схема не отражает протяженность выполнения элементов во времени. Она не является законом функционирования системы. С ее помощью можно исследовать структурные свойства последовательностей: упорядоченность элементов в последовательности, отношение связности и т.д. Нас же интересует закон функционирования системы или смысловое содержание всех действий.

Последовательность P можно рассматривать как протокол поведения системы. Впервые такой протокол был введен [26] для описания модели функционирования алгоритма в теории схем программ. Схема программы - это математическая модель алгоритма, отвечающая основным его свойствам: модель позволяет изучать свойства достаточно широкого класса алгоритмов, а не их отдельных конкретизаций; модель сохраняет все интересующие исследователя свойства и особенности рассматриваемого класса алгоритмов; модель позволяет игнорировать несущественные для изучаемой проблемы свойства, например, синтаксические детали; модель модифицируема и допускает нововведения, отслеживающие прогресс в теории программирования.

Последовательность Р - это еще не закон функционирования конкретной системы, а только его схема. Она отражает структуру закона поведения системы, строение множества допустимых переходов системы, получающихся из схемы в результате задания различных интерпретаций. Она становится законом только после однозначной интерпретации входящих в нее элементов.

В работах [3,15,23] такой подход получил дальнейшее развитие в задачах распараллеливания алгоритмов и программ. Под схемой здесь понимается вычислительный процесс над информационным базисом. Элементами этого процесса являются действия по включению либо отключению операторов информационного базиса. Также вводятся интерпретации базиса и рассматривается задача преобразования последовательной программы в параллельную, эквивалентную последовательной.

1.3 Интерпретация функционального базиса

Функционирование всей системы можно определить через функционирование ее элементов, которые, как правило, разной физической природы. Каждый из них выполняет преобразования некоторого типа: энергетического, механического, электрического и т.д. Для определения поведения всей системы абстрагируемся от разнотипности поведения ее элементов и приведем их к единому базису. В качестве базиса удобно предложить параметры, которыми описывается поведение системы. Будем интерпретировать работу каждого элемента и всю систему во времени, в логическом и информационном пространствах. Такая интерпретация позволяет иметь единый способ описания как одного элемента, так и их совокупности и, как следствие, описание всей системы. Действительно, каждый элемент и вся система работают во времени, поэтому наличие временной интерпретации очевидно.

Все измеряемые количественные преобразования носят числовой характер, выражением которых является информационное пространство. Наличие качественных преобразований, если они фиксируются некоторыми приборами, можно интерпретировать в информационном пространстве в двоичном либо целочисленном виде.

Интерпретация в логическом пространстве определяет ограничения на функционирование как отдельного элемента, так и их совокупности. Заданием логических условий можно описать согласованную работу двух и более элементов; можно согласовать работу двух и более элементов на временной оси, в информационном пространстве, на множестве имен элементов.

Указанные интерпретации, являясь свойством отдельных элементов, могут быть перенесены и на всю систему в целом. Вся система работает на некотором временном интервале, длина которого определяется через длины интервалов отдельных элементов системы. В информационном пространстве данные, формируемые отдельными элементами в объединении дают данные всей системы. В логическом пространстве всегда можно ввести условие на включение всей системы понимая его как условие, которое накладывается на каждый элемент функционального базиса.

Интерпретацией базиса [7] и области D называется функция Int(x), которая каждому элементу базиса x ставит в соответствие элемент d из области интерпретации D: d=Int(x).

Элемент системы Φ_i функционального базиса Ф будем интерпретировать во времени, в логическом и информационном пространствах, т.е.

$Int(\Phi_i) = < Int(\Phi_i)(t), Int(\Phi_i)(\alpha), Int(\Phi_i)(I) >$.

Интерпретация на временной оси:

$Int(\Phi_i)(t) := < t_{вкл}, \tau >$,

где $t_{вкл}$ - время включения элемента Φ_i;

τ - длительность работы элемента Φ_i.

Интерпретация в информационном пространстве:

$Int(\Phi_i)(I) := < F(\Phi_i), G(\Phi_i) >$,

где $F(\Phi_i)$: $D^n \rightarrow D^m$ - отображение, превращающее элемент системы в преобразователь данных;

D- область интерпретации переменных информационного пространства;

n(m)- число входных (выходных) данных элемента Φ_i.

$G(\Phi_i)$: $D^n \rightarrow Z(\Phi_i)$- отображение, в зависимости от значений входных данных выбирающее конкретную альтернативу завершения работы элемента Φ_i.

Интерпретация в логическом пространстве:

$Int(\Phi_i)(\alpha) := < L(\Phi_i) >$;

где $L(Ф_i)$: $U \rightarrow$ {TRUE, FALSE}, U={D} ∪ { t } ∪ { Ф }- объединение непересекающихся множеств.

$L(Ф_i)$ - отображение, выражающее условие включения в работу элемента системы $Ф_i$.

Интерпретация функционального базиса задается множеством интерпретаций его элементов.

Пару < P, Int >, где Int понимается как интерпретация каждого элемента функционального базиса по времени (t), в логическом (α) и информационном (I) пространствах, назовем интерпретированной схемой над базисом.

Будем фиксировать работу элемента системы в моменты времени $t_0 = t_{вкл}$ и $t_1 = t_{вкл} + \tau$. Условием включения элемента в работу является истинность логического выражения $L(Ф_i)$. При его истинности перед включением на вход элемента подаются исходные данные $Inp(Ф_i)$, если они определены к этому моменту времени. В момент t_1 работа элемента заканчивается либо автоматически, либо по команде отключения элемента. Элемент может завершить свою работу успешно либо неуспешно. Введением логической переменной можно распознать эти ситуации. После успешного завершения работы на выходе элемента формируются новые данные $Out(Ф_i)$. Так как эти данные могут быть использованы другими элементами, их необходимо переписать в некоторое поле памяти для сохранения. Значения выходных данных зависят от комбинации значений данных на входе элемента, по разным комбинациям значений входных данных получаются разные по значению выходные данные. Это означает, что у элемента может быть несколько выходных состояний.

Нетрудно заметить, что приведенные в предыдущем пункте аксиомы выполняются и для интерпретированной схемы < P, Int >.

Таким образом, состояние элемента системы может изменяться во времени, в информационном и логическом пространствах. Договоримся фиксировать эти изменения в моменты включения и/или отключения этих элементов.

1.4 Стандартная схема базиса и ее реализации

Схема функционирования P над базисом < Ф, I > определяет качественно функционирование системы. Смысловое содержание вносится правилами интерпретации базиса. В соответствии с этими правилами и с учётом допущения, что изменения состояния элемента происходят в моменты его включения и/или отключения, опишем семантику функционирования системы.

Определим состояние u_i системы в i-й момент времени t_i совокупностью вида

$$(t_i, u_i) = \{(\Phi_j, L(\Phi_j), I_j)\}, j=1,...,k.$$

Совокупность $(\Phi_j, L(\Phi_j), I_j)_i$ - подмножество элементов Φ_j, динамически включаемых и/или отключаемых в i-й момент времени, в которой Φ_i - имя элемента функционального базиса;

$L(\Phi_j)$ - условие включения или отключения элемента Φ_j (условие выполнимости элемента);

I_j - входные или выходные данные элемента Φ_j.

По правилам интерпретации базиса каждому элементу, и как следствие, состоянию системы ставится в соответствие допустимый информационный вектор

$$\mathcal{I}(u_i) = \cup_{l=0,1,2,...} Out\ (\Phi_l),$$

где $l < i$ и l пробегает множество индексов состояний, предшествующих i-му. Φ_l- базовый элемент, включаемый в l-м состоянии. Назовем $\lambda(u_i)$ - информационым вектором u_i-го состояния системы.

Интерпретированная схема <P, Int> есть последовательность, упорядоченная по элементам функционального базиса Φ_i. Она содержит полную информацию о состоянии системы как состоянии ее элементов, но не учитывает последовательности смены состояний во времени. Необходимость этого вытекает из природы элементов функционального базиса. Это как правило, приборы, работающие во времени, и все изменения их состояний также протекают во времени. Начиная с некоторого начального состояния в системе происходят его изменения до тех пор, пока не наступит заключительное состояние.

С целью приведения последовательности смены состояний к "стандартной", изменяющейся во времени, поступим следующим образом. По правилам интерпретации на временной оси каждому элементу Φ_i соответствует время его включения $t_{вкл}$ и отключения $t_{откл} = t_{вкл} + \tau$, где τ- длительность выполнения элемента Φ_i. Пусть эти моменты указаны в единой шкале времени. Упорядочим их по возрастанию для всех элементов схемы < P, Int >. При этом предполагается, что моменты включения и/или отключения различных элементов согласованы между собой. Упорядоченную по времени t-последовательность состояний системы

$$C = < t_i, u_i >, \quad i=0,1,...,к \qquad (1.1)$$

назовем стандартной схемой системы в базисе < Ф, I >. Длина С- последовательности может не совпадать с длиной

$<P, Int>$- последовательности, так как для некоторых элементов $Ф_i$ время включения может быть одинаковым.

Две системы из одного базиса называются эквивалентными, если по одному и тому же входному информационному вектору они формируют одинаковый выходной вектор.

Набор $C^1 = <\theta_j, v_j>$ назовем подсхемой для стандартной схемы $C =< t_i, u_i >$, если в схеме C $\exists$ $u_{тек}(t_i)$ такое, что $\theta_j = t_i$, и для $\forall$ $j=1,2,..$ $v_j \subseteq u_i$. ($C^1 \subseteq C$).

Реализацией стандартной схемы C назовем протокол изменения состояний системы. Ему соответствует начальное состояние системы:

$(t_0, u_0) = (Ф_0, L(Ф_0), I_0)$, в котором:

t_0- начальный момент времени, соответствующий включению нулевого элемента.

$L(Ф_0)$ - условие включения нулевого элемента $Ф_0$. Предполагается, что для нулевого момента времени это условие тождественно истинно, $L(Ф_0) \equiv TRUE$.

I_0 - набор значений исходных данных.

t_i, $i=1,...,k$ - моменты времени включения тех элементов $Ф_j$, для которых $t_{вкл}(Ф_j)=t_i$ и $L(Ф_j) = TRUE$, иначе переход к следующему моменту t_{i+1}. Если $i=k$ и к этому моменту для каждого элемента, вошедшего в реализацию, число его включений равно числу завершений, то u_k - последнее заключительное состояние протокола (протокол конечен), работа системы нормально завершается. Это означает, что функционирование системы прошло за конечное время; в информационном пространстве сформирован вектор переменных $\mathcal{X}=(x_1,...,x_m)$; в логическом пространстве вектор - $\mathcal{L}$, определяющий реализацию схемы.

В противном случае, будем считать, что функционирование системы закончилось аварийно. Т.е. хотя бы один элемент системы не завершил своей работы, либо время работы хотя бы одного элемента превысило время последнего изменения состояния системы.

Всякую реализацию стандартной схемы $C = < t_i, u_i >$ системы в заданном базисе $< Ф, I >$ назовем законом ее функционирования.

Стандартной схеме $C = < t_i, u_i >$ соответствует множество реализаций. Определим его следующим образом. Каждому состоянию $u_i = < t_i, (Ф_j, L(Ф_j), I_j) >$ стандартной схемы $C = < t_i, u_i >$ соответствует не более $n(u_i)=2^{/Фi/}$ вариантов его реализации, где $/Ф^i/$ - число элементов $Ф_j$ i-го состояния таких, что $L(Ф_j) = TRUE$.

Так как система проходит через все состояния в различные моменты времени, то верхняя оценка числа всех реализаций стандартной схемы $r(C)$ есть

$$r(C) = \Pi_{i=1,\dots,n} n(u_i).$$

Пример1. Рассмотрим стандартную схему:

$(t_0, u_0) = < Ф_0, 1, I_0 >$

$(t_1, u_1) = < (Ф_{11}, L_{11}, I_{11}), (Ф_{12}, L_{12}, I_{12}) >$

$(t_2, u_2) = < (Ф_{21}, L_{21}, I_{21}), (Ф_{22}, L_{22}, I_{22}), (Ф_{32}, L_{32}, I_{32}) >$

$(t_3, u_3) = < (Ф_{31}, L_{31}, I_{31}), (Ф_{32}, L_{32}, I_{32}) >$

$(t_4, u_4) = < Ф_0, 1, I_0 >.$

Для состояния u_1 множество вариантов реализаций есть $2^{/Ф1/}$, $Ф_1 = \{ Ф_{11}, Ф_{12} \}$; $n(u_1) = 2^2$,

если $L(Ф_{11}) \neq 0$ и $L(Ф_{12}) \neq 0$. Вот эти варианты.

$$w(u_1) = \{\varnothing, (Ф_{11}), (Ф_{12}), (Ф_{11}, Ф_{12})\}.$$

Здесь в круглых скобках перечислены подмножества одновременно включаемых элементов. Для схемы $C = < t_i, u_i >$ число всех реализаций есть

$$r(C)= n(u_0)*n(u_1)*n(u_2)*n(u_3)*n(u_4)=128.$$

Здесь $n(u_0)=n(u_4)=1$, так как для Φ_0 $L(\Phi_0)$ =TRUE как при включении (u_0), так и при отключении (u_4).

Если для состояния u_1 $L(\Phi_{11})$=false либо $L(\Phi_{12})$=false, то $n(u_1)=1$ и $r(C) = n(u_0)*n(u_1)*n(u_2)*n(u_3)*n(u_4)=64$. Практически число всех реализаций меньше расчетного, так как для некоторых состояний u_i число вариантов $n(u_i)$ меньше полного булеана. Это следует из того, что условия, накладываемые на пары элементов Φ_j, Φ_1 могут быть альтернативными ($\alpha=1$ и $\alpha=0$). А это означает, что на множестве всех вариантов пара (Φ_j, Φ_1) отсутствует. На рис.1.2 представлено дерево вариантов рассматриваемого примера.

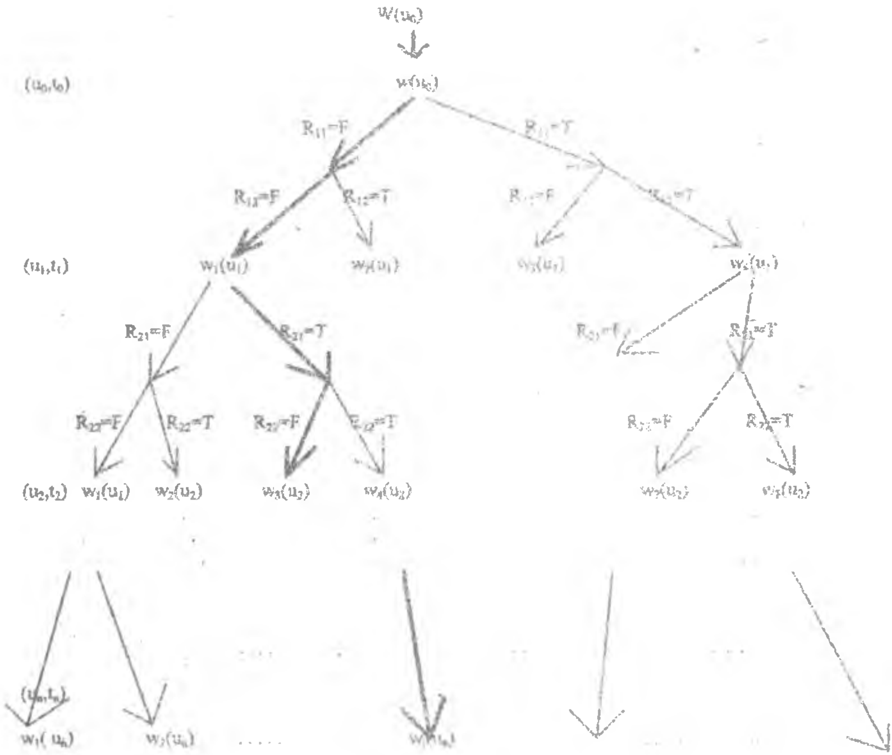

Рис.1.2 Дерево вариантов реализации стандартной схемы

1.5 Определение целевой задачи

Функционирование автономной технической системы направлено на решение заданной целевой задачи (ЦЗ) либо их совокупности. Её (ЦЗ) выполнение выражается согласованной работой приборов и устройств системы, входящих в описание ЦЗ, т.е. ЦЗ функционирует в заданном базисе или, другими словами, в заданной стандартной схеме.

Обозначим:

$u_{нач}$ - начальное состояние системы. Оно соответствует работе системы на холостом ходу и в этом смысле это состояние единственно;

$u_{тек}$- текущее состояние системы в момент времени $t_{тек}$. Это состояние, в которое может перейти система после реализации i-го шага стандартной схемы.

Введем:

$u_{цел}$ - целевое состояние системы; это некоторое подмножество состояний,- переход системы в которое означает выполнение целевой задачи.

Целевой задачей назовем преобразование

$$ЦЗ. \ z_{нач} \to z_{кон}, \qquad (1.2)$$

выполненное на стандартной схеме C (в заданном функциональном базисе), и такое, что выполняются следующие условия:

1. $z_{нач} \subseteq u_{тек}(t_i)$;

2. $z_{кон} \subseteq u_{тек}(t_j)$;

3. $t_i < t_j$,

где $z_{нач}$ - начальное состояние целевой задачи;

$z_{кон}$ - конечное состояние целевой задачи.

Целевому состоянию системы также соответствует информационный вектор $J(u_{цел})$. При этом, если элементы информационного вектора текущего состояния $u_{тек}$ выражают текущие значения измеряемых параметров, то элементы информационного вектора целевого состояния выражают требуемые значения этих параметров. Если состояние системы будет изменяться в сторону получения требуемых (целевых) значений этих параметров, то система достигнет целевого состояния, в противном случае целевая задача не будет выполнена.

О п р е д е л е н и е 1. Целевая задача существует в стандартной схеме $C = < t_i, u_i >$, если в ней имеются такие состояния $u_{тек}(t_i)$ и $u_{тек}(t_j)$, $i \neq j$, $t_i < t_j$, что

-$z_{нач} \subseteq u_{тек}(t_i)$;

-$z_{кон} \subseteq u_{тек}(t_j)$.

О п р е д е л е н и е 2. Целевая задача допустима в стандартной схеме $C = < t_i, u_i >$, если она существует в ней и ее состояния $z_{нач}$ и $z_{кон}$ включены в такие варианты $w(u_{тек}(t_i))$ и $w(u_{тек}(t_j))$, которые одновременно принадлежат хотя бы одной реализации $w(C)$ стандартной схемы $C = < t_i, u_i >$.

О п р е д е л е н и е 3. Целевая задача выполнима в стандартной схеме $C = < t_i, u_i >$, если она существует и допустима в этой схеме и

$$z_{кон} \in u_{цел}.$$

Справедливо следующее у т в е р ж д е н и е 1. Для выполнимости ЦЗ необходимо и достаточно существования стандартной схемы $C = < t_i, u_i >$, $i=0,1,...,k$ такой, что

$z_{нач} \subseteq u_i$,

$z_{нач} \subseteq u_1$, и ее реализации $w(C)$ такой, что

$w(u_i) \in w(C)$,

$w(u_j) \in w(C)$ и

$z_{кон} \in u_{цел}$.

Неформально целевая задача есть подсхема стандартной схемы. Действительно, начальному состоянию $z_{нач}$ соответствует некоторый момент времени t_i и некоторый информационный вектор начальных (исходных) параметров, при наличии которых данное состояние достижимо. Целевая задача выполняется по законам системы, т.е. путем перехода из одного состояния в новое в следующий момент времени. Такая смена состояний происходит последовательно до наступления состояния $z_{кон}$ по законам стандартной схемы. Истинность включения $z_{кон} \in u_{цел}$ определяется правильностью описания ЦЗ.

1.6 Постановка задачи асинхронного управления

При рассмотрении целевой задачи ничего не сказано о механизме перехода от i-го состояния к i+1-му. Предполагается, что этот механизм существует. Исследование этого механизма приводит нас к построению функции управления целевой задачей.

Процесс функционирования ЦЗ, так же как и системы есть смена ее состояний, происходящая в асинхронные моменты $\{t_0, t_1, ..., t_k\}$.

Набор действий в момент времени t_i, приводящий к изменению состояния системы и к переходу от t_i- го момента к следующему (i+1)- му назовем управляющим воздействием.

Последовательность таких воздействий, происходящих в моменты $\{t_0, t_1, \ldots, t_k\}$, назовем функцией управления целевой задачей. Функцию управления ЦЗ можно рассматривать как механизм выполнения ЦЗ.

Так как элементами ЦЗ являются элементы функционального базиса стандартной схемы, то и функционируют они также во времени, в логическом и информационном пространствах, соответственно и управление ими производится в этих же пространствах.

Управление во времени сводится к включению и/или отключению элемента системы в соответствующий момент времени, в логическом пространстве управление есть проверка логических признаков. В информационном пространстве процесс управления сводится к формированию данных и/или их передаче на вход элемента системы, к формированию данных и/или их записи после завершения работы элемента системы.

Выражением функции управления является совокупность команд из фиксированного множества команд. Если строить функцию управления ЦЗ в системе команд некоторого наперед заданного процессора, то процесс формирования команд управления ЦЗ назовем процессом создания функции управления ЦЗ. Упорядоченную последовательность команд, реализующих функцию управления, назовем программой управления ЦЗ. Процесс формирования программы управления, исполнение которой приводит к решению ЦЗ, назовем задачей асинхронного управления.

Задача асинхронного управления ставится следующим образом.

Дано - стандартная схема С= $<t_i, u_i>$ в базисе $< Ф, I>$,

целевая задача ЦЗ: $z_{нач} \to z_{кон}$;

$u_{цел}$ - целевое множество.

Требуется построить алгоритм асинхронного управления целевой задачей (1.2), заданной в стандартной схеме С (1.1), и сформировать программу управления средствами заданной системы команд.

2. Разработка математической модели поведения системы

2.1 Основные положения

Выше исследовалось поведение системы с целью построения ее закона функционирования. Был задан класс исследуемых систем путем введения функционального базиса (базовых элементов) и правил его интерпретации в информационном, логическом и временном пространствах. Выбор области интерпретации позволяет приписывать к одному классу разнотипные системы. Было введено понятие целевой задачи и ее функции управления. Поведение системы рассматривается как выполнение ею некоторого множества целевых задач. В качестве модели функционирования была предложена стандартная схема в заданном функциональном базисе элементов. При этом открытым остался вопрос - как в ней реализуется целевая задача. Выполнение целевой задачи под действием функции управления (задача асинхронного управления) реализуется в виде подсхемы стандартной схемы.

Ниже строится функциональное исчисление, являющееся теоретической основой механизма реализации стандартной схемы. На его основе создается и исследуется термальное описание целевой задачи. Это формализованное синтаксически строгое ее представление, доступное для анализа программными средствами. Показано, как такое описание в процессе ряда преобразований приводится к стандартной схеме.

2.2 Функциональное исчисление

Опишем множество целевых задач в заданном базисе элементов системы. Каждый объект исчисления есть формализованное описание целевой задачи. На временной оси работу каждого элемента будем интерпретировать моментом времени включения и длительностью. На базовом множестве элементов введем бинарные операции: $\to$ следования, СН - совпадения по началу, СК - совпадения по концу, + - операцию выбора динамического объекта; унарные операции: $\Rightarrow$ - навешивания предиката на терм (создание динамического объекта), IN, OUT - операции навешивания входных и выходных переменных на терм.

Множество целевых задач (или термов) определяется рекурсивно: символ элемента системы есть терм либо целевая задача; если α - предикат, а T_1, T_2 - термы, то $T_1 \to T_2$, T_1 СН T_2, T_1 СК T_2, $(\alpha \Rightarrow T_1 + /\alpha \Rightarrow T_2)$ - термы [5,9].

Множество предикатов, навешиваемых на терм, определяется рекурсивно: символ логической переменной со значением - предикат, если α, β - предикаты, то α .or. β, α .and. β, $/\alpha$ - предикаты.

Неформально каждый терм описывает целевую задачу на отрезке временной оси в заданном базисе. Операция следования ($\to$) означает последовательную работу первого и второго термов. Второй терм начинает работу после окончания работы первого. Операция совпадения по началу означает одновременное включение двух термов. Операция совпадения по концу означает одновременное завершение работы двух термов. Операция навешивания предиката на терм есть введение динамического терма, т.е. терма, включаемого в работу в случае истинности предиката.

Порядок выполнения операций в терме подчиняется правилу скобок.

Два терма называются эквивалентными [7,18], если они решают одну и ту же целевую задачу (если они построены в одном и том же базисе и в заданный момент времени формируют одинаковый информационный вектор).

На множестве термов правильно построенные формулы имеют вид $T_1 = T_2$, где T_1, T_2 - термы.

Формула исчисления называется правильной, если она выводима в исчислении. В качестве аксиом исчисления введем следующие правильно построенные формулы:

- аксиомы коммутативности

T_1 СН T_2 = T_2 СН T_1

T_1 СК T_2 = T_2 СК T_1

$T_1 + T_2 = T_2 + T_1$;

- аксиомы ассоциативности

$(T_1 \to T_2) \to T_3 = T_1 \to (T_2 \to T_3)$

(T_1 СН T_2) СН T_3 = T_1 СН (T_2 СН T_3)

(T_1 СК T_2) СК T_3 = T_1 СК (T_2 СК T_3);

- аксиомы дистрибутивности

($T_1 \to T_2$) СН ($T_1 \to T_3$) = $T_1 \to$ (T_2 СН T_3)

($T_1 \to T_2$) СК ($T_3 \to T_2$) = (T_1 СК T_3) $\to T_2$

$(T_1 \to T_2) + (T_1 \to T_3) = T_1 \to (T_2 + T_3)$

$(T_1 \to T_2) + (T_3 \to T_2) = (T_1 + T_3) \to T_2$

(T_1 СН T_2) + (T_1 СН T_3) = T_1 СН ($T_2 + T_3$)

(T_1 СН T_2) + (T_3 СН T_2) = ($T_1 + T_3$) СН T_2

(T_1 СК T_2) + (T_1 СК T_3) = T_1 СК ($T_2 + T_3$)

$$(T_1 \text{ CK } T_2) + (T_3 \text{ CK } T_2) = (T_1 + T_3) \text{ CK } T_2$$

$(\alpha_1=1) \Rightarrow (T_1 @ T_2) = ((\alpha_1=1) \Rightarrow T_1) @ ((\alpha_1=1) \Rightarrow T_2)$,

где $@ \in \{ +, \text{CH}, \text{CK} \}$;

- аксиомы идемпотентности

$$T_1 \text{ CH } T_1 = T_1$$

$$T_1 \text{ CK } T_1 = T_1$$

Введем пустой терм (λ) длительности (τ). Формально он интерпретируется как невыполняемый терм. Неформально это означает задержку на время (τ).

Введем терм нулевой длительности (θ). Формально он интерпретируется как невыполняемый. Неформально это означает синхронизацию терма с некоторым наперед заданным моментом времени.

Для любого терма T и произвольных предикатов a, b справедливы аксиомы

$$(\alpha \Rightarrow T) + (/\alpha \Rightarrow T) = T$$

$$(\alpha \Rightarrow (\beta \Rightarrow T)) = (\alpha \text{ and } \beta) \Rightarrow T.$$

Новые правильно построенные формулы выводятся из аксиом по правилам булевой алгебры (для предикатов), по правилу подстановки (для термов и предикатов), а также по правилам, вытекающим из свойств равносильности (=) формул.

2.3 Семантика термального описания целевой задачи

Формально функциональное исчисление интерпретируется следующим образом [5]. Фиксируется непустой универсум U = { D } ∪{ 0,1 } ∪ { τ }, как объединение непересекающихся множеств.

Фиксируются конечные множества символов элементов системы Ф, логических переменных Р, информационных переменных Х

Фиксируются отображения:

I_t [Ф]: Ф → t, ставящее в соответствие каждому терму его положение на временной оси,

I_f [Ф]: D^m → D^k, описывающее каждый функциональный элемент как преобразователь информации; m-мерный вектор входных переменных преобразуется в выходной k-мерный вектор;

I_p [α], ставящее в соответствие каждой логической переменной $\alpha \in P$ предикатную функцию Ip [α]: U → { 0,1 };

I_p [Ф], ставящее в соответствие каждому функциональному элементу $Ф_i \in Ф$ условие включения I I_p [Ф]: Ф → L; условие включения образуется на множестве логических переменных по законам булевой алгебры;

I_f [x]: x → d, $d \in D$, ставящее в соответствие информационной переменной ее значение на числовой оси;

I_x [x], ставящее в соответствие каждой информационной переменной $x \in X$ функцию принадлежности входному либо выходному вектору переменных I_x [x]: U →{ IN,OUT }.

Семантика логических переменных, информационных переменных, термов и операций над ними определяется следующим образом:

Ip [/α] = /Ip [α]

Ip [α @ β] = Ip [α] @ Ip [β]

для любого @ ∈ { and, or } и любых предикатов α, β .

Пусть n - число логических переменных в терме. Тогда, областью определения терма в логическом пространстве является булеан на множестве из n переменных $\mathcal{B}(n)$ либо его подмножество. Каждый элемент булеана, в котором терм определен, является условием одной реализации терма.

$I_x[x] = IN(\Phi_i)$, либо $I_x[x] = OUT(\Phi_i)$,

для любых $x \in X$, $\Phi_i \in \Phi$

$I_t[\Phi_i] = t_{вкл}(\Phi_i)$

$I_t[\Phi_1 \to \Phi_2] = (t_{вкл}(\Phi_1)\ \ t_{вкл}(\Phi_2))$, по определению операции следования $t_{вкл}(\Phi_2) = t_{вкл}(\Phi_1) + \tau_1$.

$I_t[\Phi_1\ СН\ \Phi_2] = (t_{вкл}(\Phi_1,\ t_{вкл}(\Phi_2))$, по определению операции совпадения по началу

$t_{вкл}(\Phi_2) = t_{вкл}(\Phi_1)$.

$I_t[\Phi_1\ СК\ \Phi_2] = (t_{вкл}(\Phi_1)\ t_{вкл}(\Phi_2))$, по определению операции совпадения по концу

$$t_{вкл}(\Phi_2) = \begin{cases} t_{вкл}(\Phi_1) - (\tau_2 - \tau_1), & \text{если } \tau_1 < \tau_2, \\ t_{вкл}(\Phi_1) & \text{если } \tau_1 = \tau_2, \\ t_{вкл}(\Phi_1) + (\tau_1 - \tau_2), & \text{если } \tau_1 > \tau_2. \end{cases}$$

$$It[\alpha \Rightarrow \Phi_1] = \begin{cases} It[\Phi_1], & \text{если } \alpha = TRUE, \\ \lambda, & \text{если } \alpha = FALSE, \end{cases}$$

λ - пустой терм,

τ_1, τ_2 - длительности термов Φ_1, Φ_2 соответственно.

В исчислении целевая задача может быть представлена совокупностью термальных выражений, например:

$T_1 := \Phi_1 \to \Phi_2$

$T_2 := (\alpha_1 = 1) \Rightarrow T_1 + (\alpha_1 = 0) \Rightarrow \Phi_3$

$T_3 := \Phi_4\ СН\ \Phi_5$

$T_4 := T_2 \to ((\alpha_2 = 1) \Rightarrow T_3 + \lambda)$.

Примечание. Термальное выражение целевой задачи не содержит информационной компоненты. Она предполагается заданной для каждого функционального элемента.

2.4 Неформальная интерпретация термального описания целевой задачи

Всякий терм можно представить бинарным деревом входящих в него подтермов $\{ d_i \}$ и отношением их связности. Формально это можно записать следующим образом

$$TB = < \{ d_i \}, r >,$$

где $\{ d_i \}$ - множество всех подтермов, включающее и терм T;

r - отношение связности на $\{ d_i \}$.

Каждый подтерм определим следующим образом:

$$d_i = < d_{i1} @ d_{i2} >,$$

где d_{i1}, d_{i2} - подтермы нижнего уровня либо функциональные элементы, $d_{i1} \in \{ d_i \}$, $d_{i2} \in \{ d_i \}$;

@ - операция функционального исчисления над подтермами, $@ \in \{ \to, \text{СН}, \text{СК}, + \}$.

Пример2. Терму

$T=(\alpha_1=1) \Rightarrow (\Phi_1 \to \Phi_2) + (\alpha_1=0) \Rightarrow (\Phi_3 \text{ СН } ((\alpha_2=1) \Rightarrow \Phi_4 + (\alpha_2=0) \Rightarrow \Phi_5))$ поставим в соответствие $TB = < \{ d_i \}, r >$,

где $\{ d_i \} = \{ d_1, d_2, d_3, d_4 \}$,

$d_1 = < d_2 + d_3 >$

$d_2 = < \Phi_1 \to \Phi_2 >$,

$d_3 = < \Phi_3 \text{ СН } d_4 >$,

$d_4 = < \Phi_4 + \Phi_5 >$;

$r = \{ (d_1,d_2), (d_1,d_3), (d_2,\Phi_1), (d_2,\Phi_2), (d_3,\Phi_3), (d_3,d_4),$

(d_4,Φ_4), (d_4,Φ_5) }.

Операция @ = "+" объединяет два динамических подтерма по одной логической переменной с разными значениями. Отношение связности между динамическим термом (d_1) и его компонентами (d_2, d_3) будем называть помеченным соответствующими значениями логических переменных. Для нашего примера связь (d_1,d_2) помечена значением ($\alpha_1=1$), связь (d_1,d_3) соответственно ($\alpha_1=0$), связь (d_4,Φ_4) - значением ($\alpha_2=1$), связь (d_4,Φ_5) - значением ($\alpha_2=0$).

Для каждого терма путем назовем упорядоченную последовательость отношений связности, идущих от терма верхнего уровня до висячей вершины типа функциональный элемент. Число всех путей на терме определяется числом всех висячих вершин. Для нашего примера имеем множество путей

$$P = \{ p_1, p_2, p_3, p_4, p_5 \},$$

где $p_1 = \{(d_1,d_2), (d_2,\Phi_1)\}$,

$p_2 = \{(d_1,d_2), (d_2,\Phi_2)\}$,

$p_3 = \{(d_1,d_3), (d_3,\Phi_3)\}$,

$p_4 = \{(d_1,d_3), (d_3,d_4), (d_4,\Phi_4)\}$,

$p_5 = \{(d_1,d_3), (d_3,d_4), (d_4,\Phi_5)\}$.

Путь называется помеченным логическим условием, если помечено хотя бы одно отношение этого пути. Логическое условие, которым помечен путь, будем называть также условием реализации этого пути. Если для некоторого пути имеется несколько отношений, помеченных разными логическими переменными, то логическим условием или условием реализации назовем конъюнкцию значений этих переменных. Выпишем полные условия реализации для всех путей приведенного примера:

$L_1 = (\alpha_1=1)$,

$L_2 = (\alpha_1=1)$,

$L_3 = (\alpha_1=0)$,

$L_4 = (\alpha_1=0)(\alpha_2=1)$,

$L_5 = (\alpha_1=0)(\alpha_2=0)$.

2.5 Исследование термального описания целевой задачи

Исследование поведения системы, выполняющей целевую задачу, опирается на ее термальное описание [10,11,12,13]. Оно позволяет определить асинхронную последовательность моментов смены состояний системы и набор функциональных элементов для каждого состояния. Полное исследование процесса функционирования системы невозможно без всестороннего исследования терма целевой задачи как на синтаксическом, так и семантическом уровнях. Так привязка состояний системы ко времени не решается на синтаксическом уровне. Без знаний длительностей выполнения функциональных элементов невозможно согласовать их на временной оси. Порядок их включения во времени не соответствует их порядковой записи в термальном выражении. Здесь требуется погружение в семантику задачи.

Терм $T=T(\Phi,\alpha)$ описывает взаимодействие k функциональных элементов $\Phi = \{\Phi_1,...,\Phi_k\}$ в n- мерном логическом пространстве

$$\alpha = \{\alpha_1,..., \alpha_n\}, \quad \alpha_i \in \{0,1\}, \; i=1,...,n$$

Определение логического пространства. Каждая логическая переменная α_i определяет некоторое условие, которое в зависимости от ее области определения, может накладываться на один либо несколько функциональных элементов. В свою очередь, в условие выполнимости функционального элемента может входить одна либо более одной логической переменной. Поэтому множество условий выполнимости для всех базовых элементов является подмножеством булеана $\mathcal{B}(n) = 2^{/n/}$.

Например, условие $L_1 = (\alpha_1=0).and.(\alpha_2=1)$ (в дальнейшем, там, где это не вызывает неоднозначного толкования, операцию .and. будем опускать, т.е. будем писать $L_1 = (\alpha_1=0)(\alpha_2=1)$) есть элемент булеана $\mathcal{B}(2)$ от двух логических переменных α_1, α_2. Это условие истинно, если истинно выражение $(\alpha_1=0).and.(\alpha_2=1)$, в противном случае оно ложно. Над условием L_1, как над элементом булеана $\mathcal{B}(n)$, можно проводить операции булевой алгебры.

Условие L_i есть характеристика элемента Φ_i, определенная на булеане $\mathcal{B}(n)$ и означающая в каких точках булеана это условие истинно и элемент Φ_i должен включаться, а в остальных точках булеана это условие ложно и элемент Φ_i включаться не должен. В термальном выражении условия могут накладываться как на отдельные функциональные элементы, так и на их совокупности, являясь средством выражения их согласованной работы. С целью выделения полного логического вектора для каждого функционального элемента по термальному выражению $T=T(\Phi,\alpha)$ строится функция выполнимости $F_1: \Phi \rightarrow \mathcal{B}(n)$, которая каждому элементу функционального базиса ставит в соответствие его условие выполнимости.

Однако булеан $\mathcal{B}(n)$ как характеристическая область элемента $Ф_i$ не полон. Например, на булеане $\mathcal{B}(n)$ не существует ситуации с безусловным включением элемента $Ф_i$; не существует ситуации условного включения элемента $Ф_i$ по переменной α_1, независимо от переменной α_2. Для полного определения характеристической области элемента $Ф_i$ введем следующее расширение - каждую логическую переменную α_i будем задавать одним из трех значений $\alpha_i \in \{ 0, 1, н \}$. Т.е. логическая переменная может принимать ложное значение $\alpha_i=0$, истинное значение $\alpha_i=1$ и независимое значение $\alpha_i=н$. В дальнейшем сохраним выражение булеан $\mathcal{B}(n)$, полагая, что каждая переменная принимает значение в троичном базисе и $/\mathcal{B}(n)/=3^n$. Выражение функции выполнимости при этом не изменится.

Определение информационного пространства. Информационное пространство терма зададим в виде декартова произведения D^n, где n- число переменных, D- область значений одной переменной.

Всякий функциональный элемент $Ф_i$ есть преобразователь вида $Ф_i$: $D^m \rightarrow D^k$, где m(k)- число входных (выходных) переменных и $m+k =< n$. При включении элемента $Ф_i$ вектор I_i задает входные данные, необходимые для его работы. Если к моменту времени t_i они определены, то требуется их чтение из общего поля памяти. Если они не определены, то требуется их инициализация. Выходные данные после отключения элемента сохраняются в общем поле памяти.

Определение множества вариантов выполнения терма. Рассмотрим терм $T= Ф_1$. Функционированию такого терма во времени соответствует циклограмма на рис.2.1. В момент времени t_0 происходит безусловное включение элемента $Ф_1$, в момент $t_1=t_0+\tau$ его безусловное отключение.

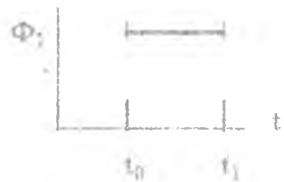

Рис.2.1. Циклограмма терма $T=Ф_1$

Терму $T=Ф_1$ соответствует однозначная привязка к временной оси. При безусловном включении и отключении элементов терму соответствует единственный вариант его выполнения. Терму $T=Ф_1$ соответствует единственный вариант.

Рассмотрим терм $T=(\alpha_1=1)\Rightarrow Ф_1+(\alpha_1=0)\Rightarrow Ф_2$. Зафиксируем значение логической переменной $\alpha_1=1$. Тогда терм будет иметь вид $T=Ф_1$ и ему будет соответствовать однозначная привязка к временной оси. Аналогично, при $\alpha_1=0$ однозначно будет выполняться терм $T=Ф_2$. Таким образом исходному терму соответствуют два варианта выполнения: $W_1(T)=Ф_1$ и $W_2(T)=Ф_2$. Возникает вопрос, сколько вариантов выполнения имеет терм, содержащий n- логических переменных. Процесс преобразования терма в его варианты назовем параметризацией терма в логическом пространстве. Вектор значений логических переменных, при которых формируется один вариант, называется условием реализации терма. Очевидно, число вариантов выполнения терма равно числу его условий реализации.

2.5.1 Построение функции выполнимости

Значения логических переменных, навешиваемых на элементы системы, задают условия включения их в работу. На один элемент может быть навешено несколько логических переменных. Для согласованного между собой включения элементов условия могут накладываться на их пары, тройки и т.д. При истинности условия элемент может быть включен в работу. Так как при реализации стандартной схемы решение о включении принимается отдельно по каждому элементу системы, то естественно возникает вопрос, какие логические переменные навешены на каждый элемент. Ответом на этот вопрос является функция выполнимости

F_1: Ф $\rightarrow$ 3^n, которая каждому функциональному элементу терма Т=Т(Ф,α) ставит в соответствие условие его выполнимости.

У т в е р ж д е н и е 2 Термальное описание целевой задачи позволяет построить функцию выполнимости для каждого элемента функционального базиса, входящего в терм.

Доказательство утверждения вытекает из построения функции выполнимости. Построение функции выполнимости сводится к одному проходу по терму целевой задачи. При этом выделяются все функциональные задачи и каждой из них ставится в соответствие логический вектор. С этой целью приведем терм к каноническому виду.

О п р е д е л е н и е 4. Каноническим называется представление терма в виде многочлена по элементам функционального базиса относительно термальных операций.

В канонической форме предикат, навешенный на функциональный элемент $Ф_i$, является условием его выполнимости. Его истинность определяется истинностью логического выражения как элемента булевой алгебры.

Алгоритм построения функции выполнимости.

п1.Приведение терма к каноническому виду.

п2.Выделение логического выражения, навешенного на функциональный элемент.

Пример3 $T=[(\alpha_1=1) \Rightarrow((\alpha_2=1) \Rightarrow\Phi_1 + (\alpha_2=0) \Rightarrow \Phi_2) +$

$(\alpha_1=0) \Rightarrow \Phi_5] \to(\Phi_3$ СН Φ_4).

Каноническая форма терма

$T=[(\alpha_1=1)(\alpha_2=1) \Rightarrow \Phi_1 + (\alpha_1=1)(\alpha_2=0) \Rightarrow \Phi_2 +(\alpha_1=0)\Rightarrow\Phi_5]$

$\to \Phi_3$ СН Φ_4.

$\alpha=\{ \alpha_1, \alpha_2 \}$.

$F_1: \Phi \to 3^2 \quad \Phi_1 \to (\alpha_1=1)(\alpha_2=1)$

$\Phi_2 \to (\alpha_1=1)(\alpha_2=0)$

$\Phi_3 \to (\alpha_1=н)(\alpha_2=н)$

$\Phi_4 \to (\alpha_1=н)(\alpha_2=н)$

$\Phi_5 \to (\alpha_1=0)(\alpha_2=н)$

2.5.2 Параметризация терма в логическом пространстве

О п р е д е л е н и е 5. Терм $T(\Phi,\alpha)$ называется динамическим, если в него входит хотя бы одна логическая переменная. В противном случае терм называется статическим.

Функционированию статического терма соответствует единственный переход из начального состояния в конечное S(вх,Т) → S(вых,Т). Функционированию динамического терма альтернативно соответствует несколько переходов S(вх,Т) → { S(вых,Т) }. Переход терма из начального состояния в одно из конечных назовем вариантом выполнения терма. Выполнение терма осуществляется по одному из допустимых вариантов. Всякий вариант есть подтерм основного терма при заданном условии реализации. Полное множество вариантов терма определяется совокупностью условий реализации и их число не более чем 2^n, где n- число логических переменных. На практике это число может быть существенно меньше. Выбор одного варианта называется параметризацией терма в логическом пространстве. Что, в свою очередь, означает формирование подтерма по заданному набору значений логических переменных.

Терму $T = T(\Phi,\alpha)$ поставим в соответствие множество вариантов выполнения F_2: $T \to W(T,\mathcal{R})$. Каждый вариант из $W(T,\mathcal{R})$ определяется условием его реализации $\mathcal{R}(\alpha)$, заданным в пространстве логических переменных.

Для терма $T=\Phi_i$ условие его реализации тождественно истинно, так как в записи терма нет никаких условий и ограничений на работу элемента Φ_i. Терму $T=\Phi_i$ соответствует единственный вариант его выполнения. Для варианта безусловного включения его условие реализации будем записывать в виде $\mathcal{R}_0$. Таким образом, для $T=\Phi_i$ имеем

$W(T=\Phi_i)=\{w_1(\Phi_i)$ при условии $\mathcal{R}_0\}$.

Для терма вида $T= (\alpha=1) \Rightarrow\Phi_1 + (\alpha=0) \Rightarrow \Phi_2)$ множество вариантов его выполнения есть

$W(T) =\{ w_1(T)=\Phi_1$ при условии $\mathcal{R}_1(\alpha) =(\alpha=1)$,

$w_2(T) = \Phi_2$ при условии $\mathcal{R}_2(\alpha) =(\alpha=0)\}$.

Число условий реализации для динамического терма равно сумме условий для подтермов, связанных операцией выбора (+). Это вытекает из того факта, что, по определению динамического терма, его условия реализации носят альтернативный характер. И их объединение дает полное множество условий реализации для динамического терма.

Обозначим: $\mathcal{R} = \{ \mathcal{R}_0, \mathcal{R}_1, ..., \mathcal{R}_m \}$ - множество условий реализации терма Т. $\mathcal{R} \subseteq \mathcal{B}(n)$, где n- число логических переменных.

$\mathcal{R}_0 = (\alpha_1 = н).and...and.(\alpha_n = н)$- условие безусловного включения и/или отключения элемента;

$\mathcal{R}_j$ - условие j-й реализации терма, $j=1,...,m$;

$\mathcal{R}_0 \subseteq \mathcal{R}_j$ для любого $j=1,...m$; т.е функциональные элементы с безусловным включением входят в каждый вариант выполнения терма.

Так как каждый вариант выполнения терма определяется его условием реализации, то очевидно, что два условия $\mathcal{R}_1$ и $\mathcal{R}_2$ называются различными, если хотя бы одна логическая переменная принимает на них различные значения. Исходя из природы условия реализации и его определения нетрудно заметить, что для любого терма его полное множество условий реализации удовлетворяет следующим равенствам:

$$(\bigcup_{j=0,1,...,m} \mathcal{R}_j) \cong TRUE \tag{2.1}$$

(полнота множества всех условий реализации терма).

$$(\bigcap_{j=0,1,...,m} \mathcal{R}_j) \cong FALSE \tag{2.2}$$

(единственность варианта выполнения терма).

Выполнимость терма заключается в реализации хотя бы одного его варианта и вытекает из условия полноты множества всех его реализаций в логическом пространстве.

Равенство (2.1) и требование учета всех логических переменных в каждой реализации используются при формировании множества условий реализации терма.

Отличие условия выполнимости от условия реализации заключается в том, что если условие выполнимости относится к одному базовому элементу Φ_i, то условие реализации определяет всю совокупность базовых элементов, выполняемых по данному варианту терма. Очевидно, между условием выполнимости L_i функционального элемента Φ_i, входящего в j-ю реализацию, и j-м условием реализации терма выполняется включение $L_i \subseteq \mathcal{R}_j$.

Из этого соотношения вытекает, что один и тот же функциональный элемент Φ_i может входить в разные варианты выполнения, имеющие общие части такие, что

$\mathcal{R}_j \cap \mathcal{R}_k \neq \varnothing$ и

$L_i \subseteq (\mathcal{R}_j \cap \mathcal{R}_k)$.

У т в е р ж д е н и е 3. Терму $T=T(\Phi,\alpha)$, где $\Phi=\Phi_1,..., \Phi_k$ - элементы функционального базиса, $\alpha= \alpha_1,..., \alpha_n$ - логические переменные, можно поставить в соответствие не более чем 2^n его варианта.

Алгоритм построения множества вариантов.

Алгоритм строится индуктивно, где индукция берется по структуре терма.

п1. Терму $T=\Phi_i$ соответствует единственный вариант его выполнения $w_1(T) = \Phi_i$ при условии $\mathcal{R}_0$.

п2. Для терма вида $T = (\alpha_i=1) \Rightarrow T_1 + (\alpha_i=0) \Rightarrow T_2$ множество вариантов $W(T)$ есть объединение вариантов подтермов, связанных операцией выбора (+)

$$W(T) = W(T_1, \alpha_i=1) \cup W(T_2, \alpha_i=0).$$

п3. Для терма вида $T = T_1 \otimes T_2$, $\otimes \in \{ \text{СН}, \text{СК}, \rightarrow\}$, множество вариантов $W(T)$ определяется через множества условий реализации $\mathcal{R}_1(\alpha)$, $\mathcal{R}_2(\alpha)$ для подтермов T_1 и T_2, связанных операцией $\otimes$, следующим образом.

Для множеств $W(T_1)$, $W(T_2)$ определяются условия их реализации соответственно $\mathcal{R}_1(\alpha)$, $\mathcal{R}_2(\alpha)$. Между элементами последних, как на декартовом произведении строится множество $\mathcal{R}(T) = \mathcal{R}_1(\alpha) \times \mathcal{R}_2(\alpha)$. Элемент этого множества $r_i \in \mathcal{R}(T)$ есть выражение, вычисляемое по правилам булевой алгебры:

$r_i = r_{1j} \& r_{2k}$, для $\forall$ j, k, $j=1,\dots,|\mathcal{R}_1(\alpha)|$, $k=1,\dots,|\mathcal{R}_2(\alpha)|$,

$r_{1j} \in \mathcal{R}_1(\alpha)$, $r_{2k} \in \mathcal{R}_2(\alpha)$.

Все выражения $r_{1j} \& r_{2k} = 0$ удаляются из множества $\mathcal{R}(T)$. Оставшиеся элементы образуют множество $\mathcal{R}(T)$ условий реализации терма T. $|W(T)| = |\mathcal{R}(T)|$.

п4. Других правил нет.

Пример4. Рассмотрим терм из примера3:

$T=[(\alpha_1=1) \Rightarrow ((\alpha_2=1) \Rightarrow \Phi_1 + (\alpha_2=0) \Rightarrow \Phi_2) + (\alpha_1=0) \Rightarrow \Phi_5] \to (\Phi_3$ СН $\Phi_4)$.

Его можно представить как $T = T_1 \to T_2$,

где $T_1 = (\alpha_1=1) \Rightarrow T_3 + (\alpha_1=0) \Rightarrow \Phi_5$,

$T_2 = \Phi_3$ СН Φ_4,

$T_3 = (\alpha_2=1) \Rightarrow \Phi_1 + (\alpha_2=0) \Rightarrow \Phi_2$.

По правилу п3 для терма $T = T_1 \to T_2$ строим

$\mathcal{R}(T,\alpha) = \mathcal{R}(T_1,\alpha) \times \mathcal{R}(T_2,\alpha)$.

Для терма T_1 по правилу п2 имеем

$W(T_1) = W(T_3, \alpha_1=1) \cup W(\Phi_5, \alpha_1=0)$.

Для терма T_2 $\mathcal{R}(T_2, \alpha) = \mathcal{R}(\Phi_3, \alpha) \times \mathcal{R}(\Phi_4,\alpha)$. Каждое из условий $\mathcal{R}(\Phi_3, \alpha)$, $\mathcal{R}(\Phi_4,\alpha)$ состоит из единственного значения $\mathcal{R}(\Phi_3,\alpha) = \mathcal{R}_0$ и $\mathcal{R}(\Phi_4,\alpha) = \mathcal{R}_0$, поэтому и

$\mathcal{R}(T_2,\alpha)=\mathcal{R}(\Phi_3,\alpha) \times \mathcal{R}(\Phi_4,\alpha)$ также состоит из единственного значения $\mathcal{R}_0 \,\&\, \mathcal{R}_0 = \mathcal{R}_0$, $w(T_2)= \Phi_3$ СН Φ_4; $|w(T_2)| =1$.

Для терма T_3 по правилу п2 имеем

$W(T_3)=W(\Phi_1, (\alpha_1=1)(\alpha_2=1)) \cup W(\Phi_2, (\alpha_1=1)(\alpha_2=0))$.

Последнее выражение можно записать в виде

$W(T_3)= \{w_1(T_3)=\Phi_1$ при условии $\mathcal{R}(\Phi_1,\alpha)=(\alpha_1=1)(\alpha_2=1)$,

$w_2(T_3)=\Phi_2$ при условии $\mathcal{R}(\Phi_2,\alpha)=(\alpha_1=1)(\alpha_2=0)\}$; $|W(T_3)|=2$.

Для терма T_1 имеем

$W(T_1)=\{w_1(T_3)=\Phi_1$ при условии $\mathcal{R}(\Phi_1,\alpha)=(\alpha_1=1)(\alpha_2=1)$,

$w_2(T_3)=\Phi_2$ при условии $\mathcal{R}(\Phi_2,\alpha)=(\alpha_1=1)(\alpha_2=0)$,

$w_3(\Phi_5)=\Phi_5$ при условии $\mathcal{R}(\Phi_5,\alpha)=(\alpha_1=0)\}$;

т.е. $\mathcal{R}(T_1,\alpha)= \{\mathcal{R}(\Phi_1,\alpha), \mathcal{R}(\Phi_2,\alpha), \mathcal{R}(\Phi_5,\alpha)\}$, $|W(T_1)|=3$.

Для терма T имеем $\mathcal{R}(T,\alpha) = \mathcal{R}(T_1,\alpha)\times \mathcal{R}(T_2,\alpha) =$

$\{\mathcal{R}(\Phi_1,\alpha), \mathcal{R}(\Phi_2,\alpha), \mathcal{R}(\Phi_5,\alpha)\} \times \{\mathcal{R}(T_2,\alpha)\} =$

$\{ (\alpha_1=1)(\alpha_2=1)\mathcal{R}_0, (\alpha_1=1)(\alpha_2=0)\mathcal{R}_0, (\alpha_1=0)\mathcal{R}_0 \}$, следовательно $|W(T)|=3$. Покажем эти варианты

$\alpha =\{ \alpha_1, \alpha_2 \}$.

Полная таблица условий реализации терма Т имеет вид (таблица 2.1):

Таблица 2.1

α_1	α_2
1	1
1	0
0	н

Из терма Т выделим подтермы, соответствующие условиям реализации $\mathcal{R}(\alpha)$ при заданных значениях логического вектора. Для $\mathcal{R}_1= (\alpha_1=1)(\alpha_2=1)$ имеем $w_1(T, \mathcal{R}_1)= \Phi_1 \rightarrow (\Phi_3$ СН $\Phi_4)$.

для $\mathcal{R}_2 = (\alpha_1=1)(\alpha_2=0)$ имеем $w_2(T, \mathcal{R}_2)= \Phi_2 \to (\Phi_3 \text{ CH } \Phi_4)$,

для $\mathcal{R}_3 = (\alpha_1=0)(\alpha_2=н)$ имеем $w_3(T, \mathcal{R}_3)= \Phi_5 \to (\Phi_3 \text{ CH } \Phi_4)$.

Таким образом, выполнению терма T соответствуют три условия реализации $\mathcal{R}(T) = \{\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$, т.е. терму T в логическом пространстве соответствуют три параметризации

$\mathcal{R} = \{ \mathcal{R}_0 \cup \mathcal{R}_1, \mathcal{R}_0 \cup \mathcal{R}_2, \mathcal{R}_0 \cup \mathcal{R}_3 \}$,

где $\mathcal{R}_0 = (\alpha_1=н)(\alpha_2=н)$.

Нетрудно заметить, что условию безусловного включения $\mathcal{R}_0$ соответствует подтерм $w_0(T, \mathcal{R}_0)= (\Phi_3 \text{ CH } \Phi_4)$.

Пример5. Рассмотрим терм $T = T_1 \text{ CH } T_2$, где

$T_1 = [(\alpha_1=1) \Rightarrow((\alpha_2=1) \Rightarrow \Phi_1 + (\alpha_2=0) \Rightarrow \Phi_2) + (\alpha_1=0) \Rightarrow \Phi_3]$,

$T_2 = [(\alpha_2=1)\Rightarrow \Phi_6 + ((\alpha_2=0) \Rightarrow ((\alpha_3=1) \Rightarrow \Phi_4 + (\alpha_3=0) \Rightarrow \Phi_5))]$.

$\alpha=\{\alpha_1, \alpha_2, \alpha_3\}$.

Выпишем элементарные конъюнкции для терма T_1:

$\mathcal{R}(T_1,\alpha) =\{ (\alpha_1=1)(\alpha_2=1), (\alpha_1=1)(\alpha_2=0), (\alpha_1=0) \}$.

Для терма T_2:

$\mathcal{R}(T_2,\alpha) = \{ (\alpha_2=1), (\alpha_2=0)(\alpha_3=1), (\alpha_2=0)(\alpha_3=0) \}$.

Построим множество $\mathcal{R}(T,\alpha) =\mathcal{R}(T_1,\alpha)\times \mathcal{R}(T_2,\alpha)$.

$\mathcal{R}(T,\alpha)=\{(\alpha_1=1)(\alpha_2=1)(\alpha_2=1), (\alpha_1=1)(\alpha_2=0)(\alpha_2=1), (\alpha_1=0)(\alpha_2=1),$

$(\alpha_1=1)(\alpha_2=1)(\alpha_2=0)(\alpha_3=1), (\alpha_1=1)(\alpha_2=0)(\alpha_2=0)(\alpha_3=1),$

$(\alpha_1=0)(\alpha_2=0)(\alpha_3=1), (\alpha_1=1)(\alpha_2=1)(\alpha_2=0)(\alpha_3=0),$

$(\alpha_1=1)(\alpha_2=0)(\alpha_2=0)(\alpha_3=0), (\alpha_1=0)(\alpha_2=0)(\alpha_3=0) \} =$

$\{ (\alpha_1=1)(\alpha_2=1), 0, (\alpha_1=0)(\alpha_2=1), 0, (\alpha_1=1)(\alpha_2=0)(\alpha_3=1),$

$(\alpha_1=0)(\alpha_2=0)(\alpha_3=1), 0, (\alpha_1=1)(\alpha_2=0)(\alpha_3=0),$

$(\alpha_1=0)(\alpha_2=0)(\alpha_3=0) \} =$

$\{ (\alpha_1=1)(\alpha_2=1), (\alpha_1=0)(\alpha_2=1), (\alpha_1=1)(\alpha_2=0)(\alpha_3=1),$

$(\alpha_1=0)(\alpha_2=0)(\alpha_3=1), (\alpha_1=1)(\alpha_2=0)(\alpha_3=0),$

$(\alpha_1=0)(\alpha_2=0)(\alpha_3=0)$ }.

$|W(T)| = |\mathcal{R}(T,\alpha)| = 6.$

Множество $\mathcal{R}(T,\alpha)$ определяет все условия реализации терма T. Здесь во всех конъюнктивных выражениях знак конъюнкции опущен. Имеем множество условий реализаций и соответственно множество вариантов выполнения терма:

$\mathcal{R}_1 = (\alpha_1=1)(\alpha_2=1)$, $\quad w_1(T, \mathcal{R}_1) = \Phi_1$ СН Φ_6;

$\mathcal{R}_2 = (\alpha_1=0)(\alpha_2=1)$, $\quad w_2(T, \mathcal{R}_2) = \Phi_3$ СН Φ_6;

$\mathcal{R}_3 = (\alpha_1=1)(\alpha_2=0)(\alpha_3=1)$, $w_3(T, \mathcal{R}_3) = \Phi_2$ СН Φ_4;

$\mathcal{R}_4 = (\alpha_1=0)(\alpha_2=0)(\alpha_3=1)$, $w_4(T, \mathcal{R}_4) = \Phi_3$ СН Φ_4;

$\mathcal{R}_5 = (\alpha_1=1)(\alpha_2=0)(\alpha_3=0)$, $w_5(T, \mathcal{R}_5) = \Phi_2$ СН Φ_5,

$\mathcal{R}_6 = (\alpha_1=0)(\alpha_2=0)(\alpha_3=0)$, $w_6(T, \mathcal{R}_6) = \Phi_3$ СН Φ_5.

2.5.3 Построение временной последовательности для одного варианта

Утверждение 4. Каждому варианту $w_i(T, \mathcal{R}_i)$ выполнения терма $T=T(\Phi,\alpha)$ можно поставить в соответствие последовательность моментов времени $t_1, \ldots, t_m$ включения элементов функционального базиса Φ_i.

Доказательство проводится построением этой последовательности. На первом проходе по варианту терма по имеющейся информации о длительностях выполнения элементов Φ_i и по правилам интерпретации операций { СН, СК, $\rightarrow$} на временной оси формируется (вычисляется) длительность выполнения всего варианта. На втором проходе, приняв момент включения терма за нуль, формируется смещение подтермов, входящих в выполняемый вариант, относительно друг друга. Так как относительное смещение формируется для всех подтермов, следовательно, будет вычислено смещение и для подтермов вида $T=\Phi_1$. Это и означает определение времени включения элементов Φ_i в относительной для данного варианта шкале.

Построение временной последовательности для базовых элементов, принадлежащих одному варианту, можно рассматривать как построение функции $F_3(w_i(T, \mathcal{R}_j))$: $\Phi \to t$ для всех $\Phi_i \in w_i(T, \mathcal{R}_j)$, которая каждому элементу функционального базиса ставит в соответствие время его включения. Из нее нетрудно построить обратную ей функцию $F_3^{-1} : t \to \Phi$, которая каждому моменту времени ставит в соответствие включаемые функциональные элементы и образует требуемую временную последовательность.

Алгоритм построения временной последовательности.

п1.Выделение из терма одного варианта $w_i(T, \mathcal{R}_j)$ путем его параметризации по заданному условию $\mathcal{R}_0 \cup \mathcal{R}_j$.

п2.Привязка функциональных элементов к собственной временной шкале. Построение функции:

$$F_3(w_i(T, \mathcal{R}_j))\ \ \Phi \to t_{вкл},$$

где $\Phi = \{\Phi_1, ..., \Phi_k\}$ - функциональные элементы из варианта $w_i(T, \mathcal{R}_j)$ терма T.

п3.Повторение п1, п2 по всем $i = 1, ..., |W(T, \mathcal{R})|$.

Алгоритм выделения одного варианта $w_i(T, \mathcal{R}_j)$.

п1.1.По заданному условию реализации $\mathcal{R}_0 \cup \mathcal{R}_j$ из терма $T=T(\Phi, \Omega)$ выписываем подтерм, для которого истинным является это условие

Привязка функциональных элементов к собственной временной шкале.

п2.1.Определение длительностей и относительных смещений для всех подтермов $w_i(T, \mathcal{R}_j)$, входящих в терм Начало всего терма принимается за нуль.

п2.2.Вычисление моментов включения всех подтермов относительно единого начала. В частности, если подтермом является функциональный элемент, то вычисляется момент его включения.

2.5.4 Приведение терма к единой временной оси

У т в е р ж д е н и е 5. Совокупность временных последовательностей, сформированных для всех вариантов термов при наличии опорного терма в каждом варианте может быть приведена к единой последовательности.

Приведем множество всех вариантов W(T) терма T к единой временной оси. При этом будем предполагать, что существует по крайней мере одна функциональная задача $\Phi_{оп}$, встречающаяся в каждом варианте w_i множества W(T). По времени включения $\Phi_{оп}$ проводится синхронизация всех вариантов целевой задачи.

Спроецируем на временную ось моменты включения функциональных задач для всех вариантов, взяв в качестве опорной точки время включения $\Phi_{оп}$. Проведем перерасчет моментов времени включения функциональных элементов для различных вариантов на единой временной шкале. Тогда, в каждый момент времени мы будем знать включаемые функциональные задачи и их условия выполнимости, т.е. { Φ_i, $F_1(\Phi_i)$ }.

Таким образом, после приведения к единой временной оси полное описание целевой задачи можно представить матрицей $M(t_i, w_j)$, каждый элемент которой содержит множества $\{\Phi_i, F_1(\Phi_i)\}$ для t_i-момента времени и w_j-го варианта алгоритма. Для фиксированного варианта w_j и для различных моментов включения t_i некоторые элементы матрицы могут быть пустыми. Для полного описания одного варианта выполнения целевой задачи, исходя из имеющейся матрицы необходимо указать функцию перехода (функцию управления) от момента времени t_i к следующему ближайшему моменту t_j между которыми элементы матрицы пусты. Для полного описания целевой задачи необходимо указать функцию перехода по всем вариантам выполнения целевой задачи.

С л е д с т в и е 1. По термальному описанию $T=T(\Phi,\alpha)$ и правилам интерпретации каждому Φ_i можно указать его время включения, приведенное к единой временной оси.

Пример. Рассмотрим терм

$T=(\alpha_1=1)=>((\alpha_2=1)=>\{[(\alpha_3=1)=>\Phi_1 + (\alpha_3=0)=>\Phi_2]$ СК $\Phi_3\}$ $+(\alpha_2=0)=>\{[(\alpha_3=1)=>\Phi_4 + (\alpha_3=0)=>\Phi_5]$ СН$\Phi_6\}) \to T_0) + (\alpha_1=0)=>$ $(\Phi_8 \to T_0)$,

где θ_{-0} - невыполняемый терм нулевой длительности.

$T_0=\theta_{-0}$ СН Φ_7.

$\alpha=\{\alpha_1, \alpha_2, \alpha_3\}$

Построим функцию выполнимости целевой задачи $L=F_1(\Phi)$:

$\Phi_1 \to (\alpha_1=1)(\alpha_2=1)(\alpha_3=1)$,

$\Phi_2 \to (\alpha_1=1)(\alpha_2=1)(\alpha_3=0)$,

$\Phi_3 \to (\alpha_1=1)(\alpha_2=1)(\alpha_3=*)$,

$\Phi_4 \to (\alpha_1=1)(\alpha_2=0)(\alpha_3=1)$,

$\Phi_5 \to (\alpha_1=1)(\alpha_2=0)(\alpha_3=0)$,

$\Phi_6 \to (\alpha_1=1)(\alpha_2=0)(\alpha_3=н)$,

$\Phi_7 \to (\alpha_1=н)(\alpha_2=н)(\alpha_3=н)$,

$\Phi_8 \to (\alpha_1=0)(\alpha_2=н)(\alpha_3=н)$.

Таблица условий реализации целевой задачи имеет вид (таблица 2.2).

Таблица 2.2

Условия реализации $\mathcal{R}$	Значения логических переменных α_1	α_2	α_3
$\mathcal{R}_1$	1	1	1
$\mathcal{R}_2$	1	1	0
$\mathcal{R}_3$	1	0	1
$\mathcal{R}_4$	1	0	0
$\mathcal{R}_5$	0	н	н

Проведем параметризацию целевой задачи в логическом пространстве, выпишем все варианты ее выполнения :

$w_1(T, \mathcal{R}_1)= (\Phi_1 \text{ СК } \Phi_3) \to \theta_{оп} \text{ СН } \Phi_7$,

$w_2(T, \mathcal{R}_2)= (\Phi_2 \text{ СК } \Phi_3) \to \theta_{оп} \text{ СН } \Phi_7$,

$w_3(T, \mathcal{R}_3)= (\Phi_4 \to \Phi_6) \to \theta_{оп} \text{ СН } \Phi_7$,

$w_4(T, \mathcal{R}_4)= (\Phi_5 \to \Phi_6) \to \theta_{оп} \text{ СН } \Phi_7$,

$w_5(T, \mathcal{R}_5)=\Phi_8 \to \theta_{оп} \text{ СН } \Phi_7$.

Таким образом, терму Т соответствует множество вариантов $W(T)=\{w_1, w_2, w_3, w_4, w_5\}$.

Зададим длительности функциональных задач (таблица 2.3):

Таблица 2.3

Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	Φ_7	Φ_8	$t_{оп}$

10	50	30	60	40	20	70	80	0

Приведем к единой временной оси (рис.2.2.):

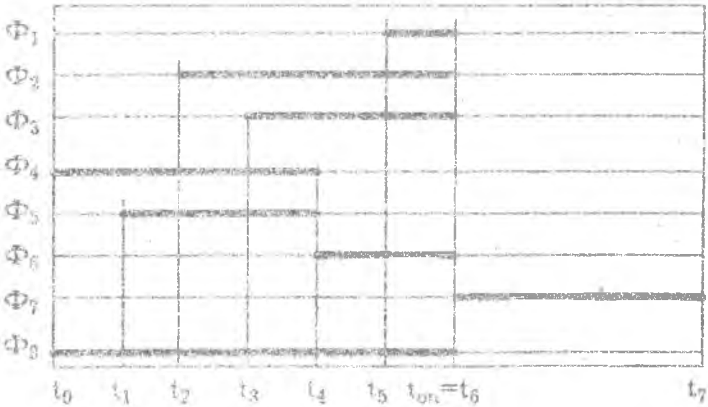

Рис.2.2. Циклограмма целевой задачи

Построим матрицу $M(t_i,w_j)$ вариантов выполнения целевой задачи, упорядоченную по времени выполнения, для t_i, $i=0,...,6$, w_j, $j=1,...,5$ (таблица 2.4).

Таблица 2.4

Время выпол.	Множество вариантов				
	w_1	w_2	w_3	w_4	w_5
t_0			$Ф_4,F_1(Ф_4)$		$Ф_8,F_1(Ф_8)$
t_1				$Ф_5,F_1(Ф_5)$	
t_2		$Ф_2,F_1(Ф_2)$			
t_3	$Ф_3,F_1(Ф_3)$	$Ф_3,F_1(Ф_3)$			
t_4			$Ф_6,F_1(Ф_6)$	$Ф_6,F_1(Ф_6)$	
t_5	$Ф_1,F_1(Ф_1)$				
t_6	$Ф_7,F_1(Ф_7)$	$Ф_7,F_1(Ф_7)$	$Ф_7,F_1(Ф_7)$	$Ф_7,F_1(Ф_7)$	$Ф_7,F_1(Ф_7)$

Как видно из этой матрицы, вариант w_1 целевой задачи реализуется в моменты времени t_3, t_5, t_6.

2.6 Определение протокола выполнения терма

Введем понятие состояния терма. Состояние терма - это состояние входящих в него элементов функционального базиса. Так как их работа интерпретируется во времени, в логическом и информационном пространствах, то и их состояние будем определять в этих же пространствах. Определим состояние элемента функционального базиса $S(Ф_i)$ в j-й момент времени следующей тройкой:

$S(Ф_i)(t_j) = < Ф_i, L_i, I_i >$, где

а) для включения элемента

$Ф_i$- имя элемента;

L_i- условие включения элемента $Ф_i$;

I_i- входные данные элемента $Ф_i$;

б) для отключения элемента

$Ф_i$- имя элемента;

L_i- условие отключения элемента $Ф_i$;

I_i- выходные данные элемента $Ф_i$.

Отметим, что в состоянии $S(Ф_i)$ все параметры относятся к одному элементу функционального базиса.

Будем называть определенную выше тройку $S(Ф_i)$ состоянием элемента $Ф_i$ в момент времени t_j. Совокупность таких наборов для фиксированного момента времени t_j и различных элементов $Ф_i$ назовем состоянием терма в момент t_j.

Упорядоченную по времени последовательность состояний терма в моменты t_j, $j=1,...,m$ назовем протоколом выполнения терма.

Утверждение 6. Всякая матрица $M(t_i, w_j)$ вариантов выполнения целевой задачи реализует протокол выполнения терма.

Доказательство вытекает из способа построения матрицы вариантов. При этом каждая строка матрицы определяет состояние терма в фиксированный момент времени. Предварительно необходимо каждому функциональному элементу Φ_i указать его информационный вектор I_i.

Каждому терму $T = T(\Phi, \alpha)$ поставим в соответствие два состояния $S(вх,T)$ и $S(вых,T)$, для его включения и отключения. Процесс выполнения терма по некоторому варианту, определяемому условием реализации $\mathcal{R}_j(T, \alpha)$, есть процесс смены его состояний из $S(вх,T)$ в $S(вых,T)$, т.е. переход

$S(вх,T) \rightarrow S(вых,T) =$

$\{S(вх,T)(t_0), S(T)(t_1), \dots S(T)(t_i), \dots, S(T)(t_k) = S(вых,T)\}$.

Здесь t_0 - момент запуска терма, t_k - момент окончания его работы. Множество состояний и моменты времени определяются заданным функциональным базисом, в частности длительностью выполнения базовых элементов, и структурой терма. Под этими состояниями будем понимать состояния его базовых элементов в соответствующие моменты времени $\{t_0, t_1, \dots, t_k\}$ их включения и отключения. Всякий терм включается в работу при текущем состоянии системы $S_{тек}$. Под текущим состоянием системы будем понимать

$$S_{тек}(t_i) = \langle \mathcal{P}, \mathcal{L}, \mathcal{I} \rangle,$$

где $\mathcal{P}$ - функциональный базис системы; $\mathcal{L}$ - логический вектор системы. Он указывает значения инициализированных логических переменных, определенных к текущему моменту.

Например, $\mathcal{L}$ = <(α_1=0),(α_2=1)>. Данный логический вектор может означать, например, что элемент $Ф_1$ не включен, элемент $Ф_2$ включен, а другие логические переменные непроинициализированы. $\mathcal{J}$ - текущий вектор информационных переменных системы. Имеется общее поле памяти, в котором каждая проинициализированная переменная задана своим именем и значением. Условиями включения базового элемента $Ф_i$ являются:

$$L_i \subseteq \mathcal{L}$$

- включение условия выполнимости элемента $Ф_i$ в логический вектор $\mathcal{L}$ текущего состояния системы, либо истинность условия выполнимости $L_i \equiv 1$;

$$I_i \subseteq \mathcal{J}$$

- принадлежность входных данных этого элемента текущему вектору $\mathcal{J}$ информационных переменных системы.

При отключении элемента его выходные переменные вводятся в информационный вектор системы со своими значениями либо их значения изменяются, если они уже были проинициализированы в информационном векторе, т.е. меняется текущее состояние системы: $\mathcal{J} = \mathcal{J}\ [+]\ I_i$, $\mathcal{L} = \mathcal{L}\ [+]\ L_i$.

Здесь I_i - вектор выходных переменных элемента $Ф_i$, L_i - условие отключения элемента $Ф_i$.

Состояния терма соответственно перед его включением и отключением определяются следующим образом.

Для терма вида $T = Ф_i$.

$S(вх, Ф_i)(t_0) = < Ф_i, L_i, I_i > = <Ф_i, 1, InpФ_i >$,

где $L_{вх} \equiv 1$- условие выполнимости элемента $Ф_i$. Для терма заданного вида оно тождественно истинно

$S(вых, Ф_i)(t_1) = < Ф_i, L_i, I_i > = < Ф_i, L_i, OutФ_i >$, где $t_1 = t_0 + \tau(\Phi_i)$, $\tau(\Phi_i)$ - длительность выполнения элемента Φ_i.

L_i - условие выполнимости операции отключения элемента Φ_i.

Выполнение терма или переход $S(вх, \Phi_i) \to S(вых, \Phi_i)$ есть последовательность вида $\{S(вх,\Phi_i)(t_0), S(\Phi_i)(t_1) = S(вых,\Phi_i)\}$

После выполнения элемента Φ_i текущее состояние системы изменится следующим образом:

$$\mathcal{S}_{тек}(t_1) = < \mathcal{F}, \mathcal{L}, \mathcal{J} >;$$

$$\mathcal{J} = \mathcal{J}\,[+]\,I_i = \mathcal{J}\,[+]\,\mathrm{Out}\Phi_i;$$

$$\mathcal{L} = \mathcal{L}\,[+]\,L_i.$$

где

$$L_i = \begin{cases} L_i = 0\text{, если после работы элемента } \Phi_i \text{ не формируется никаких признаков} \\ L_i = (\alpha_1 = 1)\text{, если после работы элемента } \Phi_i \text{ сформирован признак, например, вида } (\alpha_1 = 1) \end{cases}$$

Для терма вида $T = (\alpha=1) \Rightarrow \Phi_i + (\alpha=0) \Rightarrow \Phi_j$:

$S(вх, T)(t_0) = <\Phi_i, (\alpha=1), \mathrm{Inp}(\Phi_i)>, <\Phi_j, (\alpha=0), \mathrm{Inp}(\Phi_j)>$,

где t_0 - время включения терма, $L_i = \alpha^{s1}$,

$$\alpha^{s1} = \begin{cases} 1\text{, если } s1 = 1, \\ 0\text{, если } s1 = 0, \end{cases}$$

$$I = \begin{cases} \mathrm{Inp}(\Phi_i)\text{, если } \alpha^{s1} = 1, \\ \mathrm{Inp}(\Phi_j)\text{ если } \alpha^{s1} = 0, \end{cases}$$

$S(T)(t_1) = < \Phi_i, L_i, I_i > = < \Phi_i, (\beta=1), Out(\Phi_i) >,$

$S(вых, T)(t_2) = < \Phi_j, L_j, I_j > = < \Phi_j, (\chi=0), Out(\Phi_j) >,$

$$t_{вых} = \begin{cases} t_1 = t_0 + \tau(\Phi_i), \text{ если } \beta = 1, \\ t_2 = t_0 + \tau(\Phi_j), \text{ если } \chi = 0. \end{cases}$$

Выполнению динамического терма соответствует переход $S(вх,T) \to S(вых,T) = \{S(вх,T)(t_0), S(T)(t_1), S(T)(t_2) = S(вых,T)\}$.

После выполнения терма T текущее состояние системы изменится следующим образом:

$\mathcal{S}_{тек}(t_1) = < \mathcal{P}, \mathcal{L}, \mathcal{J} >;$

$\mathcal{J} = \mathcal{J} [+] I_i = \mathcal{J} [+] Out\Phi_i$, если $\beta = 1$,

$\mathcal{L} = \mathcal{L} [+] L_i$,

где

$$L_i = \begin{cases} L_i = 0 \text{ , если после работы элемента } \Phi_i \text{ не формируется никаких признаков} \\ L_i = (\beta=0), \text{ где } (\beta=0) \text{ признак завершения работы элемента } \Phi_i; \end{cases}$$

$\mathcal{S}_{тек}(t_2) = < \mathcal{P}, \mathcal{L}, \mathcal{J} >;$

$\mathcal{J} = \mathcal{J} [+] I_j = \mathcal{J} [+] Out\Phi_j$, если $\chi = 0$,

$\mathcal{L} = \mathcal{L} [+] L_j$,

где

$$L_j = \begin{cases} L_j = 0 \text{ , если после работы элемента } \Phi_j \text{ не формируется никаких признаков} \\ L_j = (\chi=1), \text{ где } (\chi=1) \text{ признак завершения работы элемента } \Phi_j; \end{cases}$$

$$J = \begin{cases} J\,[+]\,Out(\Phi_i), \text{ если } \beta = 1, \\ \quad\setminus \\ J\,[+]\,Out(\Phi_j), \text{ если } \chi = 0. \end{cases}$$

Для терма вида $T = \Phi_i$ СН Φ_j:

$S(вх, T)(t_0) = \{<\Phi_i, 1, Inp(\Phi_i)>, <\Phi_j, 1, Inp(\Phi_j)>\}$,

$S(T)(t_1) = \{<\Phi_i, L_i, Out(\Phi_i)>\}$,

$S(вых, T)(t_2) = \{<\Phi_j; L_j, Out(\Phi_j)>\}$.

Здесь $t_1 = t_0 + \tau(\Phi_i)$, $t_2 = t_0 + \tau(\Phi_j)$, и для определенности $\tau(\Phi_i) < \tau(\Phi_j)$;

$S(вх,T) \rightarrow S(вых,T) = \{S(вх,T)(t_0), S(T)(t_1), S(T)(t_2) = S(вых,T)\}$.

Для терма вида $T = \Phi_i$ СК Φ_j:

$S(вх, T)(t_0) = \{<\Phi_i, 1, Inp(\Phi_i)>\}$,

$S(T)(t_1) = \{<\Phi_j, 1, Inp(\Phi_j)>\}$,

$S(вых, T)(t_2) = \{<\Phi_j, L_j, Out(\Phi_j)>, <\Phi_i, L_i, Out(\Phi_i)>\}$.

Здесь $t_1 = t_0 + \tau(\Phi_i) - \tau(\Phi_j)$, $t_2 = t_0 + \tau(\Phi_i)$, и для определенности $\tau(\Phi_i) > \tau(\Phi_j)$;

$S(вх,T) \rightarrow S(вых,T) = \{S(вх,T)(t_0), S(T)(t_1), S(T)(t_2) = S(вых,T)\}$.

Для терма вида $T = \Phi_i \rightarrow \Phi_j$:

$S(вх, T)(t_0) = \{<\Phi_i, 1, Inp(\Phi_i)>\}$,

$S(T)(t_1) = \{<\Phi_i, L_i, Out(\Phi_i)>, <\Phi_j, 1, Inp(\Phi_j)>\}$,

$S(вых, T)(t_2) = \{<\Phi_j, L_j, Out(\Phi_j)>\}$.

Здесь $t_1 = t_0 + \tau(\Phi_i)$, $t_2 = t_0 + \tau(\Phi_i) + \tau(\Phi_j)$;

$S(вх,T) \rightarrow S(вых,T) = \{S(вх,T)(t_0), S(T)(t_1), S(T)(t_2) = S(вых,T)\}$.

После выполнения терма текущее состояние системы будет изменяться по правилу

$$S_{тек} = S_{тек}\ [+]\ S(вых,T),$$

где операция текущего изменения состояния системы [+] определяется на множествах интерпретации следующим образом:

$t_{тек} = t_{тек}$,

$\mathcal{L} = \mathcal{L}$ [+] L(вых, T),

$\mathcal{J} = \mathcal{J}$ [+] I(вых. T)

Операция объединения [+] в логическом и информационном пространствах определяется следующим образом:

а) если переменная из выходного состояния терма не определена в текущем состоянии системы $\mathcal{S}_{тек}$, то она вводится в это состояние со значением, сформированным в терме;

б) если переменная из выходного состояния терма определена в текущем состоянии системы $\mathcal{S}_{тек}$, то ее значение изменяется в соответствии со значением, сформированным в терме.

Пользуясь введенными определениями и правилами формирования состояний терма T, а также алгоритмами параметризации терма, можно построить матрицу вариантов реализации терма и, тем самым, протокол поведения системы.

Непосредственно проверкой $\mathcal{S}_{тек} \subseteq S_{цел}$ убеждаемся, реализует ли терм T целевую задачу. В случае ложности условия включения необходимо изменить терм T. Для матрицы, изображенной в таблице 2.4, протокол поведения системы будет иметь вид:

$\mathcal{S}_{тек}(t_{0-0})$ - состояние системы перед запуском терма,

$\mathcal{S}_{тек}(t_0) = \{<\Phi_4, L_4(\Phi_4, w_3), I_4>, <\Phi_8, L_8(\Phi_8, w_5), I_8>\}$,

$\mathcal{S}_{тек}(t_1) = \{<\Phi_5, L_5(\Phi_5, w_4), I_5>\}$,

$\mathcal{S}_{тек}(t_2) = \{<\Phi_2, L_2(\Phi_2, w_2), I_2>\}$,

$$S_{\text{рек}}(t_3) = \{<\Phi_3, L_3(\Phi_3, w_1), I_3>,$$

$$<\Phi_3, L_3(\Phi_3, w_2), I_3> \},$$

$$S_{\text{рек}}(t_4) = \{<\Phi_6, L_6(\Phi_6, w_3), I_6>,$$

$$<\Phi_6, L_6(\Phi_6, w_4), I_6> \},$$

$$S_{\text{рек}}(t_5) = \{<\Phi_1, L_1(\Phi_1, w_1), I_1>\},$$

$$S_{\text{рек}}(t_6) = \{<\Phi_7, L_7(\Phi_7, w_1), I_7>,$$

$$<\Phi_7, L_7(\Phi_7, w_2), I_7>,$$

$$<\Phi_7, L_7(\Phi_7, w_3), I_7>,$$

$$<\Phi_7, L_7(\Phi_7, w_4), I_7>,$$

$$<\Phi_7, L_7(\Phi_7, w_5), I_7> \}.$$

3. Разработка функции асинхронного управления

3.1 Основные положения

В предыдущей главе было проведено исследование термального описания целевой задачи. Была осуществлена параметризация терма в логическом пространстве с выделением полного множества вариантов. Для каждого варианта сделана привязка к относительной временной шкале. Все варианты синхронизированы в единой временной шкале и параметризованы по времени. В результате получили множество состояний для различных моментов времени, каждое из которых содержит набор базовых элементов с их условиями выполнимости. Множество таких состояний, упорядоченных по времени, соответствует стандартной схеме системы.

Построено множество вариантов терма, соответствующее множеству реализаций стандартной схемы. При этом предполагается существование некоторого механизма выделения одного варианта из их полного множества. В частности, выбор варианта для состояния терма в i-й момент времени и переход к следующему (i+1)-му моменту. В данной главе в качестве такого механизма строится и исследуется функция асинхронного управления.

3.2 Построение функции управления

Состояние $S(T)(t_i)$ терма $T=T(\Phi)$ в i-й момент времени описывается совокупностью k наборов

$S(T)(t_i) = \{ <\Phi_{j1}, L_{j1}, I_{j1}>,$

$<\Phi_{j2}, L_{j2}, l_{j2}>$,

...

$<\Phi_{jk}, L_{jk}, l_{jk}>$ }, где $k \geq 1$.

Каждому состоянию $S(T)(t_i)$ терма $T=T(\Phi)$ в i-й момент времени поставим в соответствие управляющий вектор вида

$< v(t_i), u(\Phi, L_q), inp(\Phi) >$,

где $\Phi=\{ \Phi_1,...,\Phi_m \}$ - множество функциональных элементов, входящих в данное состояние терма;

$L=\{ L_1, ..., L_k \}$ - множество условий включения и/или отключения элементов Φ, заданных на множестве логических переменных $\alpha = \{\alpha_1, \ldots, \alpha_n\}$.

Предполагается заданной полностью определенная функция F_1: $\Phi \rightarrow \mathcal{B}(n)$, которая каждому элементу Φ_j ставит в соответствие условие его выполнимости. Одно условие L_q может содержать несколько логических переменных. Одна логическая переменная может входить в разные условия L_q как с одинаковыми, так и с разными значениям;

$v(t_i)$ - функция разрешения управления в момент t_i;

$u(\Phi, L_q)$ - функция включения и/или отключения элементов Φ_j, $j=1,...,m$ при истинности условия L_q, $q=1,...,k$ в момент времени t_i. В дальнейшем аргумент времени включения t_i будем опускать, так как весь набор определяется при фиксированном времени $t=t_i$ и функция разрешения $v(t_i)$ фиксирует этот момент,

$inp(\Phi)$ - функция ввода и/или вывода переменных для элементов Φ_j, $j=1,...,m$.

При отключении элемента Φ_j условие L_q есть условие отключения, а функция ввода переменных заменяется функцией вывода.

Функция разрешения $v(t_i)$, определяющая момент проведения управляющего воздействия и смены состояния терма, определена на оси времени и может принимать одно из двух значений: { работа, не работа }. Имеется ввиду "работа" или "не работа" органа управления.

Функцию включения $u(\Phi,L)$ элементов множества Φ при истинности условий из множества L в момент времени t_i определим следующим образом.

Управление в i-й момент распространяется на каждый элемент данного состояния системы. Управляющее воздействие реализует две операции: включение элемента Φ_j, отключение элемента Φ_j. Операция включения элемента предполагает:

-проверку истинности условия включения;

-в случае его истинности передачу исходных данных элементу Φ_j;

-включение элемента Φ_j.

Если условие включения ложно, то элемент не включается. Аналогично для операции отключения элемента. Таким образом, поведение функционального элемента Φ_i для операции включения можно описать двумя состояниями:

{включение, невключение}. Аналогично для операции отключения: { отключение, неотключение}. Эти операции не независимы. Операция включения имеет место для элемента, находящегося в состоянии "отключение", и наоборот.

Условие включения и/или отключения j-го элемента в i-й момент времени может содержать либо не содержать m-ю логическую переменную. Если она содержится в условии, то может принимать одно из двух значений - истина или ложь. Если она не входит в условие, то элемент включается независимо от значения m- й переменой. Таким образом, поведение m-й логической переменной можно описать тремя значениями: н – независимость условия включения от m-й переменой, 1 – включение элемента в i-й момент времени; 0 – невключение элемента. Аналогично для операции отключения. Обозначим область значений m-й переменной E= { н, 1, 0 }. Тогда область значений n переменных есть декартово произведение E^n. В принципе, m-я переменная может быть и k-значной.

Конкретный вид функции включения строится по заданным наборам в состоянии $S(T)(t_i)$ терма.

Функция ввода входных переменных $inp(Ф_i)$ для элемента $Ф_i$ является унарным отношением в пространстве переменных D^n, истинным для входных переменных элемента $Ф_i$ и выполняется как чтение данных из общего поля памяти в случае включения элемента $Ф_i$.

Управляющий вектор для терма, состоящего из одного набора

$$<Ф_1, L_1, 1_i>,$$

в i-й момент времени определяется по следующему правилу:

$v(t) = [\ ; \ ]$

$< v(t_i), u(Ф_1,L_1), inp(Ф_1) >$

Таким образом, для состояния терма в i-й момент времени и состоящего из одного набора, функция включения имеет вид

$$u(Ф_1,L_1) = \begin{cases} вкл(Ф_1), & если\ L_1 = 1, \\ невкл(Ф_1), & в\ противном\ случае; \end{cases}$$

$inp(Ф_1)$ – функция ввода входных переменных для элемента $Ф_1$.

При отключении элемента имеем аналогичный управляющий вектор.

$v(t) = [t_i]$,

$< v(t_i), u(L_1), out(\Phi_1) >$.

Функция отключения имеет вид

$$u(\Phi_1,L_1) = \begin{cases} \text{откл}(\Phi_1), \text{ если } L_1=1, \\ \text{неоткл}(\Phi_1), \text{ в противном случае;} \end{cases}$$

$out(\Phi_1)$ - функция вывода выходных переменных для элемента Φ_1.

Пример7. Пусть $L_1=((\alpha_1=0) \& (\alpha_2=1))$.

$$u(\Phi_1,L_1) \in \{ \text{вкл, невкл} \}.$$

Функцию включения зададим таблицей значений (таблица 3.1):

Таблица 3.1

α_1	α_2	$u(\Phi_1,L_1)$
н	н	невкл
н	0	невкл
н	1	невкл
0	н	невкл
0	0	невкл
0	1	вкл
1	н	невкл
1	0	невкл
1	1	невкл

Пример8. Рассмотрим состояние терма из двух наборов

$< v(t_i), u(\Phi_1,L_1), inp(\Phi_1) >$,

$< v(t_i), u(\Phi_2,L_2), inp(\Phi_2) >$,

и пусть $L_1=((\alpha_1=0)(\alpha_2=1))$, $L_2=(\alpha_1=1)$, $/L/=2$.

Функцию включения зададим таблицей значений (таблица 3.2):

Таблица 3.2

α_1	α_2	$u(\Phi_1, L_1)$	$u(\Phi_2, L_2)$
н	н	невкл	невкл
н	0	невкл	невкл
н	1	невкл	невкл
0	н	невкл	невкл
0	0	невкл	невкл
0	1	вкл	невкл
1	н	невкл	вкл
1	0	невкл	вкл
1	1	невкл	вкл

Пример9.

Управляющий вектор для состояния из четырех наборов $<\Phi_1, L_1, I_1>$, $<\Phi_2, L_2, I_2>$, $<\Phi_3, L_3, I_3>$, $<\Phi_4, L_4, I_4>$ и $L_1=(\alpha_1=0)(\alpha_2=1)$. $L_2= L_1$, $L_3=(\alpha_2=0)$, $L_4=(\alpha_1=н)(\alpha_2=н)$

определим следующим образом:

$v(t) = [t_1]$.

Функция включения для четырех наборов имеет вид (таблэца 3.3):

Таблица 3.3

α_1	α_2	$u(\Phi_1, L_1)$	$u(\Phi_2, L_2)$	$u(\Phi_3, L_3)$	$u(\Phi_4, L_4)$
н	н	невкл	невкл	невкл	вкл
н	0	невкл	невкл	вкл	вкл
н	1	невкл	невкл	невкл	вкл
0	н	невкл	невкл	невкл	вкл
0	0	невкл	невкл	вкл	вкл

0	1	вкл	вкл	невкл	вкл
1	н	невкл	невкл	невкл	вкл
1	0	невкл	невкл	вкл	вкл
1	1	невкл	невкл	невкл	вкл

Пусть в i- й момент состояние терма $T=T(\Phi,L)$ содержит m функциональных элементов, а условия их выполнимости в совокупности имеют k логических переменных. Тогда функцию включения в i-й момент времени определим как отображение вида

$$u(\Phi_1,...,\Phi_m,L): E^k \to \mathcal{B}(m),$$

где E^k - множество всех логических выражений длины k,

$E^k=\{(e_1,..., e_i,..., e_k)\}$, $e_i \in \{$ н,0,1$\}$ для $\forall i=1,...,k$;

$\mathcal{B}(m)$ - булеан на множестве m функциональных элементов. Для m=2 $\mathcal{B}(2)=\{\varnothing, \Phi_1, \Phi_2, \{\Phi_1,\Phi_2\}\}$ Для функции включения будем его интерпретировать в базисе {вкл, невкл} следующим образом: $\mathcal{B}(2)$ ={(невклΦ_1, невклΦ_2), (вклΦ_1, невклΦ_2),(невклΦ_1, вклΦ_2), (вклΦ_1, вклΦ_2)}. В асинхронные моменты времени t_i, i=1,2,... вектор $e=(e_1,...,e_i,...,e_k)$ может принимать одно из значений на E^k. Условие включения элемента Φ_i также есть вектор длины k $L_i=(\alpha_1,...,\alpha_i,...,\alpha_k)$. Элемент Φ_i с условием L_i включается в t_i -й момент времени, если $(\alpha_1,...,\alpha_i,...,\alpha_k) \subseteq (e_1,...,e_i,...,e_k)$, т.е. $\alpha_i \subseteq e_i$ для всех i=1,...,k. В базисе {н,0,1} отношение включения определяется следующим образом: $1 \subseteq 1$, $0 \subseteq 0$, н $\subseteq 1$, н $\subseteq 0$.

Алгоритм построения отображения $u(\Phi_1,...,\Phi_m,L)$: $E^k \to \mathcal{B}(m)$.

п1. Для элемента $e \in E^k$ выделяем подмножество базовых элементов $\{\Phi_i\}$ с условиями выполнимости, удовлетворяющих включению $L(\Phi_i) \subseteq e$. В частности, такое подмножество может быть и пустым. Пара $<e_i, \{\Phi_j\}>$ образует одно значение отображения.

п2. Повторить п1 для всех элементов множества E^k. Для всех элементов Φ_j, не удовлетворяющих отношению включения, образуется одна пара $<\varnothing, \{\Phi_j\}>$, означающая невключение элементов $\{\Phi_j\}$.

Функцию ввода входных переменных $inp(\Phi_1,\dots,\Phi_m)$ определим как мультипликативную функцию вида

$$inp(\Phi_1,\dots,\Phi_m) = inp(\Phi_1)*\dots*inp(\Phi_m),$$

где одноместная функция $inp(\Phi_j)$ выполняется только при включении элемента Φ_j. Свойство мультипликативности функции ввода распространяется и на m элементов. Мультипликативность функции ввода вытекает из ее определения: чтение подмножества переменных для элементов $\Phi_1, \dots, \Phi_m$ равносильно чтению переменных для элементов $\Phi_1, \dots, \Phi_m$.

Аналогично определяется функция управления для операции отключения.

Пусть i-е состояние терма $T=T(\Phi)$ характеризуется m_1 наборами с включением элементов и m_2 наборами с отключением элементов. Такому состоянию терма поставим в соответствие управляющий вектор:

$v(t) = [\ t_i\ ]$,

$< v(t_i), u(\Phi,L), inp(\Phi) >$, $\Phi = \Phi_1 \cup \Phi_2$, $L = L_1 \cup L_2$,

$\Phi_1 = \{ \Phi_{11},\dots,\Phi_{1m1} \}$, $\Phi_2 = \{ \Phi_{21},\dots,\Phi_{2m2} \}$,

$L_1 = \{ L_{11},\dots,L_{1k1} \}$, $L_2 = \{ L_{21},\dots,L_{2k2} \}$

Функция включения и/или отключения будет иметь вид:

$u(\Phi,L) = E^{(k1+k2)} \to \mathcal{J}(m_1+m_2)$, где (k_1+k_2) - мощность объединения множеств логических переменных по включению и отключению элементов;

(m_1+m_2)- мощность объединения множеств функциональных элементов.

3.3 Построение множества управлений

Каждому терму Т=Т(Ф) из множества термов целевых задач поставим в соответствие функцию управления вида:

$< v(T)(t), \ u(T)(t), inp(T)(t)>$.

Функцию разрешения управления v(T)(t) для терма Т=Т(Ф), заданную на том же временном отрезке $[t_{нач}, t_{кон}]$, что и терм Т=Т(Ф), определим как кусочно-постоянную функцию, принимающую два значения: { работа, не работа}. Значение "работа" определим в моменты времени, соответствующие смене состояний терма Т, а "не работа" - во все остальные моменты на заданном временном отрезке:

$$v(T)(t) = \begin{cases} 1, \text{ если } t=t_i, i=0,1,...,k \\ 0, \text{ в противном случае.} \end{cases} \qquad (3.1)$$

Здесь $v(T)(t_i) = 1$ означает состояние "работа" в моменты времени $t_0,t_1,...,t_k$. Напомним, что между соседними моментами включений t_i и t_i+1, i=0,...,k-1 интервалы времени могут быть разной длительности.

Неформально функция v(T)(t) определяет моменты времени подачи команд управления. Исходя из определения функцию разрешения v(T)(t) можно задать асинхронной последовательностью моментов времени $(t_0,t_1,...,t_k)$, соответствующей смене состояний терма Т=Т(Ф).

Функция $u(T)(t)$ производит включение и/или отключение функциональных элементов, содержащихся в терме $T=T(\Phi,L)$, в моменты, соответствующие смене состояний. Ее можно задать последовательностью функций включения $u(\Phi_j,L_j)$ для различных моментов времени t_i, $i=1,\dots,k$.

Неформально функция $u(T)(t_i)$ в t_i-й момент для функционального элемента Φ_j выполняет упорядоченный набор действий: проверку логических условий для элемента Φ_j; в случае их истинности, чтение и инициализацию данных для элемента Φ_j, включение элемента Φ_j. Если в t_i-й момент времени включается несколько функциональных элементов, то перечисленные действия выполняются для всех элементов.

Функция $inp(T)(t)$ производит чтение и/или запись данных в моменты, соответствующие смене состояний терма.

Множество всех управлений есть множество всех троек вида $< v(T)(t), u(T)(t), inp(T)(t)>$. Выделим на нем подмножество управлений, привязанных к единой временной шкале. Введем на нем операцию сложения по различным термам следующим образом.

$$< v(T_1)(t_1), u(T_1)(t_1), inp(T_1)(t_1) > (+)$$

$$< v(T_2)(t_2), u(T_2)(t_2), inp(T_2)(t_2) > =$$

$$< v(T)(t), u(T)(t), inp(T)(t)>,$$

где область определения функции суммы есть

$$t_{нач}= \min(t_{1нач}, t_{2нач}), t_{кон}= \max(t_{1кон}, t_{2кон}).$$

$$v(T)(t) = \begin{cases} 1, \text{ если } t=t_i,\ i=0,1,\dots,k \text{ или} \\ \qquad\qquad t=t_j,\ j=0,1,\dots,m \\ 0, \text{ в противном случае} \end{cases} \tag{3.2}$$

$$u(T)(t) = u(T_1)(t) (+) u(T_2)(t) \tag{3.3}$$

$$inp(T)(t)= inp(T_1)(t_1)\ (+)\ inp(T_2)(t_2). \qquad (3.4)$$

Выражение (3.2) есть функция разрешения для двух термов. В моменты времени t_i осуществляется управление элементами терма T_1; в моменты t_j - терма T_2; если $t_i=t_j$, то разрешается управление как элементами терма T_1, так и T_2.

Функция включения (3.3) в момент t_i осуществляет включение элементов терма T_1; в момент t_j - терма T_2; если $t_i=t_j$, то включаются элементы как терма T_1, так и терма T_2. Выражение для такой функции включения можно записать следующим образом:

$$u(T)(t) : E^{(n1+n2)} \rightarrow \mathcal{B}(k_1+k_2),$$

где $n_1(n_2)$ - число логических переменных в терме $T_1(T_2)$;
$k_1(k_2)$ - число функциональных элементов с условиями включения в терме T_1 (T_2).

Функция ввода переменных (3.4) в моменты t_i читает элементы терма T_1; в моменты t_j - терма T_2; в моменты $t_i=t_j$ читает входные данные как терма T_1, так и терма T_2

Сумма функций управления есть новая функция из этого множества. Неформально сумма двух функций есть функция управления двумя термами T_1 и T_2 с помощью одного управляющего органа.

Введем управляющий вектор для невыполняемого терма λ длительности τ. Это терм, в котором есть пустой функциональный элемент длительности τ. Этому элементу по умолчанию приписано тождественно ложное условие его включения, т.е. функция включения элемента его безусловно не включает. Так как он имеет ненулевую длительность, то устанавливается временая задержка на время τ. Определим его функцию управления

$$<v(\lambda)(t), u(\lambda)(t), inp(\lambda)(t)>$$

следующим образом: функция разрешения

$v(\lambda)(t) = \{t_{нач}, t_{кон}\}$, где $t_{кон} = t_{нач} + \tau$;

$u(\lambda)(t) = E^n \to \mathcal{A}(m)$, где $n=0$ - число логических переменных; $m=0$ - число функциональных элементов в терме. Функция вырождается в одну точку определения и с одним значением "невкл", так как элемент пустой

Функция чтения $inp(\lambda)(t) = \{ \varnothing \}$ также не выполняется, так как подмножество переменных, подлежащих чтению, пусто. Операция сложения с пустым термом дает

$$< v(T_1)(t_1), u(T_1)(t_1), inp(T_1)(t_1) > \quad (+)$$

$$< v(\lambda)(t_2), u(\lambda)(t_2), inp(\lambda)(t_2)> =$$

$$< v(T)(t), u(T)(t), inp(T)(t) >,$$

$t_{нач} = \min(t_{1нач}, t_{2нач}), \quad t_{кон} = \max(t_{1кон}, t_{2кон})$.

Функция разрешения $v(T)(t)$ продолжена до области $(t_{нач}, t_{кон})$.

$$u(T)(t) = u(T_1)(t_1) \ (+) \ u(\lambda)(t_2),$$

где функция включения терма T расширена на временной оси новыми моментами времени, в которые она принимает значение "невкл".

$$inp(T)(t_1) = inp(T_1)(t_1) \ (+) \ inp(\lambda)(t_2).$$

Введем управляющий вектор для невыполняемого терма θ нулевой длительности. Это терм, в котором есть пустой функциональный элемент нулевой длительности. Этому элементу по умолчанию приписано тождественно ложное условие его включения, т.е. функция включения элемента его безусловно не включает. Определим

$< v(\theta)(t),\ u(\theta)(t), inp(\theta)(t)>$

следующим образом: область определения функции разрешения $v(\theta)(t)$ задана одним значением $t_{нач}=t_{кон}$

$u(\theta)(t)= E^n \to \mathscr{B}(m)$, где n=0 -число логических переменных; m=0 - число функциональных элементов в терме. Функция вырождается в одну точку определения и с одним значением "откл" , так как элемент пустой.

Функция чтения $inp(\theta)(t) =\{ \varnothing \}$ также не выполняется, так как подмножество переменных, подлежащих чтению, пусто.

Аналогично определяется операция сложения:

$< v(T_1)(t_1), u(T_1)(t_1), inp(T_1)(t_1) >\ (+)$

$< v(\theta)(t),\ u(\theta)(t), inp(\theta)(t)> \ =$

$< v(T_1)(t_1), u(T_1)(t_1), inp(T_1)(t_1) >$.

Очевидно, что на множестве управлений, заданных на одном и том же временном отрезке, операция сложения обладает свойством идемпотентности, коммутативности и ассоциативности.

Из определения множества управлений вытекают следующие стратегии управления: единое управление несколькими термами; независимое и одновременное (параллельное) управление каждым термом. В дальнейшем речь будет идти об управлении только одним термом.

Рассмотрим два способа задания функции управления: последовательное (событийное) и термальное управление.

3.4 Последовательное управление

Последовательное или событийное управление есть управление по времени. Терму ставится в соответствие функция управления, действующая на том же временном отрезке. Предварительно терм преобразуется в допустимую стандартную схему, в которой однозначно определены события включения и/или отключения элементов. В эти моменты функция управления производит соответствующие действия.

Задача ставится следующим образом: по заданному терму $T=T(\Phi,L)$ построить его функцию управления, т.е. определить протокол выполнения терма: определить последовательность моментов времени включения алгоритма управления и для каждого момента t_i, $i=0,1,\ldots,k$ построить функцию включения и/или отключения элементов функционального базиса, функцию ввода / вывода данных.

Обозначим $< v(T)(t), u(T)(t), inp(T)(t) >$ - функция управления термом $T=T(\Phi,L)$, где $\Phi=\{\Phi_0,\Phi_1,\ldots,\Phi_k\}$ - элементы функционального базиса, $L=\{\alpha_1,\ldots,\alpha_n\}$ - вектор логических переменных.

На основании утверждения2 каждому элементу функционального базиса Φ_i можно поставить в соответствие его полный логический вектор $L_i=\{\alpha_{i1},\ldots,\alpha_{im}\}$, $L_i=F_1(\Phi_i)$. Таким образом пара $(\Phi_i,F_1(\Phi_i))$ полностью определяет включаемость элемента Φ_i.

Состояние $S(T)(t_i)$ терма $T=T(\Phi)$ в i-й момент времени описывается совокупностью k наборов:

$S(T)(t_i) =\{ <\Phi_{j1}, L_{j1}, I_{i1}>,$

$<\Phi_{j2}, L_{j2}, I_{i2}>,$

$<\Phi_{jk}, L_{jk}, I_{jk}> \}$, где $k \geq 1$.

Каждому набору $<\Phi_j, L_j, I_j>$ этого состояния поставим в соответствие управляющий вектор:

$$< v(t_i), u(\Phi_j,L_j), inp(\Phi_j) >.$$

На основании утверждения 4 каждой реализации терма $T=T(\Phi,L)$ ставится в соответствие временная последовательность $t=F_2(\Phi)$. Этим самым определена функция $v(R(T))(t)$ для одной реализации терма. Суммируя их по всем реализациям, получим первую компоненту функции управления $v(T)(t)=\{t_0,...,t_k\}$.

Для каждой реализации терма построим $u(R(T))(t)= F_2^{-1}(t)$. Суммируя их по всем реализациям, получим вторую компоненту функции управления $u(T)(t)$. Отметим, что функция $u(T)(t)$ определена полностью. Моменты времени определяются функцией $t=F_2(\Phi)$. Функция $F_2^{-1}(t)= \Phi$ определяет включаемые функциональные элементы в соответетствующие моменты времени. Функция $L=F_1(\Phi)$ каждому элементу указывает его полный логический вектор. Входной и выходной информационые векторы для элемента Φ_i заданы по определению. Таким образом, для терма $T=T(\Phi,L)$ полностью определена функция управления

$$< v(T)(t), u(T)(t), inp(T)(t) >.$$

Правила интерпретации команд управления.

$U(\Phi_i)$: $Int(I,\Phi_i)$: $\Phi_i \rightarrow Input\ \Phi_i$,

$Int(L,\Phi_i)$: $\Phi_i \rightarrow L$

$Int(t,\Phi_i)$: $\Phi_i \rightarrow$ ВКЛ(Φ_i) в момент $t_{вкл}=t$

$U(\Phi_i$ CH $\Phi_j)$: $Int(I,(\Phi_i$ CH $\Phi_j))$: $\{\Phi_i \rightarrow Input\ \Phi_i, \Phi_j \rightarrow input\ \Phi_j\}$,

$Int(L,(\Phi_i$ CH $\Phi_j))$: $\{\Phi_i \rightarrow L, \Phi_j \rightarrow L\}$,

$Int(t,(\Phi_i$ CH $\Phi_j))$: $\{\Phi_i \rightarrow$ ВКЛ(Φ_i) в момент $t_{вкл}=t$,

$\Phi_j \to$ ВКЛ(Φ_j) в момент $t_{вкл}=t\}$.

$U(\Phi_i$ СК $\Phi_j)$: Int$(I,(\Phi_i$ СК $\Phi_j))$: $\{\Phi_i \to$ Input Φ_i, $\Phi_j \to$ Input $\Phi_j\}$,

Int$(L,(\Phi_i$ СК $\Phi_j))$: $\{\Phi_i \to L$, $\Phi_j \to L\}$,

Int$(t,(\Phi_i$ СК $\Phi_j))$: $\{\Phi_i \to$ ВКЛ(Φ_i) в момент $t_{вкл}=t$,

$\Phi_j \to$ ВКЛ(Φ_j) в момент $t_{вкл}=t+\tau$,

если $\tau_i > \tau_j\}$, либо

$\{\Phi_i \to$ ВКЛ(Φ_i) в момент $t_{вкл}=t+\tau$,

$\Phi_j \to$ ВКЛ(Φ_j) в момент $t_{вкл}=t$,

если $\tau_i =< \tau_j\}$,

где

$$\tau = \begin{cases} \tau_i - \tau_j, \text{ если } \tau_i > \tau_j, \\ \tau_j - \tau_i, \text{ если } \tau_i =< \tau_j. \end{cases}$$

$U(\Phi_i \to \Phi_j)$: Int$(I,(\Phi_i \to \Phi_j))$: $\{\Phi_i \to$ Input Φ_i, $\Phi_j \to$ Input $\Phi_j\}$,

Int$(L,(\Phi_i \to \Phi_j))$: $\{\Phi_i \to L$, $\Phi_j \to L\}$,

Int$(t,(\Phi_i \to \Phi_j))$: $\{\Phi_i \to$ ВКЛ(Φ_i) в момент $t_{вкл}=t$,

$\Phi_j \to$ ВКЛ(Φ_j) в момент $t_{вкл}=t+\tau_i\}$.

3.5 Термальное управление

Термальное управление есть управление по структуре терма. Каждому подтерму ставится в соответствие функция управления, которая вызывает на выполнение функции управления подтермов нижнего уровня. Для подтерма типа функциональный элемент выполняются команды включения и передачи данных.

Отличительной особенностью такого управления является тот факт, что разные подтермы имеют независимое, параллельное управление.

Всякий терм целевой задачи можно представить в виде дерева бинарных подтермов (п2.3). Бинарный подтерм - это терм, имеющий два операнда и одну операцию. При этом операндами могут быть как составные термы, так и функциональные элементы. Последние выступают в качестве висячих вершин на нижнем уровне такого дерева. Смысл термального управления: функции управления термом ставятся в соответствие функции управления операндами, входящими в терм. Таких функций будет столько, сколько бинарных подтермов в терме целевой задачи. Эти функции образуют дерево, в котором каждая дуга означает передачу управления от терма верхнего уровня к термам нижнего уровня. Каждой висячей вершине также ставится в соответствие функция управления. Если функции управления для подтермов только вызывают друг друга, то функции управления висячих вершин непосредственно управляют базовыми объектами.

Установим соответствие между термом T целевой задачи и его функцией управления U(T) следующим образом.

1. Для терма $T=\Phi_i$ функция управления есть $U(T) = U(\Phi_i)$;
2. Для терма $T=T1 \otimes T2$ функция управления определяется как:

   $U(T) = U(T1 \otimes T2) = U(T1) \otimes U(T2)$,

   где $\otimes \in \{$ СН, СК, $\rightarrow \}$.
3. Для динамического терма $T=(\alpha=1) \Rightarrow T1 + (\alpha=0) \Rightarrow T2$ функция управления имеет вид:

$$U(T) = \begin{cases} U(T1), & \text{если } (\alpha=1) = \text{TRUE} \\ U(T2), & \text{в противном случае} \end{cases}$$

4. Других правил нет.

Любое термальное выражение ЦЗ можно привести к совокупности термов, состоящих из двух операндов и операции. Поэтому при переходе к термальной функции управления необходимо описать переход к функциям управления операндов и операции, входящим в терм ЦЗ.

Опишем правила интерпретации функции управления для операндов терма Т. Так как терм ЦЗ определен в информационном, логическом и временном пространствах, то и его функцию управления будем интерпретировать в этих же пространствах.

Под управлением термом Т во времени будем понимать команду включения и/или отключения терма Т в соответствующие моменты времени $\{t_0, t_1\}$, $t_1=t_0+\tau$, где τ - длительность терма Т.

Под управлением термом Т в информационном пространстве будем понимать формирование входных данных перед включением терма Т и запоминание его выходных данных после отработки.

Под управлением термом в логическом пространстве будем понимать проверку его условия выполнимости, при истинности которого производится включение и/или отключение терма Т.

Операции на временной оси, входящие в термальное управление, будем интерпретировать так же, как и операции над функциональными элементами терма ЦЗ, т.е. как смещение на временной оси одной функции управления относительно другой.

Исходя из правил интерпретации функции управления термом в различных пространствах зафиксируем порядок ее выполнения.

Функция включения терма Т представляет собой следующую упорядоченную последовательность операций, проводимых в момент времени $t=t_0$.

1. Формирование вектора значений входных данных Inp(T).
2. Проверка истинности условия выполнимости терма.
3. Включение (невключение) терма при истинности (ложности) условия выполнимости.

Функция отключения терма Т представляет собой следующую упорядоченную последовательность операций, проводимых в момент времени $t_1 = t_0 + \tau$, где τ - длительность терма Т.

1. Проверка истинности условия выполнимости терма.
2. Отключение (неотключение) терма при истинности (ложности) условия выполнимости.
3. Запоминание вектора значений выходных данных Out(T).

Операция включения/отключения терма.

1. Для $T=\Phi_i$ эта операция включает базовый элемент Φ_i,
$U(\Phi_i) = ВКЛ\ (\Phi_i)$.
2. Для $T=T1 \otimes T2$, где $\otimes \in \{$ СН, СК, $\rightarrow\}$, эта операция осуществляет вызов функций управления для подтермов Т1, Т2, не являющихся базовыми элементами (ВЫЗОВ U(T1), ВЫЗОВ U(T2)). Порядок их вызова на временной оси определяется правилами интерпретации операций над термами.
3. Для $T=(\alpha=1)\Rightarrow T1+(\alpha=0)\Rightarrow T2$ эта операция осуществляет альтернативный вызов функции управления подтермами Т1,Т2 (ВЫЗОВ U(T1), ВЫЗОВ U(T2)), если они не являются базовыми элементами.

Пример3. Рассмотрим терм вида

$T = T1 \rightarrow T2,$

где $T1 = Ф_1$ СН $Ф_2$, $T2 = Ф_3$ СК $Ф_4$. Ему соответствуют функции управления:

$U(T) = U(T1 \rightarrow T2) = U(T1) \rightarrow U(T2);$

$U(T1) = U(Ф_1)$ СН $U(Ф_2)$; $U(T2) = U(Ф_3)$ СК $U(Ф_4)$.

Правила интерпретации функции управления для операций терма T опишем в следующих пунктах отдельно для каждой операции. Опишем функцию термального управления для различных подтермов и базовых элементов.

3.6 Управление функциональной задачей

Под базовым будем понимать управление функциональной задачей $Ф_i$.

Пусть $T=Ф_i$, функция разрешения $v(T)$ терма T есть $v(T) = v(Ф_i) = \{t_0, t_1\}$. Для терма $T=Ф_i$, $F_1(Ф_i)=L_i$ и ($L_i \equiv$ и), т.е. условие выполнимости L_i элемента $Ф_i$ тождественно истинно. Запись ($L_i \equiv$ и) означает, что включение элемента $Ф_i$ независит ни от одной логической переменной, входящей в условие выполнимости L_i. А это означает безусловное включение элемента $Ф_i$. С учетом правил интерпретация функциональной задачи в информационном, логическом и временном пространствах будем и команды управления формировать в этих же пространствах.

При формировании функции управления будем исходить из предположения, введенного в п1.1, что длительность команд управления практически равна нулю на фоне работы элементов системы.

Функционирование элемента $Ф_i$ определяется циклограммой, изображенной на рис.3.1. Процесс смены состояний осуществляется в моменты времени t_0, t_1. В момент t_0 происходит включение $Ф_i$ и в момент t_1 - отключение.

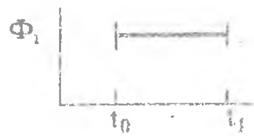

Рис.3.1 Циклограмма функциональной задачи $Ф_i$

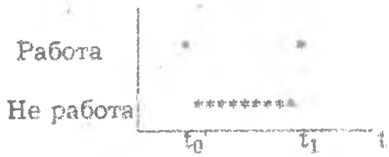

Рис.3.2 Состояние процессора при управлении $Ф_i$

Команды управления будут выполняться процессором в моменты t_0, t_1. В силу предположения об их нулевой длительности, в интервале времени $\tau = t_1 - t_0$ процессор будет свободен от управления этим термом. На рис. 3.2 показано состояние процессора при управлении функциональным элементом $Ф_i$.

Для терма $T = Ф_i$ управляющий вектор запишется в следующем виде:

$v(Ф_i)(t) = \{t_0, t_1\}$,

$<v(Ф_i), u(Ф_i), Inp(Ф_i)>(t_0)$,

$<v(Ф_i), u(Ф_i), Out(Ф_i)>(t_1)$,

где $v(Ф_i)(t) = \{t_0, t_1\}$ - последовательность моментов времени работы органа управления (функция разрешения управления термом $T=Ф_i$); t_0 - включение $Ф_i$, $t_1 = t_0 + \tau_1$ - отключение $Ф_i$.

Для момента $t = t_0$ управляющий вектор будет иметь вид:

$Inp(Ф_i)$ - формирование вектора значений входных данных;

$u(Ф_i)$ = ВКЛ ($Ф_i$), если $t = t_0$.

Для момента отключения элемента $Ф_i$ при $t_1 = t_0+\tau_i$, управляющий вектор имеет вид:

$u(Ф_i)$ = ОТКЛ ($Ф_i$), если $t = t_1$;

$Out(Ф_i)$ - запоминание вектора значений выходных данных.

3.6.1 Интерпретация на временной оси

Исходя из вышеизложенного, имеем

INTERPR (t):

$v(Ф_i)(t) = \{t_0, t_1\}$;

$u(Ф_i)(t_0) = \{$ вкл$(Ф_i)(t_0)$, невкл$(Ф_i)(t_0)$ $\}$,

$u(Ф_i)(t_1) = \{$ откл$(Ф_i)(t_1)$, неоткл$(Ф_i)(t_1)$ $\}$.

На временной оси функция разрешения $v(Ф_i)(t)$ интерпретируется как функция включения алгоритма управления в соответствующие моменты времени. Для функции управления $u(Ф_i)(t)$ указывается ее область допустимых значений в соответствующие моменты времени.

3.6.2 Интерпретация в логическом пространстве

INTERPR (α):

Проверка $(\alpha(Ф_i) = 0) = \{$ TRUE, FALSE $\}$,

где $\alpha(Ф_i)$ - признак состояния функциональной задачи $Ф_i$;

$\alpha(Ф_i) = 0$ - невключенное состояние;

$\alpha(Ф_i) = 1$ - включенное состояние.

3.6.3 Интерпретация в информационном пространстве

INTERPR (I): {Формирование вектора значений входных данных $Inp(Ф_i)$} $\cup$ {Запоминание вектора значений выходных данных $Out(Ф_i)$}. Область значений функции управления в информационном пространстве разнесена во времени. Если формирование входных данных происходит при включении терма, то запоминание выходных данных – при его отключении.

3.6.4 Общая схема управления

Функция включения:

$v(\Phi_i) = t_0$,

$Inp(\Phi_i)$, (3.5)

$\{$ВКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\bmod 2\}$, если $(\alpha(\Phi_i)=0)$ = TRUE.

Функция отключения:

$v(\Phi_i) = t_1$,

$\{$ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\bmod 2\}$, если $(\alpha(\Phi_i)=1)$ =TRUE

$OUT(\Phi_i)$. (3.6)

В выражениях (3.5), (3.6) в фигурных скобках дается перечисление выполняемых операций управления при всех интерпретациях. Предполагается, что признак состояния $\alpha(\Phi_i)$ инициирован к моменту управления функциональной задачей Φ_i и изменяется одновременно с выполнением операций {ВКЛ, ОТКЛ}.

Управление составными термами выразим через базовое управление.

3.7 Управление термом вида $T=T_i$ CH T_j

На рис. 3.3 изображена циклограмма терма.

Рис. 3.3 Циклограмма терма с операцией CH

$U(T) = U(T_i\ СН\ T_j) = U(T_i)\ СН\ U(T_j)$.

Выполнению терма $T= T_i$ СН T_j под управлением функции $U(T)$ соответствуют процессы $\{t_0,t_1\}$, $\{t_0,t_2\}$ для подтермов T_i и T_j соответственно, происходящие под управлением функций $U(T_i)$, $U(T_j)$. Для операции "совпадение по началу" эти процессы начинаются одновременно в момент t_0, а заканчиваются в моменты t_1, t_2, соответствующие длительностям термов.

Если T_i, T_j - составные термы, то для них функция управления формируется точно так же, как и для терма T. Таким образом, в область значений функции управления $U(T)$ термом T входит и вызов функций управления подтермами T_i, T_j терма T.

Функция управления составным термом с операцией "совпадение по началу (СН)" имеет вид

$U(T)$ = ВЫЗОВ $U(T_i)$, ВЫЗОВ $U(T_j)$ при $t=t_0$. (3.7)

Функционированию составного терма T вида $T=Ф_i$ СН $Ф_j$ соответствует функция управления

$U(T)$ = ВЫЗОВ $U(Ф_i)$, ВЫЗОВ $U(Ф_j)$ при $t=t_0$,

t_0 - время включения терма T.

Выполнению такого терма соответствуют процессы $\{t_0,t_1\}$, $\{t_0,t_2\}$, происходящие под управлением функций $U(Ф_i)$, $U(Ф_j)$. В момент t_0 происходит одновременное включение термов $Ф_i$ и $Ф_j$, в моменты t_1, t_2 их соответствующее отключение. Естественно, что команды управления включением $Ф_i$ и $Ф_j$ будут выполняться последовательно друг за другом, но в силу предположения о бесконечно малой их длительности будем считать, что они включаются одновременно в момент времени t_0. Аналогично команды отключения элементов $Ф_i$ и $Ф_j$ выполняются в моменты t_1, t_2 соответственно с нулевой длительностью. Тогда состояние процессора можно описать во времени следующей функцией (рис.3.4).

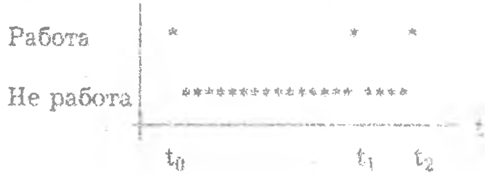

Рис.3.4 Состояние процессора для операции СН

Так как операндами терма $Т= Ф_i$ СН $Ф_j$ являются базовые элементы, то для них функция управления определена в п. 3.5.

Для терма $Т= Ф_i$ СН $Ф_j$ поведение функции управления в t_0-й момент времени будет иметь вид:

ВЫЗОВ $U(Ф_i)$ при $t=t_0$,

ВЫЗОВ $U(Ф_j)$ при $t=t_0$.

Для терма $Т=Ф_i$, согласно п.3.5, функция управления имеет вид: при $t=t_0$

$inp(Ф_i)$ - формирование вектора значений входных данных для терма $Т=Ф_i$;

$U(Ф_i)$ - включение базового элемента $Ф_i$ (ВКЛ $Ф_i$).

Для момента $t=t_1$ управляющий вектор будет иметь вид:

$U(Ф_i)$ - отключение элемента $Ф_i$ (ОТКЛ $Ф_i$);

$Out(Ф_i)$ - запоминание вектора значений выходных данных операнда $Ф_i$.

Аналогично записывается функция управления для терма $Т=Ф_j$.

3.7.1 Интерпретация на временной оси

Исходя из вышеизложенного, имеем

INTERPR (t):

$v(\Phi_i)(t) = \{t_0, t_1\}$,

$v(\Phi_j)(t) = \{t_0, t_2\}$,

$U(\Phi_i)(t_0) = \{$ ВКЛ$(\Phi_i)(t_0)$, НЕВКЛ$(\Phi_i)(t_0)$ $\}$,

$U(\Phi_j)(t_0) = \{$ ВКЛ$(\Phi_j)(t_0)$, НЕВКЛ$(\Phi_j)(t_0)$ $\}$,

$U(\Phi_i)(t_1) = \{$ ОТКЛ$(\Phi_i)(t_1)$, НЕОТКЛ$(\Phi_i)(t_1)$ $\}$,

$U(\Phi_j)(t_2) = \{$ ОТКЛ$(\Phi_j)(t_2)$, НЕОТКЛ$(\Phi_j)(t_2)$ $\}$.

3.7.2 Интерпретация в логическом пространстве

INTERPR (α):

1. Проверка ($\alpha(T) = 0$) = TRUE,

где $\alpha(T)$ - признак состояния терма T. Представляет собой вектор состояний входящих в терм базовых элементов

$\alpha(T) = (\alpha(\Phi_i), \alpha(\Phi_j))$.

2. Проверка условия совместимости элементов Φ_i и Φ_j, если оно задано.

$$\alpha_{совм} = \begin{cases} 1, & \text{если допустима совместная работа элементов } \Phi_i \text{ и } \Phi_j; \\ 0, & \text{в противном случае.} \end{cases}$$

3.7.3 Интерпретация в информационном пространстве

INTERPR (I):{Формирование вектора значений входных данных Inp(T)} $\cup$ {Запоминание вектора значений выходных данных Out(T)}. Вектор значений входных данных формируется как объединение входных данных базовых элементов, входящих в терм. Аналогично определяется вектор значений выходных данных терма T.

3.7.4 Общая схема управления

Для момента $t = t_0$:

$v(T) = t_0$,

ВЫЗОВ $U(\Phi_i)$ при $t=t_0$ - начало процесса $\{t_0, t_1\}$,

ВЫЗОВ $U(\Phi_j)$ при $t=t_0$ - начало процесса $\{t_0, t_2\}$ (3.7)

В выражениях (3.7) происходит одновременное включение двух процессов, соответетствующих термам $T= \Phi_i$, $T= \Phi_j$.

3.8 Управление термом вида $T=T_i$ СК T_j

Функция управления составным термом с операцией "совпадение по концу (СК)" имеет вид:

$$U(T) = U(T_i \text{ СК } T_j) = U(T_i) \text{ СК } U(T_j).$$

$$U(T)= \begin{cases} \text{ВЫЗОВ } U(T_i), \text{ если } \tau_i >= \tau_j, \text{ при } t=t_0, \\ \text{ВЫЗОВ } U(T_j), \text{ если } \tau_i < \tau_j, \text{ при } t=t_0; \\ \\ \text{ВЫЗОВ } U(T_j), \text{ если } \tau_i >= \tau_j, \text{ при } t=t_0+ \tau_i-\tau_j; \\ \text{ВЫЗОВ } U(T_i), \text{ если } \tau_i < \tau_j, \text{ при } t=t_0+|\tau_i - \tau_j|. \end{cases} \quad (3.8)$$

На рис.3.5 изображена циклограмма терма.

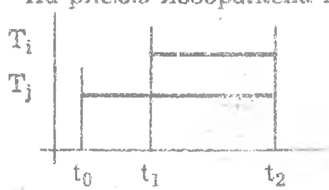

Рис.3.5 Циклограмма операции СК

Данная запись означает, что для операции СК - совпадение по концу, функция управления U(T) выполняется в различные моменты времени $\{t_0, t_1\}$. Порядок вызова функций управления подтермами зависит от длительностей их выполнении. В записи (3.8) описаны все возможные ситуации. Если T_i, T_j - составные термы, то для них функция управления формируется точно так же, как и для терма T.

Функционированию составного терма T вида $T=Ф_i$ СК $Ф_j$ соответствуют процессы $\{t_1, t_2\}$, $\{t_0, t_2\}$, происходящие под действием функций управления $U(Ф_i)$, $U(Ф_j)$. В момент t_0 происходит включение $Ф_j$, если длительность его выполнения больше длительности $Ф_i$ ($\tau_i \le \tau_j$), тогда в момент t_1 - включение $Ф_i$, либо наоборот, а в момент t_2 их соответствующее отключение. Состояние процессора во времени описывается функцией, изображенной на рис.3.6.

Функционированию составного терма T вида $T=Ф_i$ СК $Ф_j$ соответствует функция управления:

$$U(T)=\begin{cases} \text{ВЫЗОВ } U(Ф_i), \text{ если } \tau_i >= \tau_j, \text{ при } t=t_0, \\ \text{ВЫЗОВ } U(Ф_j), \text{ если } \tau_i < \tau_j, \text{ при } t=t_0; \\ \\ \text{ВЫЗОВ } U(Ф_j), \text{ если } \tau_i >= \tau_j, \text{ при } t=t_0+\tau_i-\tau_j; \\ \text{ВЫЗОВ } U(Ф_i), \text{ если } \tau_i < \tau_j, \text{ при } t=t_0+|\tau_i-\tau_j|. \end{cases} \quad (3.9)$$

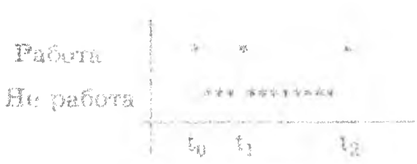

Рис.3.6 Состояние процессора для операции СК

Так как операндами терма $T = Ф_i$ СК $Ф_j$ являются базовые элементы, то для них функция управлёния определена в п. 3.5. Функция управления операцией "совпадения по концу" выражает смещение по времени функции управления первого операнда относительно второго.

Для терма $T = Ф_i$ СК $Ф_j$, изображенного на рис.3.5, поведение функции управления в t_0-й момент времени будет иметь вид:

inp(T) - формирование вектора значений входных данных для операнда $Ф_j$ терма T;

$U(Ф_j)$ = ВКЛ $Ф_j$ - включение базового элемента $Ф_j$.

Для момента $t_1 = t_0 + |\tau_i - \tau_j|$ управляющий вектор для элемента $Ф_i$ определяется аналогично

3.8.1 Интерпретация на временной оси

Исходя из вышеизложенного, имеем

INTERPR (t):

$v(T)(t) = \{t_1, t_2\}$,

$v(T)(t) = \{t_0, t_2\}$,

$U(Ф_j)(t_0)$= { ВКЛ$(Ф_j)(t_0)$, НЕВКЛ$(Ф_j)(t_0)$ },

$U(Ф_i)(t_1)$= { ВКЛ$(Ф_i)(t_1)$, НЕВКЛ$(Ф_i)(t_1)$ },

$U(Ф_i)(t_2)$= { ОТКЛ$(Ф_i)(t_2)$, НЕОТКЛ$(Ф_i)(t_2)$ },

$U(Ф_j)(t_2)$= { ОТКЛ$(Ф_j)(t_2)$, НЕОТКЛ$(Ф_j)(t_2)$ }.

3.8.2 Интерпретация в логическом пространстве

INTERPR (α):

1. Проверка ($\alpha(T) = 0$) = { TRUE, FALSE }, где

$\alpha(T)$ - признак состояния терма T. Представляет собой вектор состояний входящих в терм базовых элементов

$\alpha(T) = \{\alpha(\Phi_i), \alpha(\Phi_j)\}$

2. Проверка условия совместимости элементов Φ_i и Φ_j, если оно задано.

$$\alpha_{совм} = \begin{cases} 1, \text{ если допустима совместная работа} \\ \quad \text{элементов } \Phi_i \text{ и } \Phi_j; \\ 0, \text{ в противном случае.} \end{cases}$$

3.8.3 Интерпретация в информационном пространстве

INTERPR (T){Формирование вектора значений входных данных Inp(T)} ∪ {Запоминание вектора значений выходных данных Out(T)} Вектор значений входных данных формируется в моменты t_0, t_1 соответственно для базовых элементов Φ_j, Φ_i, входящих в терм. Вектор значений выходных данных терма T формируется как объединение данных базовых элементов Φ_j, Φ_i.

3.8.4 Общая схема управления

Для момента $t = t_0$:

$v(T) = t_0$,

ВЫЗОВ $U(\Phi_j)$ при $t = t_0$ - начало процесса $\{t_0, t_2\}$.

Для момента $t = t_1$:

$v(T) = t_1$,

ВЫЗОВ $U(\Phi_i)$ при $t = t_1$ - начало процесса $\{t_1, t_2\}$

3.9 Управление термом вида $T = T_i \rightarrow T_j$

По аналогии с п.3.2.1 для терма $T = T_i \rightarrow T_j$ его циклограмма (рис.3.7), состояние процессора (рис.3.8) и функция управления будут иметь вид:

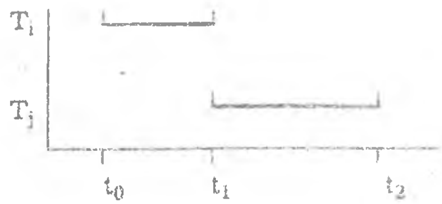

Рис.3.7 Циклограмма операции следования

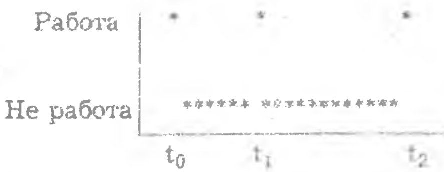

Рис.3.8 Состояние процессора для операции ($\to$)

$$U(T) = U(T1 \to T2) = U(T1) \to U(T2).$$

Выполнению терма $T = T_i \to T_j$ под управлением функции U(T) соответствуют процессы $\{t_0,t_1\}$, $\{t_1,t_2\}$ для подтермов T_i и T_j соответственно, происходящие под управлением функций $U(T_i)$, $U(T_j)$. Для операции "следования" эти процессы начинаются в моменты t_0, t_1, а заканчиваются соответственно в моменты t_1, t_2.

Если T_i, T_j - составные термы, то для них функция управления формируется точно так же, как и для терма T.

Функция управления составным термом с операцией "следования ($\to$)" имеет вид:

$$U(T)= \begin{cases} \text{ВЫЗОВ } U(T_i) & \text{при } t=t_0, \\ \text{ВЫЗОВ } U(T_j) & \text{при } t=t_0+t_i; \end{cases} \qquad (3.10)$$

t_0 - время включения терма T. Для операции $\to$ - следование время включения терма T_j сдвинуто на время t_i относительно времени включения терма T_i.

Функционированию составного терма T вида $T=Ф_i \rightarrow Ф_j$ соответствуют процессы вида $\{t_0, t_1\}$, $\{t_1, t_2\}$. В момент t_0 происходит включение $Ф_i$, в момент t_1 - отключение $Ф_i$ и включение $Ф_j$, в момент t_2 отключение $Ф_j$. Состояние процессора описывается во времени функцией, изображенной на рис.3.8.

Для терма $T= Ф_i \rightarrow Ф_j$ функция управления имеет вид $U(Ф_i \rightarrow Ф_j) = U(Ф_i) \rightarrow U(Ф_j)$. Так как операндами терма являются базовые элементы, то для них функция управления определена в п. 3.5. Функция управления операцией "следование" выражает смещение по времени функции управления второго операнда относительно первого.

Для терма $T= Ф_i \rightarrow Ф_j$, изображенного на рис.3.7, функция управления имеет вид:

$v(T)= t_0$,

ВЫЗОВ $U(Ф_i)$ - начало процесса $\{t_0, t_1\}$.

$v(T)= t_1$.

ВЫЗОВ $U(Ф_j)$ - начало процесса $\{t_1, t_2\}$.

Для процесса $\{t_0, t_1\}$ функция управления $U(Ф_i)$ имеет вид:

$v(Ф_i)= t_0$,

$Inp(Ф_i)$ - формирование вектора значений входных данных для операнда $Ф_i$ терма T,.

$U(Ф_i)=$ ВКЛ $Ф_i$ - включение базового элемента $Ф_i$.

$v(Ф_i)= t_1$.

$Out(Ф_i)$ - запоминание вектора значений выходных данных для операнда $Ф_i$ терма T:

$U(Ф_i)=$ ОТКЛ $Ф_i$ - отключение базового элемента $Ф_i$;

Аналогично определяется функция управления $U(Ф_j)$ для процесса $\{t_1, t_2\}$.

3.9.1 Интерпретация на временной оси

Исходя из вышеизложенного, имеем

INTERPR (t):

$v(T)(t) = \{t_0, t_1, t_2\}$,

$U(Ф1)(t_0)= \{$ ВКЛ$(Ф1)(t_0)$, НЕВКЛ$(Ф1)(t_0)$ $\}$,

$U(Ф1)(t_1)= \{$ ОТКЛ$(Ф1)(t_1)$, НЕОТКЛ$(Ф1)(t_1)$ $\}$,

$U(Ф2)(t_1)= \{$ ВКЛ$(Ф2)(t_1)$, НЕВКЛ$(Ф2)(t_1)$ $\}$,

$U(Ф2)(t_2)= \{$ ОТКЛ$(Ф2)(t_2)$, НЕОТКЛ$(Ф2)(t_2)$ $\}$.

3.9.2 Интерпретация в логическом пространстве

INTERPR (α):

1. Проверка ($\alpha(T)=0$) = { TRUE, FALSE }, где

$\alpha(T)$ - признак состояния терма Т. Представляет собой вектор состояний входящих в терм базовых элементов

$\alpha(T)= (\alpha(Ф1), \alpha(Ф2))$.

2. Проверка условия совместимости элементов Ф1 и Ф2, если оно задано.

$$\alpha_{совм} = \begin{cases} 1, \text{ если допустима совместная работа} \\ \quad \text{элементов Ф1 и Ф2;} \\ 0, \text{ в противном случае.} \end{cases}$$

3.9.3 Интерпретация в информационном пространстве

INTERPR (I):{Формирование вектора значений входных данных Inp(T)} $\cup$ {Запоминание вектора значений выходных данных Out(T)}. Вектор значений входных данных формируется в моменты t_0, t_1 соответственно для базовых элементов Ф1, Ф2, входящих в терм. Вектор значений выходных данных терма Т формируется как объединение данных базовых элементов Ф1, Ф2.

3.9.4 Общая схема управления

Для момента $t = t_0$

ВЫЗОВ $U(Ф_i)$.

Для момента $t = t_1$

ВЫЗОВ $U(Ф_j)$.

3.10 Управление динамическим термом

Рассмотрим динамический терм $T = (\alpha=1) \Rightarrow T_i + (\alpha=0) \Rightarrow T_j$. По аналогии с п.2.2.1 для терма T его циклограммы (рис.3.9), состояния процессов (рис.3.10) и функции управления будут иметь вид.

$U(T) = U(T_i \rightarrow T_j) \cdot U(T_i) \rightarrow U(T_j)$.

Выполнению динамического терма T под управлением функции U(T) соответствуют процессы $\{t_0, t_1\}$, $\{t_0, t_2\}$ для подтермов T_i и T_j соответственно, происходящие под управлением функций $U(T_i)$, $U(T_j)$. Подтермы T_i и T_j выполняются альтернативно, в зависимости от истинности логического выражения, наведенного на терм. Для операции "выбора (+)" эти процессы начинаются в момент t_0, а заканчиваются соответственно в моменты t_1, t_2.

а) при $\alpha=1$ б) при $\alpha=0$

Рис.3.9 Циклограммы динамического терма

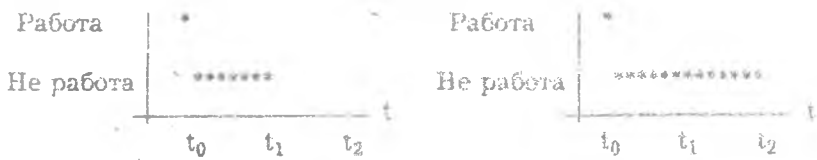

Рис.3.10 Состояние процессора для динамического терма

Если T_i, T_j - составные термы, то для них функция управления формируется точно так же, как и для терма T.

Функционированию составного терма T вида $T=(\alpha=1)\Rightarrow T_i+(\alpha=0)\Rightarrow T_j$ соответствуют альтернативно два процесса, происходящие в моменты $\{t_0, t_1\}$ и $\{t_0, t_2\}$. В момент t_0 происходит выполнение первого процесса $\{t_0, t_1\}$, если истинно условие выполнимости терма $T=T_i$ $(\alpha=1)$, либо второго процесса $\{t_0, t_2\}$, если истинно условие выполнимости терма $T=T_j$ $(\alpha=0)$. Состояние процессора будет изменяться в соответствии с поведением одной из функций, изображенных на рис.3.10.

Функция управления динамическим термом с операцией "выбора (+)" имеет вид:

$$U(T)=\begin{cases}\text{ВЫЗОВ } U(T_i), & \text{если } (\alpha=1), \quad \text{при } t=t_0,\\ \text{ВЫЗОВ } U(T_j), & \text{если } (\alpha=0), \quad \text{при } t=t_0;\end{cases} \qquad (3.11)$$

t_0 - время включения терма T. Для операции + - выбора время включения терма T_j совпадает со временем включения терма T_i.

Для терма $T=(\alpha=1)\Rightarrow\Phi_i+(\alpha=0)\Rightarrow\Phi_j$ операндами являются базовые элементы, и для них функция управления определена в п. 3.5. Функция управления операцией "выбора" выражает выбор одного из двух альтернативных процессов и его выполнение.

Для терма $T=(\alpha=1)\Rightarrow\Phi_i+ (\alpha=0)\Rightarrow\Phi_j$, изображенного на рис.3.9, поведение функции управления в t_0-й момент времени будет следующим: если ($\alpha=1$), то выполняется процесс $v(T)=\{t_0, t_1\}$, в противном случае выполняется процесс $v(T)=\{t_0, t_2\}$.

Для первого процесса функция управления имеет вид.

В t_0-й момент времени:

Inp(T) - формирование вектора значений входных данных для операнда Φ_i терма T;

$U(\Phi_i)$= ВКЛ Φ_i - включение базового элемента Φ_i.

Для момента $t_1 = t_0 + \tau_i$.

$Out(\Phi_i)$ - запоминание вектора значений выходных данных для операнда Φ_i терма T;

$U(\Phi_i)$= ОТКЛ Φ_i - отключение базового элемента Φ_i.

Для второго процесса функция управления аналогична.

3.10.1 Интерпретация на временной оси

Исходя из вышеизложенного, имеем

INTERPR (t):

$v(T)(t) = \{t_0, t_1\}$, если ($\alpha=1$),

$\{t_0, t_2\}$, если ($\alpha=0$);

$U(\Phi_i)(t_0)$= { ВКЛ$(\Phi_i)(t_0)$, НЕВКЛ$(\Phi_i)(t_0)$ },

$U(\Phi_i(t_1)$= { ОТКЛ$(\Phi_i)(t_1)$, НЕОТКЛ$(\Phi_i)(t_1)$ },

$U(\Phi_j)(t_0)$= { ВКЛ$(\Phi_j)(t_0)$, НЕВКЛ$(\Phi_j)(t_0)$ },

$U(\Phi_j)(t_2)$= { ОТКЛ$(\Phi_j)(t_2)$, НЕОТКЛ$(\Phi_j)(t_2)$ }.

3.10.2 Интерпретация в логическом пространстве

INTERPR (α):

1. Выбор истинного условия выполнимости:

если ($\alpha=1$)=TRUE, то реализуется первый процесс $v(T)(t) = \{t_0, t_1\}$, или выполняется терм $T=\Phi_i$, в противном случае реализуется второй процесс $v(T)(t) = \{t_0, t_2\}$, или выполняется терм $T=Ф_j$.

2. Проверка $(\alpha(T) = 0) = \{$ TRUE, FALSE $\}$, где

$\alpha(T)$ - признак состояния терма T. Представляет собой вектор состояний входящих в терм базовых элементов

$\alpha(T) = \alpha(Ф_i)$, если $(\alpha=1)$, либо $\alpha(Ф_j)$ в противном случае.

3.10.3 Интерпретация в информационном пространстве

INTERPR (I):{Формирование вектора значений входных данных Inp(T)} $\cup$ {Запоминание вектора значений выходных данных Out(T)}. Вектор значений входных данных формируется в моменты t_0, t_1 соответственно для базовых элементов $Ф_i$, $Ф_j$, входящих в терм. Вектор значений выходных данных терма T формируется как объединение данных базовых элементов $Ф_i$, $Ф_j$.

3.10.4 Общая схема управления

$$U(T) = \begin{cases} \text{ВЫЗОВ } U(Ф_i), \text{ если } (\alpha=1), & \text{при } t=t_0, \\ \text{ВЫЗОВ } U(Ф_j), \text{ если } (\alpha=0), & \text{при } t=t_0. \end{cases}$$

В выражении (3.11) дается перечисление выполняемых операций управления для различных моментов времени и для всех интерпретаций. При термальном управлении на выполнение вызывается только один подтерм, для которого истинным является его условие выполнимости. В свою очередь управление подтермом сводится к одному из рассмотренных выше случаев.

3.11 Построение функции управления составным термом

Правило построения функции управления составным термом определим индуктивно, где индукция берется по длине терма.

Для терма вида $T=\Phi_i$ функция управления определяется с помощью выражения (3.5).

Для статического терма вида $T=T_i \otimes T_j$, где $\otimes$ - символ операции на временной оси, $\otimes \in \{$ СН, СК, $\rightarrow\}$, функция управлении определяется с помощью выражений (3.6) - (3.10).

Если T_i, T_j - составные термы, то для них функция управления формируется точно так же, как и для терма T. Для динамического терма вида $T=(\alpha=1)\Rightarrow\Phi_i + (\alpha=0\Rightarrow\Phi_j$ функция управления определяется по выражениям (3.11).

4 Разработка алгоритмов асинхронного управления

4.1 Основные положения

В предыдущей главе для целевой задачи, заданной термом $T=T(\Phi,\alpha)$, построена функция управления. Управление сводится к реализации целевой задачи по одному из вариантов, в зависимости от состояния системы в логическом пространстве. В разделе рассматриваются два способа управления: последовательный и термальный.

Последовательное (событийное) управление представлено: функцией выделения момента времени t_i (функцией разрешения $v(t_i)$), функцией управления $u(t_i)$ элементами функционального базиса в этот момент времени и функцией чтения или записи данных.

Термальное управление существенно опирается на структуру терма и задается согласованной совокупностью простейших функций управления двух типов - управление составным термом и управление функциональным элементом.

В данной главе для заданной функции строится соответствующий алгоритм асинхронного управления. При создании таких алгоритмов желательно знать архитектуру компьютера, на котором они будут выполняться. Для неймановской однопроцессорной машины естественным является последовательное управление. При наличии на ней мультипрограммного режима возможна реализация термального управления. В многопроцессорных системах естественным является термальное управление. Последовательное управление также возможно с применением прави распараллеливания алгоритма. Реализацию алгоритмов асинхронного управления рассмотрим на классической однопроцессорной машине.

4.2 Неймановская модель языков программирования

В качестве модели алгоритма управления взята классическая неймановская модель языка программирования высокого уровня. Фиксируются конечные множества символов предикатов L, символов программ U и множество команд σ. Обозначим σ~- множество всевозможных цепочек символов из σ. На множестве σ~ определена операция конкатенации (*) таким образом, что если $<x_1, x_2, ..., x_k> \in \sigma^{\sim}$ и $< y_1, y_2, ..., y_l> \in \sigma^{\sim}$, то

$<x_1,x_2,...,x_k> * <y_1,y_2,...,y_l> = < x_1, x_2, ..., x_k, y_1, y_2, ..., y_l> \in \sigma^{\sim}$. Алгебраическую систему $A =< \sigma^{\sim}, * >$, где множество цепочек $\sigma^{\sim}$ построено из символов множества σ, назовем моделью множества программ.

Расширим множество σ командой УВИ(τ) - установка временного интервала на время τ; $\sigma^{\wedge} = \sigma \cup$ УВИ. Интерпретация команды УВИ(τ): после ее выполнения программа передает управление процессору и будет вновь включена через время τ, указанное в операнде команды. При этом управление передается команде, следующей за командой УВИ(τ). Тогда алгебраическую систему $A^{\wedge}=<\sigma^{\sim},*>$ в базисе $\sigma^{\wedge}$ назовем моделью множества алгоритмов асинхронного управления. В данной системе всякая последовательность $<x_1, x_2, ..., x_k>$ не содержащая команды УВИ(τ), есть обычная (одномоментная) программа. Если последовательность имеет вид $<x_1$, УВИ(τ), $x_2>$, то команда x_2 будет выполняться через время

τ после включения команды x_1. Последовательность $<x_1$, УВИ(τ_1), $\{x_2,x_3\}>$ будет выполняться следующим образом: через время τ после включения команды x_1 будет выполняться блок команд $\{x_2, x_3\}$. В последовательности

$$<x_1, \text{УВИ}(\tau_1), \text{УВИ}(\tau_2), x_2, x_3>$$

после включения команды x_1 через время τ_1 будет включена команда x_2, а через время τ_2 после включения команды x_1 будет включена команда x_3. В отличие от предыдущей, в последовательности $<x_1$, УВИ(τ_1), x_2, УВИ(τ_2), $x_3>$ команда x_3 будет включена через время τ_2 после включения команды x_2.

4.3. Базовые команды внутреннего языка описания алгоритмов асинхронного управления

Определим систему команд внутреннего языка формирования алгоритмов управления.

На временной оси: команда включения (отключения) функционального элемента $Ф_i$ ВКЛ($Ф_i$), (ОТКЛ($Ф_i$)).

В логическом пространстве команды условного перехода:

ЕСЛИ < условие >

ТО цепочка команд1

ИНАЧЕ цепочка команд2 ;

ЕСЛИ < условие >

ТО цепочка команд ;

Цепочка команд может состоять из одной команды.

В информационном пространстве: команды чтения (записи) данных из общего поля памяти ЧТЕНИЕ ($Ф_i$), (ЗАПИСЬ ($Ф_i$)).

Освобождение процессора от выполняемой программы с последующим захватом процессора этой же программой через время τ будем задавать командой установки временного интервала (УВИ(τ)).

Для пустого элемента λ длительности τ:

ВКЛ(λ) - невыполняемая команда в смысле включения терма. После передачи управления такой команде в тексте алгоритма следует переход к следующей по порядку команде через время τ, равное длительности пустого терма λ.

ОТКЛ(λ) - невыполняемая. Команды ЧТЕНИЕ (λ), ЗАПИСЬ (λ) невыполняемы, т.е. управление переходит к следующей за этой команде.

Для элемента θ нулевой длительности команды ВКЛ(θ), ОТКЛ(θ), ЧТЕНИЕ (θ), ЗАПИСЬ (θ) - невыполняемые. После передачи им управления в тексте алгоритма следует переход к следующей по порядку команде.

На множестве всех цепочек из команд внутреннего языка для фиксированного операнда $Ф_i$ вводятся следующие условия, истинность которых определяет правильность формирования цепочек:

-для всякого элемента $Ф_i$ должны выполняться следующие отношения порядка

ЧТЕНИЕ ($Ф_i$) < ВКЛ ($Ф_i$),

ОТКЛ ($Ф_i$) < ЗАПИСЬ ($Ф_i$);

-для всякого элемента $Ф_i$ число команд ЧТЕНИЕ($Ф_i$) (ЗАПИСЬ($Ф_i$)) равно числу команд ВКЛ($Ф_i$) (ОТКЛ($Ф_i$)).

Это означает, что в тексте алгоритма управления для всякого элемента $Ф_i$ всегда должны выполняться эти отношения.

4.4 Последовательный метод построения асинхронного алгоритма

Каждому управляющему вектору терма поставим в соответствие алгоритм асинхронного управления. В системе команд, помимо стандартных, могут быть и нестандартные макрокоманды, написанные и поименованные пользователем.

Будем формировать алгоритм управления на некотором внутреннем символическом языке макрокоманд. Каждому функциональному элементу $Ф_i$ к моменту его включения ставится в соответствие стандартный набор команд управления: формирование данных, чтение данных из общего поля памяти, проверка логических признаков, включение функционального элемента в работу в случае истинности логических признаков. По завершении работы управляющий алгоритм выполняет стандартный набор команд:

отключение функционального элемента, если это не происходит автоматически;

формирование логических признаков;

запись выходных данных в общее поле памяти.

Сформулируем это правило.

п1. Формирование текста алгоритма управления для фиксированного момента времени по заданному управляющему вектору.

п2. Повторение пункта п1 требуемое число раз по всем моментам времени.

п3. Формирование переходов между последовательными асинхронными событиями с освобождением процессора на время $t = t_i - t_{i-1}$ и возвратом к следующей команде по тексту алгоритма в реальном времени.

4.4.1 Формирование текста алгоритма управления

Для состояния терма в i-й момент времени, состоящего из одного набора, управляющий вектор имеет вид

$< v(t_i), u(\Phi_j, L_j, t_i), inp(\Phi_j)>$.

Функция разрешения $v(t_i)$ определяет моменты времени включения алгоритма управления. Функции включения $u(\Phi_j, L_j)$ элемента Φ_j поставим в соответствие команду

ЕСЛИ L_j=TRUE ТО ВКЛ (Φ_j) .

Функции ввода переменных $inp(\Phi_j)$ поставим в соответствие команду ЧТЕНИЕ (Φ_j) .

Текст алгоритма управления для i-го момента времени включения элемента Φ_j есть:

1. ЧТЕНИЕ (Φ_j)

2. ЕСЛИ L_j=TRUE ТО ВКЛ (Φ_j), или этот текст, состоящий из двух операторов, можно переписать в один оператор:

1. ЕСЛИ L_j=TRUE ТО (ЧТЕНИЕ (Φ_j), ВКЛ (Φ_j)) .

При отключении элемента Φ_j строится цепочка из двух команд:

1. ЕСЛИ L_j=TRUE ТО ОТКЛ (Φ_j) .

2. ЗАПИСЬ (Φ_j) , здесь L_j- есть условие отключения элемента Φ_j. Этот текст также можно переписать одним оператором:

1. ЕСЛИ L_j=TRUE ТО (ОТКЛ (Φ_j), ЗАПИСЬ (Φ_j)) .

Если отключение элемента Φ_j производится автоматически, то текст алгоритма будет состоять из одного оператора:

1. ЗАПИСЬ (Φ_j)

Заметим, что автоматическое отключение может быть только безусловным, когда L_{ij}= TRUE.

Для терма в i-й момент времени, состоящего из двух наборов

$<\Phi_1,L_1,I_1>$, $<\Phi_2,L_2,I_2>$ и $L_1=(\alpha_1=0)$, $L_2=(\alpha_2=1)$,

управляющий вектор имеет вид

$< v(t_i), u(\Phi_1,\Phi_2,L_i), inp(\Phi_1,\Phi_2)>(t_i)$.

Функция разрешения $v(t_i)$ определяет моменты времени включения алгоритма управления.

Функция включения для двух наборов имеет вид:

$$u(\Phi_1,\Phi_2,L)=\begin{cases} \text{невкл}(\Phi_1)*\text{невкл}(\Phi_2), & \text{если } (\alpha_1=\text{н} \,\&\, \alpha_2=\text{н}) \\ & \text{or } (\alpha_1=1 \,\&\, \alpha_2=\text{н}) \\ & \text{or } (\alpha_1=1 \,\&\, \alpha_2=0) \\ & \text{or } ((\alpha_1=\text{н}) \,\&\, (\alpha_2=0)) \\ \text{вкл}(\Phi_1)*\text{невкл}(\Phi_2), & \text{если } ((\alpha_1=0) \,\&\, (\alpha_2=\text{н})) \\ & \text{or } ((\alpha_1=0) \,\&\, (\alpha_2=0)) \\ \text{невкл}(\Phi_1)*\text{вкл}(\Phi_2), & \text{если} ((\alpha_1=\text{н}) \,\&\, (\alpha_2=1)) \\ & \text{or } ((\alpha_1=1 \,\&\, \alpha_2=1)) \\ \text{вкл}(\Phi_1)*\text{вкл}(\Phi_2), & \text{если} ((\alpha_1=0)\&(\alpha_2=1)), \end{cases}$$

где операция * означает одновременное выполнение двух одноместных функций.

Функции включения $u(\Phi_1,\Phi_2,L)$ элементов Φ_1, Φ_2 поставим в соответствие цепочку команд.

1. ЕСЛИ$((\alpha_1=0) \&(\alpha_2=1))$ ТО { ВКЛ (Φ_1) , ВКЛ (Φ_2) }
2. ЕСЛИ$((\alpha_1=\text{н}) \,\&\, (\alpha_2=1))$ or $((\alpha_1=1 \,\&\, \alpha_2=1))$ ТО ВКЛ (Φ_2)
3. ЕСЛИ $((\alpha_1=0) \,\&\, (\alpha_2=\text{н}))$ or $((\alpha_1=0) \,\&\, (\alpha_2=0))$ ТО ВКЛ (Φ_1).

Отметим, что алгоритм управления реализует функцию включения только в тех точках булеана, в которых есть включение хотя бы одного функционального элемента. Значение "н" не вносится в алгоритм, так как над функциональными элементами не производится никаких действий.

Функция ввода входных переменных $inp(\Phi_1,\Phi_2)$ определена как мультипликативная функция вида

$inp(\Phi_1,\Phi_2)= inp(\Phi_1)*inp(\Phi_2)$,

где одноместная функция $inp(\Phi_j)$ выполняется только при включении элемента Φ_j. Поставим ей в соответствие цепочку:

1. ЧТЕНИЕ (Φ_1)

2. ЧТЕНИЕ (Φ_2) .

С учетом заданных ограничений: функция ввода выполняется только при включении элемента, сохранение отношения порядка для фиксированного элемента Φ_j - алгоритм управления запишется в следующем виде:

1. ЕСЛИ ((α_1 =0) & (α_2=1))

   ТО (ЧТЕНИЕ (Φ_1), ВКЛ (Φ_1), ЧТЕНИЕ(Φ_2), ВКЛ (Φ_2))

2. ЕСЛИ ((α_1=н & α_2=1) or (α_1=1 & α_2=1))

   ТО (ЧТЕНИЕ (Φ_2) ВКЛ (Φ_2))

3. ЕСЛИ ((α_1=0 & α_2=н) or (α_1=0 & α_2=0))

   ТО (ЧТЕНИЕ (Φ_1), ВКЛ (Φ_1)) .

Аналогично формируется текст при отключении элементов Φ_1, Φ_2.

Аналогично формируется текст при включении или отключении m элементов.

Пусть состояние терма в i-й момент времени характеризуется m_1 набором с включением элементов и m_2 наборами с отключением элементов. Тогда строится текст алгоритма для m_1 набора с включением элементов. Аналогично строится текст для m_2 наборов с отключением элементов.

Общий текст алгоритма управления складывается из двух полученных текстов по правилу конкатенации.

Формирование переходов между последовательными асинхронными событиями с освобождением процессора на время $\tau = t_i - t_{i-1}$ и возвратом к следующей команде по тексту алгоритма в реальном времени будем проводить командой УВИ(τ), где $\tau = t_i - t_{i-1}$.

Отметим, что представленный текст алгоритма неэффективен в смысле наличия повторяющихся операторов. Например, в тексте встречается несколько раз проверка условий ($\alpha_2=1$) или ($\alpha_1=0$). Поэтому над текстом проводится некоторые оптимизирующие преобразования.

4.5 Термальный метод построения асинхронного алгоритма

Представим терм в виде бинарного дерева, в котором каждому поддереву соответствует тройка

< операнд, операция, операнд>.

Здесь операнд есть либо поддерево нижнего уровня, либо висячая вершина, которой соответствует функциональный элемент. Операцией может быть любая бинарная операция функционального исчисления.

4.5.1 Алгоритмизация функции управления функциональным элементом

Рассмотрим терм $T=\Phi_i$. Его управляющий вектор есть

$v(\Phi_i)(t) = \{t_0, t_1\}$,

$<v(\Phi_i), u(\Phi_i), Inp(\Phi_i)>(t_0)$,

$<v(\Phi_i), u(\Phi_i), Out(\Phi_i)>(t_1)$.

Как видно из записи терма, L_i=TRUE как при включении, так и отключении элемента Φ_i.

Функция включения терма в момент времени $t=t_0$ есть

$\gamma(\Phi_i) = t_0$,

$\mathrm{Inp}(\Phi_i)$,

{ВКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$}, если ($\alpha(\Phi_i)=0$) = TRUE;

аналогично в момент $t_1=t_0+\tau_i$ применяется функция отключения (3.6):

$\gamma(\Phi_i) = t_1$,

{ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$}, если ($\alpha(\Phi_i)=1$) =TRUE,

OUT(Φ_i).

Данному управляющему вектору терма $T=\Phi_i$ поставим в соответствие следующий текст алгоритма управления на внутреннем языке, исполняемый в момент времени t_0:

1. ЧТЕНИЕ (Φ_i)
2. ЕСЛИ ($\alpha(\Phi_i)=0$) =TRUE

   ТО {ВКЛ(Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$}
3. УВИ (t_i) (4.1)
4. ЕСЛИ ($\alpha(\Phi_i)=1$) =TRUE

   ТО {ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$}
5. ЗАПИСЬ (Φ_i)

Полученный текст (4.1) алгоритма управления функциональной задачей Φ_i состоит из пяти операторов. Первый оператор (или команда) есть управление функциональным элементом в информационном пространстве - ввод данных для работы Φ_i. Второй оператор включает Φ_i в работу при условии, что элемент Φ_i в это время отключен. Естественным считается перед включением Φ_i в работу передать ему данные, а после отключения Φ_i (оператор 4) передать вычисленные данные в общее поле памяти (оператор 5). Особенность третьего оператора заключается в том, что, сохраняя непрерывность функции управления на отрезке выполнения функциональной задачи $Ф_i$, он реализует механизм освобождения процессора на время (τ_i) с последующим включением в момент времени ($t_0+\tau_i$) следующего за ним оператора. Таким образом, приведенный текст алгоритма (4.1) начинает свою работу в момент t_0 и заканчивает ее в момент $t_1=t_0+\tau_i$.

Отметим, что приведенный алгоритм управления справедлив для такого функционального элемента $Ф_i$, время работы которого является заданной константой и требуется команда отключения элемента. Для функционального элемента $Ф_i$ с автоматическим отключением оператор 4 не требуется. Для однозначного и автоматизированного формирования алгоритма управления эти сведения должны быть заранее известны по каждому элементу функционального базиса.

4.5.2 Алгоритмизация функции управления термом вида $T=Ф_iСНФ_j$

Функционированию составного терма T вида $T=Ф_i$ СН $Ф_j$ соответствует функция управления (3.7):

U(T) = ВЫЗОВ U($Ф_i$), ВЫЗОВ U($Ф_j$) при $t=t_0$,

где t_0 - время включения терма T.

Выполнению такого терма соответствуют процессы $\{t_0,t_1\}$, $\{t_0,t_2\}$, происходящие под управлением функций U($Ф_i$), U($Ф_j$). В момент t_0 происходит одновременное включение термов $Ф_i$ и $Ф_j$, в моменты t_1, t_2 - их соответствующее отключение.

Тогда функции управления U(T) термом $T=Ф_i$ СН $Ф_j$ поставим в соответствие следующий алгоритм управления U(T)(t_0):

1.ВЫЗОВ U($Ф_i$)(t_0), (4.2)

2.ВЫЗОВ U($Ф_j$)(t_0).

Данный алгоритм в момент времени t_0 вызывает на выполнение алгоритмы управления термами $T=\Phi_i$ и $T=\Phi_j$.

Функции включения базового терма $T=\Phi_i$ (3.5) поставим в соответствие следующий алгоритм асинхронного управления $U(\Phi_i)(t_0)$:

1. ЧТЕНИЕ (Φ_i),
2. ЕСЛИ ($\alpha(\Phi_i)=0$) = TRUE TO
   {ВКЛ(Φ_i), $\alpha(\Phi_i) := (\alpha(\Phi_i)+1) \bmod 2$},
3. УВИ (τ_i),
4. ЕСЛИ ($\alpha(\Phi_i)=1$) =TRUE TO
   {ОТКЛ (Φ_i), $\alpha(\Phi_i) := (\alpha(\Phi_i)+1) \bmod 2$},
5. ЗАПИСЬ (Φ_i).

Аналогично записывается ААУ $U(\Phi_j)$ для этого же момента времени.

Для учета всех ситуаций на временной оси и создания единого алгоритма введем логическую переменную β_1:

$$\beta_1 = \begin{cases} 1, \text{ если } \tau_i > \tau_j, \\ 0, \text{ если } \tau_i \le \tau_j. \end{cases}$$

Единый алгоритм запишется в следующем виде:

1. ЧТЕНИЕ Φ_i,
2. ВКЛ (Φ_i),
3. ЧТЕНИЕ Φ_j,
4. ВКЛ (Φ_j)
5. ЕСЛИ $\beta_1=1$ ТО { УВИ (τ_j), ОТКЛ (Φ_j). ЗАПИСЬ (Φ_j)}
   ИНАЧЕ {УВИ (τ_i), ОТКЛ (Φ_i), ЗАПИСЬ (Φ_i)}
6. ЕСЛИ $\beta_1=1$ ТО {УВИ ($\tau_i-\tau_j$), ОТКЛ (Φ_i), ЗАПИСЬ (Φ_i)}
   ИНАЧЕ {УВИ ($\tau_j-\tau_i$), ОТКЛ (Φ_j), ЗАПИСЬ (Φ_j)}.

4.5.3 Алгоритмизация функции управления термом вида $T=Ф_i$ СК $Ф_j$

Функционированию составного терма T вида $T=Ф_i$ СК $Ф_j$ соответствует функция управления (3.9):

$$U(T)=\begin{cases} \text{ВЫЗОВ } U(Ф_i), \text{ если } \tau_i >= \tau_j, \text{ при } t=t_0, \\ \text{ВЫЗОВ } U(Ф_j), \text{ если } \tau_i < \tau_j, \text{ при } t=t_0; \\ \\ \text{ВЫЗОВ } U(Ф_j), \text{ если } \tau_i >= \tau_j, \text{ при } t=t_0+\tau_i-\tau_j, \\ \text{ВЫЗОВ } U(Ф_i), \text{ если } \tau_i < \tau_j, \text{ при } t=t_0+|\tau_i-\tau_j|. \end{cases}$$

Выполнение терма T вида $T=Ф_i$ СК $Ф_j$ сводится к запуску процессов $\{t_0, t_2\}$, $\{t_1, t_2\}$, происходящих под действием функций управления $U(Ф_i)$, $U(Ф_j)$. В момент t_0 происходит включение элемента $Ф_j$, если длительность его выполнения больше длительности $Ф_i$ ($\tau_i \le \tau_j$), тогда в момент t_1 – включение $Ф_i$, либо наоборот, а в момент t_2 их соответствующее отключение.

Пусть для определенности $\tau_i > \tau_j$. Тогда функции управления U(T) термом $T=Ф_i$ СК $Ф_j$ поставим в соответстане следующий алгоритм управления $U(T)(t_0)$:

1.ВЫЗОВ $U(Ф_i)(t_0)$,

2.УВИ(τ), (4.3)

3.ВЫЗОВ $U(Ф_j)(t_0+\tau)$.

Данный алгоритм в момент времени t_0 вызывает на выполнение алгоритм управления термом $T=Ф_i$. Через время $\tau=\tau_i-\tau_j$ он самовключается и вызывает алгоритм управления термом $T=Ф_j$.

Функции включения базового терма $T=Ф_i$ (3.5) поставим в соответствие следующий алгоритм асинхронного управления $U(Ф_i)(t_0)$:

1. ЧТЕНИЕ ($Ф_i$)

2 ЕСЛИ $(\alpha(\Phi_i)=0)$ = TRUE TO

$\{$ВКЛ(Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2\}$,

3. УВИ (τ_i),

4. ЕСЛИ $(\alpha(\Phi_i)=1)$ =TRUE TO

$\{$ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2\}$,

5. ЗАПИСЬ (Φ_i).

Аналогично записывается ААУ $U(\Phi_i)$ для момента времени $t = t_0 + \tau$.

По аналогии с операцией "совпадение по началу" введем логическую переменную β_1:

$$\beta_1 = \begin{cases} 1, \text{ если } \tau_i > \tau_j, \\ 0, \text{ если } \tau_i \leq \tau_j. \end{cases}$$

Единый алгоритм запишется в следующем виде:

1. ЕСЛИ $\beta_1=1$ ТО (ЧТЕНИЕ (Φ_i), ВКЛ (Φ_i), УВИ $(\tau_i-\tau_j)$)

ИНАЧЕ (ЧТЕНИЕ (Φ_j), ВКЛ (Φ_j), УВИ $(\tau_j-\tau_i)$),

2. ЕСЛИ $\beta_1=1$ ТО (ЧТЕНИЕ (Φ_j), ВКЛ (Φ_j), УВИ $(\tau_j-\tau_i)$)

ИНАЧЕ (ЧТЕНИЕ (Φ_i), ВКЛ (Φ_i), УВИ $(\tau_i-\tau_j)$),

3. ОТКЛ (Φ_i),

4. ЗАПИСЬ (Φ_i),

5. ОТКЛ (Φ_j),

6. ЗАПИСЬ (Φ_j).

В едином алгоритме выполняются команды как по включению, так и по отключению базовых элементов. В принципе, это не обязательно, так как многие базовые элементы после завершения работы отключаются автоматически.

4.5.4 Алгоритмизация функции управления термом вида $T = \Phi_i \to \Phi_j$

Функционированию составного терма T вида $T=\Phi_i \to \Phi_j$ соответствует функция управления (3.10):

$$U(T)= \begin{cases} \text{ВЫЗОВ } U(\Phi_i), \text{ при } t=t_0, \\ \text{ВЫЗОВ } U(\Phi_j), \text{ при } t=t_0+\tau_i; \end{cases}$$

Выполнение терма T вида $T=\Phi_i \to \Phi_j$ сводится к запуску процессов $\{t_0, t_1\}$, $\{t_1, t_2\}$, происходящих под действием функций управления $U(\Phi_i)$, $U(\Phi_j)$. В момент t_0 происходит включение элемента Φ_i, а в момент t_1 - отключение Φ_i и включение Φ_j, в момент t_2 отключение Φ_j.

Функции управления U(T) термом $T=\Phi_i \to \Phi_j$ поставим в соответствие следующий алгоритм управления $U(T)(t_0)$:

1.ВЫЗОВ $U(\Phi_i)(t_0)$,

2.УВИ(τ_i), (4.4)

3.ВЫЗОВ $U(\Phi_j)(t_0+\tau)$.

Данный алгоритм в момент времени t_0 вызывает на выполнение алгоритм управления термом $T=\Phi_i$. Через время $\tau=\tau_i$ он самовключается и вызывает алгоритм управления термом $T=\Phi_j$.

Функции включения базового терма $T=\Phi_i$ (3.5) поставим в соответствие следующий алгоритм асинхронного управления $U(\Phi_i)(t_0)$:

1. ЧТЕНИЕ (Φ_i)
2. ЕСЛИ ($\alpha(\Phi_i)=0$) = TRUE TO

   (ВКЛ(Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$),
3. УВИ (τ_i),

4. ЕСЛИ ($\alpha(\Phi_i)$ =1) =TRUE TO

{ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$},

5. ЗАПИСЬ (Φ_j)

Аналогично записывается ААУ $U(\Phi_i)$ для момента времени $t = t_0 + \tau_i$.

Единый алгоритм запишется в следующем виде:

1. ЧТЕНИЕ (Φ_i)
2. ЕСЛИ ($\alpha(\Phi_i)$ =0) = TRUE TO

   {ВКЛ(Φ_i), $\alpha(\Phi_i):= (\alpha(\Phi_i)+1)\mathrm{mod}2$},
3. УВИ (τ_i),
4. ЕСЛИ ($\alpha(\Phi_i)$ =1) =TRUE TO

   {ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$},
5. ЗАПИСЬ (Φ_i)
6. ЧТЕНИЕ (Φ_j)
7. ЕСЛИ ($\alpha(\Phi_j)$ =0) = TRUE TO

   {ВКЛ(Φ_j), $\alpha(\Phi_j):= (\alpha(\Phi_j)+1)\mathrm{mod}2$},
8. УВИ (τ_j),
9. ЕСЛИ ($\alpha(\Phi_i)$ =1) =TRUE TO

   {ОТКЛ (Φ_i), $\alpha(\Phi_i):=(\alpha(\Phi_i)+1)\mathrm{mod}2$},
10. ОТКЛ (Φ_j),
11. ЗАПИСЬ (Φ_j).

4.5.5 Алгоритмизация функции управления динамическим термом

Выполнению динамического терма T под управлением функции U(T) соответствуют процессы $\{t_0,t_1\}$, $\{t_0,t_2\}$ для подтермов Φ_i и Φ_j соответственно, происходящие под управлением функций $U(\Phi_i)$, $U(\Phi_j)$. Подтермы Φ_i и Φ_j выполняются альтернативно, в зависимости от истинности логического выражения, навешенного на терм. Для операции "выбора (+)" эти процессы начинаются в момент t_0, а заканчиваются соответственно в моменты t_1 либо t_2.

Функция управления динамическим термом с операцией "выбора (+)" имеет вид (3.11):

$$U(T)=\begin{cases} \text{ВЫЗОВ } U(\Phi_i), \text{ если } (\alpha=1), & \text{при } t=t_0, \\ \text{ВЫЗОВ } U(\Phi_j), \text{ если } (\alpha=0), & \text{при } t=t_0; \end{cases}$$

t_0 - время включения терма T.

Для операции + - выбора, время включения терма Φ_i совпадает со временем включения терма Φ_i.

Для терма $T=(\alpha=1)\Rightarrow\Phi_i+(\alpha=0)\Rightarrow\Phi_j$, изображенного на рис.3.9, поведение функции управления в t_0-й момент времени можно описать с помощью алгоритма асинхронного управления следующего вида:

1. ЕСЛИ $\alpha=1$

   ТО {ВЫЗОВ $U(\Phi_i)$} (4.5)

   ИНАЧЕ {ВЫЗОВ $U(\Phi_j)$}.

Для терма $T=(\alpha=1)\Rightarrow\Phi_i+(\alpha=0)\Rightarrow\Phi_j$ операндами являются базовые элементы, и для них функция управления определена в п. 3.5.

4.5.6 Алгоритмизация функции управления составным термом

Правило построения функции управления составным термом определим индуктивно, где индукция берется по длине терма.

Для терма вида $T=\Phi_i$ функция управления определяется с помощью выражения (3.5).

Для статического терма вида $T=T_i \otimes T_j$, где $\otimes$ - символ операции на временной оси, $\otimes \in$ { СН, СК, $\rightarrow$}, функция управления определяется с помощью выражений (3.8)-(3.10)

Если T_i, T_j - составные термы, то для них функция управления формируется точно так же, как и для терма T.

Для динамического терма вида $T=(\alpha=1)\Rightarrow\Phi_i$ + $(\alpha=0\Rightarrow\Phi_j$ функция управления определяется по выражениям (3.11).

Рассмотрим метод построения алгоритма асинхронного управления компилирующего типа для составного терма. Суть метода компилирующего типа состоит в том, что он создает ААУ до момента его выполнения.

Будем создавать метод в заданной системе команд $\sigma^{\wedge}$, расширенной командой вызова новых процессов, $\sigma^{м} = \sigma^{\wedge} \cup$ ВЫЗОВ (имя процесса). Метод состоит из двух частей: алгоритма настройки параметров терма и алгоритма генерации вызываемых процессов

п.1.Алгоритм настройки параметров терма.

Метод предполагает работу с составным термом, представленным в виде бинарного дерева вложенных друг в друга подтермов с висячими вершинами из элементов функционального базиса. В качестве настраиваемых параметров для каждого подтерма составного терма, включая и висячие вершины, выступают:

-$t_{вкл i}$ - время включения подтермa;

-$t_{смi}$ - смещение по времени включения второго операнда относительно первого в записи подтерма;

-τ_i - длительность подтерма

Дополнительными параметрами, участвующими в реализации метода, являются адресные указатели на описание подтермов нижнего и верхнего уровней.

Алгоритм настройки параметров заключается в их вычислении по известной структуре терма и формированию системной таблицы с их описанием для всех составных компонент терма. Вход в таблицу определяется помеченной записью.

Одна запись такой таблицы описывает один подтерм либо висячую вершину. Семантика одной записи:

поле1 - признак записи: 0- необработанная запись; 1- запись обработана по первому операнду, 2- запись обработана по второму операнду;

поле2 - код операции: 1- для составного терма; 2- для висячей вершины; для висячей вершины признак записи в поле1 может принимать только два значения { 0, 2}.

поле3 - термальная операция; для кода операции 1 это

{СН, СК, →}; для кода 2 это $\{Ф_i\}$- элементы висячих вершин.

поле4 - время включения подтерма;

поле5 - время включения второго операнда относительно первого;

поле6 - длительность терма;

поле7 - указатель на запись с первым операндом;

поле8 - указатель на запись со вторым операндом;

поле9 - указатель на терм верхнего уровня;

поле10-значение логической переменной, при истинности которого выполняется терм в этой записи.

Алгоритм настройки параметров выполняется в два этапа:

п.1.1. Формирование системной таблицы.

п.1.2. Вычисление настраиваемых параметров.

п.1.1. Формирование системной таблицы.

Таблица формируется путем синтаксического анализа термального описания целевой задачи.

п.1.1.1.Алгоритм синтаксического разбора термального описания целевой задачи с получением обратной польской записи (Poliz). Используется алгоритм Дейкстры со стеком операций.

п1.1.2. Алгоритм преобразования обратной польской записи в системную таблицу. Разработанный алгоритм использует стек операндов. Данный алгоритм для каждой записи формирует нулевое значение поля признаков, код операции, в поле3 вносит термальную операцию, формирует поля с указателями (поля7,8,9) и запоминает (помечает (*)) запись с описанием терма верхнего уровня.

п.1.2. Вычисление настраиваемых параметров.

п.1.2.1. Выполняется алгоритм обхода дерева терма. Данный алгоритм совершает обход по системной таблице, начиная и заканчивая его помеченной записью. При обходе в каждую запись алгоритм входит и выходит "сверху" - один раз и "снизу" - два раза, по числу операндов, входящих в подтерм. Значение поля признаков и кода операций однозначно определяют эти ситуации. В процессе движения алгоритм меняет значение поля признаков. Всего проводится два прохода.

п.1.2.1.1. Первый проход. Формируются поля: длительность подтерма (поле6), смещение второго операнда относительно первого (поле5). Длительность подтерма для базового элемента есть длительность его выполнения, для составного элемента есть величина, определяемая по правилам интерпретации операции подтерма.

Заполнение временных параметров для составного подтерма происходит только после вторичного прихода в этот терм, после обработки второго операнда. По имеющейся информации о длительностях выполнения двух операндов и операции, их связывающей, определяются длительность выполнения подтерма (поле6) и время включения второго операнда относительно первого (поле5). После первого прохода остается не вычисленным только время включения каждого подтерма в единой шкале.

п.1.2.1.2. Второй проход - вычисление времени включения каждого подтерма в единой шкале. Обход начинается с вершины дерева (помеченная запись) и заканчивается последней висячей вершиной.

Для подтерма верхнего уровня время включения равно нулю. Предполагается, что терм начинает свою работу в нулевой момент времени. Это равносильно следующему:

-для операции "совпадение по началу" время включения терма есть время включения первого и второго операндов; время включения первого операнда есть время включения подтерма, описанного в записи по указателю в поле7; время включения второго операнда есть время включения подтерма, описанного в записи по указателю в поле8;

-для операции "совпадение по концу" время включения терма есть время включения первого операнда, если время включения второго операнда относительно первого в поле5 есть величина положительная, в противном случае время включения терма есть время включения второго операнда; если в поле5 время положительное, то второй операнд включается в момент времени, равный сумме четвертого и пятого полей, в противном случае расчетное время есть время включения первого операнда;

-для операции "следования" время включения терма есть время включения первого операнда; второй операнд включается в момент, равный сумме четвертого и пятого полей.

Для каждой из перечисленных операций, если операндом является базовый элемент, то время включения операнда есть время включения базового элемента.

п.2.Алгоритм генерации процессов.

Генерация термальных процессов осуществляется по системной таблице с использованием прямого (сверху- вниз) и обратного (снизу- вверх) хода по указателям в записях таблицы. Как и в п.1. в основе алгоритма лежит метод обхода дерева терма. На каждом шаге движения вызывается новый процесс для составного терма либо описывается исполняемый процесс для базового элемента. Работу алгоритма покажем на следующем примере.

Пример11. Рассмотрим терм из примера6 раздела 2.4.

$T_0=\theta_{оп}$ СН Φ_7

$T=(\alpha_1=1)\Rightarrow(((\alpha_2=1)\Rightarrow\{\ [(\alpha_3=1)\Rightarrow\Phi_1 + (\alpha_3=0)\Rightarrow\Phi_2]$ СК $\Phi_3\}$ $+(\alpha_2=0)\Rightarrow\{[(\alpha_3=1)\Rightarrow\Phi_4 + (\alpha_3=0)\Rightarrow\Phi_5]$ СНΦ_6 $\}) \to T_0\ ) + (\alpha_1=0)\Rightarrow$ $(\Phi_8 \to T_0)$,

где $\theta_{оп}$ -невыполняемый терм нулевой длительности.

$\alpha=\{\alpha_1,\ \alpha_2,\ \alpha_3\}$.

Представим его в виде дерева термов.

$T= (\alpha_1=1)\Rightarrow T_1 +(\alpha_1=0)\Rightarrow T_2$;

$T_1=T_{11}\to T_0$;

$T_{11}=(\alpha_2=1)\Rightarrow T_{111}+(\alpha_2=0)\Rightarrow T_{112};$

$T_{111}=T_{1111}$ СК $\Phi_3;$

$T_{1111}=(\alpha_3=1)\Rightarrow\Phi_1+(\alpha_3=0)\Rightarrow\Phi_2;$

$T_{112}=T_{1121}$ СН $\Phi_6;$

$T_{1121}=(\alpha_3=1)\Rightarrow\Phi_4+(\alpha_3=0)\Rightarrow\Phi_5;$

$T_2=\Phi_8\rightarrow T_0;$

$T_0=\theta_{оп}$ СН $\Phi_7.$

По заданному термальному описанию построим системную таблицу, имеющую структуру дерева терма с помеченной записью (таблица 4.1).

На втором проходе вычислим требуемые моменты времени, используя при этом длительности базовых элементов (таблица 4.2). При этом будут заполнены поля: время включения терма (поле4), относительное смещение (поле5), длительность терма (поле6). Как видно из таблицы 4.1 помеченная запись $z_{21}(^*)$ содержит описание терма верхнего уровня

$$T=(\alpha_1=1)\Rightarrow T_1+(\alpha_1=0)\Rightarrow T_2$$

Таблица 4.1

Ном. зап.	Призн зап.	Код опер.	Терм опер.	Вр вкл.	Относ. смещ.	Длит. терма	Указ оп1	Указ оп2	Указ отц	Лог. Пер.
z_1	0	2	Φ_1	0	0	10	—	—	$u(z_3)$	$\alpha_3=1$
z_2	0	2	Φ_7	0	0	50	—	—	$u(z_3)$	$\alpha_3=0$
z_3	0	1	+	0	0	50	$u(z_1)$	$u(z_2)$	$u(z_5)$	α_3
z_4	0	2	Φ_3	20	0	30	—	—	$u(z_5)$	
z_5	0	1	СК	0	20	50	$u(z_3)$	$u(z_4)$	$u(z_{11})$	$\alpha_2=1$
z_6	0	2	Φ_4	0	0	60	—	—	$u(z_8)$	$\alpha_3=1$
z_7	0	2	Φ_5	0	0	40	—	—	$u(z_8)$	$\alpha_3=0$
z_8	0	1	+	0	0	60	$u(z_6)$	$u(z_7)$	$u(z_{10})$	α_3
z_9	0	2	Φ_6	0	0	20	—	—	$u(z_{10})$	
z_{10}	0	1	СН	0	0	80	$u(z_8)$	$u(z_9)$	$u(z_{11})$	$\alpha_2=0$
z_{11}	0	1	+	0	0	60	$u(z_5)$	$u(z_{10})$	$u(z_{15})$	α_2

z_{12}	0	2	$\theta_{оп}$	60	0	0	—	—	$u(z_{14})$	
z_{13}	0	2	Φ_4	60	0	70	—	—	$u(z_{14})$	
z_{14}	0	1	CH	60	0	70	$u(z_{12})$	$u(z_{13})$	$u(z_{15})$	
z_{15}	0	1	$\to$	0	60	130	$u(z_{11})$	$u(z_{14})$	$u(z_{21})$	$\alpha_1=1$
z_{16}	0	2	Φ_8	0	0	80	—	—	$u(z_{20})$	
z_{17}	0	2	$\theta_{оп}$	80	0	0	—	—	$u(z_{19})$	
z_{18}	0	2	Φ_7	80	0	70	—	—	$u(z_{19})$	
z_{19}	0	1	CH	80	0	70	$u(z_{17})$	$u(z_{18})$	$u(z_{20})$	
z_{20}	0	1	$\to$	0	80	150	$u(z_{16})$	$u(z_{19})$	$u(z_{21})$	$\alpha_1=0$
$z_{21}(^*)$	0	1	+	0	0	150	$u(z_{15})$	$u(z_{20})$	—	α_1

Из этой записи также видно, что термы T_1 и T_2, входящие как два операнда в терм верхнего уровня, описаны соответственно в записях z_{15} и z_{20}. Терм $T_1=T_{11}\to T_0$ связывает подтермы T_{11} и T_0, которые как операнды описаны соответственно в записях z_{11} и z_{14}. Продолжая таким образом, можно установить соответствие между любым подтермом и его записью в системной таблице.

Длительности функциональных задач зададим таблицей 4.2:

Таблица 4.2

Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	Φ_7	Φ_8	$\theta_{оп}$
10	50	30	60	40	20	70	80	0

Алгоритм генерации процессов создает их, начиная с терма Т. Входим в помеченную запись. Она описывает составной терм с операцией выбора. Для динамического терма формируем управление в виде (4.5)

U(T)(t_0=0): ЕСЛИ α_1=1

ТО {ВЫЗОВ U(T_1)(t_1=t_0)}

ИНАЧЕ {ВЫЗОВ U(T_2)(t_2=t_0)}.

Одновременно с формированием текста процесса формируется и время его включения в виде команд установки временного интервала УВИ(τ); τ - время смещения относительно запускаемого процесса. В данном процессе время вызова процессов $U(T_1)$, $U(T_2)$ совпадает со временем запуска основного процесса.

Переходим в запись z_{15}, соответствующую первому операнду. Она описывает составной терм с операцией следования. Для процесса $U(T_1)$ формируем управление в виде (4.4):

$U(T_1)(t_1)$: ВЫЗОВ $U(T_{11})(t_1)$,

УВИ($\tau(T_{11})$),

ВЫЗОВ $U(T_0)(t_1+\tau(T_{11}))$.

$\tau(T_{11})$ - длительность составного терма T_{11} или содержимое поля6 записи z_{11}. В записи z_{11}, соответствующей процессу $U(T_{11})$, описан динамический объект, поэтому текст процесса будет иметь вид (4.5):

$U(T_{11})(t_1)$: ЕСЛИ $\alpha_2=1$

ТО {ВЫЗОВ $U(T_{111})(t_1)$}

ИНАЧЕ {ВЫЗОВ $U(T_{112})(t_1)$}.

Процесс $U(T_{111})$ является составным и описывается операцией "совпадение по концу". Ему соответствует текст процесса вида (4.3):

$U(T_{111})(t_1)$: ЕСЛИ $\tau>0$

ТО ВЫЗОВ $U(T_{1111})(t_1)$

ИНАЧЕ ВЫЗОВ $U(\Phi_3)(t_1)$,

УВИ(τ),

ЕСЛИ $\tau>0$

ТО ВЫЗОВ $U(\Phi_3)(t_1+\tau)$

ИНАЧЕ ВЫЗОВ $U(T_{1111})(t_1+\tau)$.

Здесь $\tau=\tau(T_{111}) - \tau(\Phi_3)$ или содержимое поля5 системной таблицы записи z_3. Для рассматриваемого примера $\tau=20$.

Процессу $U(T_{1111})$, описанному в записи z_3 и включаемому динамически в момент t_{1111}, ставится в соответствие динамический объект, вызов которого имеет вид

$U(T_{1111})(t_{1111})$: ЕСЛИ $\alpha_3=1$

ТО ВЫЗОВ $U(\Phi_1)(t_{1111})$}

ИНАЧЕ {ВЫЗОВ $U(\Phi_2)(t_{1111})$}.

Здесь время включения t_{1111} определяется динамически:

$$t_{1111} = \begin{cases} t_1, \text{ если } \tau=\tau(T_{1111}) - \tau(\Phi_3)>0, \\ t_1+\tau, \text{ в противном случае.} \end{cases}$$

Вызов базовых процессов $U(\Phi_1)(t_{1111})$, $U(\Phi_2)(t_{1111})$ запишется в виде (4.1):

$U(\Phi_1)(t_{1111})$:

1. ЧТЕНИЕ (Φ_1)
2. ЕСЛИ ($\alpha(\Phi_1)=0$) =TRUE

   ТО {ВКЛ(Φ_1), $\alpha(\Phi_1):=(\alpha(\Phi_1)+1)\bmod 2$}
3. УВИ ($\tau(\Phi_1)$)
4. ЕСЛИ ($\alpha(\Phi_1)=1$) =TRUE

   ТО {ОТКЛ (Φ_1), $\alpha(\Phi_1):=(\alpha(\Phi_1)+1)\bmod 2$}
5. ЗАПИСЬ (Φ_1).

$U(\Phi_2)(t_{1111})$:

1. ЧТЕНИЕ (Φ_2)
2. ЕСЛИ ($\alpha(\Phi_2)=0$) =TRUE

   ТО {ВКЛ(Φ_2), $\alpha(\Phi_2):=(\alpha(\Phi_2)+1)\bmod 2$}
3. УВИ ($\tau(\Phi_2)$)

4. ЕСЛИ $(\alpha(\Phi_2) = 1) = TRUE$

ТО {ОТКЛ (Φ_2), $\alpha(\Phi_2):=(\alpha(\Phi_2)+1)mod2$}

5. ЗАПИСЬ (Φ_2).

После формирования процессов $U(\Phi_1)(t_{1111})$, $U(\Phi_2)(t_{1111})$, описанных в записях z_1, z_2, происходит возврат вверх по дереву терма в запись z_3, по указателю в поле9 $(u(z_3))$. Так как запись z_3 проанализирована полностью, происходит возврат в запись z_5. Для текущей записи обработка второго операнда сводится к движению вниз по дереву терма в запись z_4.

Формируем процесс $U(\Phi_3)(t_1+\tau)$.

$U(\Phi_3)(t_1+\tau)$:

1. ЧТЕНИЕ (Φ_3)

2. ЕСЛИ $(\alpha(\Phi_3) = 0) = TRUE$

ТО {ВКЛ(Φ_3), $\alpha(\Phi_3):=(\alpha(\Phi_3)+1)mod2$}

3. УВИ $(\tau(\Phi_3))$

4. ЕСЛИ $(\alpha(\Phi_3) = 1) = TRUE$

ТО {ОТКЛ (Φ_3), $\alpha(\Phi_3):=(\alpha(\Phi_3)+1)mod2$}

5. ЗАПИСЬ (Φ_3).

Из записи z_5 после перехода в z_{11} возвращаемся в запись z_{10}, содержащую составной терм с операцией "совпадение по началу".

$U(T_{112})(t_1)$:

ВЫЗОВ $U(T_{1121})(t_1)$

ВЫЗОВ $U(\Phi_6)(t_1)$.

Для первого операнда из записи z_8 вытекает динамический процесс.

$U(T_{1121})(t_1)$:

ЕСЛИ $\alpha_3=1$

ТО ВЫЗОВ $U(\Phi_4)(t_1)$

ИНАЧЕ {ВЫЗОВ $U(\Phi_5)(t_1)$}.

Для базовых элементов Φ_4, Φ_5 получим исполнительные процессы.

$U(\Phi_4)(t_1)$:

1. ЧТЕНИЕ (Φ_4)
2. ЕСЛИ ($\alpha(\Phi_4)=0$) =TRUE

   ТО {ВКЛ(Φ_4), $\alpha(\Phi_4):=(\alpha(\Phi_4)+1)\mathrm{mod}2$}
3. УВИ ($\tau(\Phi_4)$)
4. ЕСЛИ ($\alpha(\Phi_4)=1$) =TRUE

   ТО {ОТКЛ (Φ_4), $\alpha(\Phi_4)=(\alpha(\Phi_4)+1)\mathrm{mod}2$}
5. ЗАПИСЬ (Φ_4)

$U(\Phi_5)(t_1)$:

1. ЧТЕНИЕ (Φ_5)
2. ЕСЛИ ($\alpha(\Phi_5)=0$) =TRUE

   ТО {ВКЛ(Φ_5), $\alpha(\Phi_5):=(\alpha(\Phi_5)+1)\mathrm{mod}2$}
3. УВИ ($\tau(\Phi_5)$)
4. ЕСЛИ ($\alpha(\Phi_5)=1$) =TRUE

   ТО {ОТКЛ (Φ_5), $\alpha(\Phi_5):=(\alpha(\Phi_5)+1)\mathrm{mod}2$}
5. ЗАПИСЬ (Φ_5).

Возврат из записей z_6, z_7 и z_8 в z_{10} приводит к формированию базового процесса из z_9.

$U(\Phi_6)(t_1)$:

1. ЧТЕНИЕ (Φ_6)
2. ЕСЛИ ($\alpha(\Phi_6)=0$) =TRUE

   ТО {ВКЛ(Φ_6), $\alpha(\Phi_6):=(\alpha(\Phi_6)+1)\mathrm{mod}2$}
3. УВИ ($\tau(\Phi_6)$)
4. ЕСЛИ ($\alpha(\Phi_6)=1$) =TRUE

ТО {ОТКЛ (Φ_6), $\alpha(\Phi_6):=(\alpha(\Phi_6)+1)mod2$}

5. ЗАПИСЬ (Φ_6)

Последовательный подъем по дереву приводит нас к записи z_{14}. Выпишем процессы для нее и ее висячих вершин.

$U(T_0)(t_1+\tau(T_{11}))$:

ВЫЗОВ $U(\theta_{оп})(t_1+\tau(T_{11}))$

ВЫЗОВ $U(\Phi_7)(t_1+\tau(T_{11}))$.

$U(\theta_{оп})(t_1+\tau(T_{11}))$: $\varnothing$ (пустой оператор, исходя из определения терма $\theta_{оп}$).

$U(\Phi_7)(t_1+\tau(T_{11}))$:

1. ЧТЕНИЕ (Φ_7)

2. ЕСЛИ ($\alpha(\Phi_7)=0$) =TRUE

ТО {ВКЛ(Φ_7), $\alpha(\Phi_7):=(\alpha(\Phi_7)+1)mod2$}

3. УВИ ($\tau(\Phi_7)$)

4. ЕСЛИ ($\alpha(\Phi_7)=1$) =TRUE

ТО {ОТКЛ (Φ_7), $\alpha(\Phi_7):=(\alpha(\Phi_7)+1)mod2$}

5. ЗАПИСЬ (Φ_7).

Дальнейший возврат по дереву терма приводит к состоянию, описываемому в записи z_{20}. Для нее и нижестоящих записей имеем следующие процессы:

$U(T_2)(t_2)$: ВЫЗОВ $U(\Phi_8)(t_2)$,

УВИ($\tau(\Phi_8)$),

ВЫЗОВ $U(T_0)(t_2+\tau(\Phi_8))$.

$U(\Phi_8)(t_2)$:

1. ЧТЕНИЕ (Φ_8)

2. ЕСЛИ ($\alpha(\Phi_8)=0$) =TRUE

ТО {ВКЛ(Φ_8), $\alpha(\Phi_8):=(\alpha(\Phi_8)+1)mod2$}

3. УВИ ($\tau(\Phi_8)$)

4. ЕСЛИ ($\alpha(\Phi_8)=1$) =TRUE

ТО {ОТКЛ (Φ_8), $\alpha(\Phi_8):=(\alpha(\Phi_8)+1)\mathrm{mod}2$}

5. ЗАПИСЬ (Φ_8).

$U(T_0)(t_2+\tau(\Phi_8))$:

ВЫЗОВ $U(\theta_{оп})(t_2+\tau(\Phi_8))$

ВЫЗОВ $U(\Phi_7)(t_2+\tau(\Phi_8))$.

$U(\theta_{оп})(t_2+\tau(\Phi_8))$: $\varnothing$ (пустой оператор, исходя из определения терма $\theta_{оп}$).

$U(\Phi_7)(t_2+\tau(\Phi_8))$:

1. ЧТЕНИЕ (Φ_7)

2. ЕСЛИ ($\alpha(\Phi_7)=0$) =TRUE

ТО {ВКЛ(Φ_7), $\alpha(\Phi_7):=(\alpha(\Phi_7)+1)\mathrm{mod}2$}

3. УВИ ($\tau(\Phi_7)$)

4. ЕСЛИ ($\alpha(\Phi_7)=1$) =TRUE

ТО {ОТКЛ (Φ_7), $\alpha(\Phi_7):=(\alpha(\Phi_7)+1)\mathrm{mod}2$}

5. ЗАПИСЬ (Φ_7).

Как видно из примера, процессы $U(\theta_{оп})$, $U(\Phi_7)$ описаны дважды. По сути, это два экземпляра одного процесса, включаемые по разным логическим условиям и в разное время.

5 Задача приведения космического аппарата в ориентированное положение

5.1 Основные положения

В качестве технической системы рассмотрим космический аппарат (КА) и задачу приведения его в ориентированное положение. Он состоит из функциональных подсистем и бортового вычислительного комплекса (БВК)[14,20]. Подсистемы состоят из приборов (функциональных элементов), имеют сложную иерархическую структуру и взаимодействуют между собой на уровне приборов в процессе выполнения целевой задачи. Такое взаимодейстие осуществляется через БВК средствами специального бортового программного обеспечения (БПО). Для космических аппаратов зондирования Земли такое БПО образует иерархическую структуру, аналогичную структуре подсистем КА. На верхнем уровне это алгоритмы планирования (АП), которые по определенному в них правилу и исходным данным с наземного комплекса управления осуществляют планирование последовательности включения функциональных режимов, образующих следующий по порядку иерархический уровень. Эти алгоритмы обеспечивают комплексное функционирование подсистем и элементов КА при выполнении определенной целевой задачи либо их совокупности. При этом каждой целевой задаче соответствует свой алгоритм.

Следующий иерархический уровень образуют программы бортового программного обеспечения, обеспечивающие работу соответствующих подсистем бортового комплекса управления. Их включение производится как по определенной жесткой временной диаграмме, так и условиям, истинность которых определяется на множестве данных по результатам собственного анализа.

Функциональные программы, образующие следующий уровень иерархии, выполняют расчет данных для работы подсистем КА, управления БА, формирования контрольной информации о работе подсистем БА.

Решение задачи « Приведение КА в ориентированное положение» сводится к выполнению действий:

1) Подготовка к включению, включение систем и приборов КА, участвующих в выполнении данной задачи,
2) Подготовка к работе комплексной двигательной установки (КДУ);
3) Гашение угловых скоростей (ГУС) КА;
4) Перевод КА в ориентированное положение.

К моменту выполнения данной задачи бортовая аппаратура системы управления движением (БА СУД) выключена. В данной задаче используются следующие приборы (элементы функционального базиса): комплексная двигательная установка (КДУ), блок датчиков угловых скоростей (БДУС1, БДУС2), построитель местной вертикали (ПМВ1, ПМВ2), построитель местной вертикали высотный (ПВВ1, ПВВ2), приборы включения питания (ВП), отключение нагревателя воздуха (ОНВ), сигнализатор давления гидроаккумуляторов (СДГА), сигнализатор давления наддува (СДН).

КДУ состоит из маршевого двигателя для управления движением центра масс и управляемых реактивных двигателей (УРД) для управления движением относительно центра масс, в частности, для гашения угловых скоростей. Данные с блока датчиков угловых скоростей используются в операции гашения угловых скоростей. Перевод КА в ориентированное положение производится путем построения (установки) местной вертикали. Два типа приборов ПМВ и ПВВ необходимы для построения местной вертикали на разновысотных орбитах. Прибор ВП подает питание на все системы КА. Отключение нагревателя воздуха производится при переходе из дежурного полета в ориентированный. С использованием сигнализаторов давления УРД выполняют переход КА в ориентированное положение.

Рассмотрим одну из возможных реализаций, имеющих место в практике функционирования КА. Задача приведения КА в ориентированное положение выполняется как первоначально, при выходе на орбиту, так и неоднократно в нештатных ситуациях, связанных с потерей КА ориентации. Признак первоначального включения определяется единичным значением индикатора первичного включения ($\alpha 1=1$).

В начале работы проверяется значение индикатора первичного включения. Если его значение равно единице, включается прибор включения питания всех систем ВП.

Через 110с с момента включения ВП (при первом включении) должна включиться комплексная двигательная установка (КДУ) в режиме подготовки и выполниться команда повторного включения путем изменения значения индикатора первичного включения ($\alpha 1:=0$).

Через 33с от момента включения КДУ необходимо обеспечить включение построителей местной высотной (ПМВ1, 2) и высотной (ПВВ1,2) вертикали по следующей логике: если признак отработки разгонного импульса КДУ ($\alpha 2$) равен единице и признак неработоспособности прибора ПВВ1 ($\alpha 3$) равен нулю, то включить его, или если $\alpha 2=1$ и пр.

неработоспособности прибора ПВВ2 ($\alpha4$) равен нулю ($\alpha4=0$), то включить ПВВ2. В случае, если признак отработки разгонного импульса КДУ ($\alpha2$) равен нулю ($\alpha2=0$), то включить ПМВ1 и ПМВ2, включить ПВВ1 при $\alpha3=0$ и ПВВ2 при $\alpha4=0$.

Через 96с после включения КДУ в режиме подготовки необходимо включить БДУС1 и БДУС-2.

Через 145 с после включения КДУ в режиме подготовки необходимо включить прибор отключения нагревателей воздуха (ОНВ), подключить сигнализатор давления гидроаккумуляторов (СДГА), включить управляемые реактивные двигатели (УРД) для гашения угловых скоростей изделия, после чего включить сигнализатор давления наддува (СДН) для наддува баков окислителя и горючего. После гашения угловых скоростей (ГУС) необходимо перейти в режим поиска Земли, построения вертикали места и плоскости орбиты, который продолжается около 1800с. По окончании его начинается участок коррекции от построителя местной вертикали, длящийся порядка 5480с.

В соответствии с общими материалами по логике управления КА при приведении его в ориентированное положение опишем целевую задачу на символическом языке.

Решение задачи реализуется следующей бортовой аппаратурой.

1) КДУ - комплексная двигательная установка;
2) БДУС1 и БДУС2 - блоки датчиков угловых скоростей;
3) БУС - блок устройств согласования;
4) ПМВ1 и ПМВ2 - построитель местной вертикали первый и второй;
5) ПВВ1 и ПВВ2 - построитель вертикали высотный первый и второй;

6) ВП - прибор включения питания;

7) ОНВ - прибор отключения нагревателя воздуха;

8) СДГА - сигнализатор давления гидроаккумуляторов;

9) СДН - сигнализатор давления наддува.

Список функциональных задач (приборов) с длительностью и условным обозначением:

ВП	10 с	f001;
КДУ в режиме подготовки	125 с	f002;
ПМВ1	200 с	f003;
ПМВ2	200 с	f004,
ПВВ1	150 с	f005;
ПВВ2	150 с	f006;
БДУС-1	600 с	f007;
БДУС-2	600 с	f008;
ОНВ	100 с	f009;
СПр «СДГА»	200 с	f010;
СПр «СДН»	100 с	f011;
Участок поиска Земли	1800 с	f012;
Участок коррекции	5840 с	f014;
Пустой терм 1	110 с	f100;
Пустой терм 2	33 с	f101,
Пустой терм 3	96 с	f102;
Пустой терм 4	145 с	f103;
Пустой лямбда - терм	0 с	f200.

5.2 Термальное описание асинхронного алгоритма

Согласованная работа построителей высотной вертикали описывается термом $t9:=((\alpha 3=0)=>f005+(\alpha 3=1)=>f200)$ СН $((\alpha 4=0)=>f006+(\alpha 4=1)=>f201)$.

Согласованная работа построителей вертикали высотной и местной высотной задается термом t10:=(f003CHf004)CHt9. С учетом признака отработки разгонного импульса (α2) их согласованная работа запишется термом

t8:=(α2=1)=>t10 + (α2=0)=>t9. Согласуем работу КДУ с построителем местной и высотной вертикали с помощью терма t6=f002 CH (f101->t8), где f101- пустой терм, определяющий задержку выполнения построителей вертикали относительно начала работы КДУ.

Отключение нагревателя воздуха и подключение сигнализатора давления гидроаккумуляторов описывается термом t18:=f009CHf010. Включение программы для наддува баков окислителя и горючего, затем переход в режим поиска Земли и после этого переход на участок коррекции описываются термом t19:=((f011->f012)->f014). Учитывая, что два последних терма включаются через 145 с после включения КДУ, можно составить терм t15:=f103->t17, где t17:=t18->t19, f103 - пустой терм, определяющий задержку на 145с. Включение датчиков угловых скоростей на 96с после включения КДУ описывается термом t14:=f102->t16, t16:=f007CHf008, f102- пустой терм, определяющий задержку на 96с. Согласованность работы датчиков угловых скоростей с приборами ОНВ и СДГА зададим термом t7:=t14CHt15. Их согласованность с приборами построения вертикали зададим термом t5:=t6CHt7. С учетом работы КДУ получим терм t4:=f002CHt5. Если работу прибора включения питания описать термом t3:=f001CHf100, то его согласованную с КДУ работу можно описать термом t2:=t3->t4. С учетом возможности повторного включения получим терм t1=(α1=1)=>t2 +

($\alpha1=0$)=>t41, где терм t41 эквивалентен терму t4 и описывает совместную работу перевод КА в ориентированное положение при повторном включении.

Базовые элементы системы:

f001 - прибор включения питания ВП;

f002 - комплексная двигательная установка КДУ;

f003 - построитель местной вертикали ПМВ1;

f004 - построитель местной вертикали ПМВ2.

f005 - построитель местной вертикали высотный ПВВ1;

f006 - построитель местной вертикали высотный ПВВ2;

f007 - блок датчиков угловых скоростей БДУС1,

f008 - блок датчиков угловых скоростей БДУС2;

f009 - прибор отключения нагревателя воздуха;

f010 - сигнализатор давления гидроаккумуляторов;

f011 - сигнализатор давления наддува.

f012 - участок поиска Земли;

f014 - участок коррекции от ПМВ;

f100 - пустой терм 1;

f101 - пустой терм 2;

f102 - пустой терм 3;

f103 - пустой терм 4;

f200 - пустой θ - терм.

Переменные логического пространства и их семантика:

$\alpha1$ - показания индикатора повторной выставки (ПВ);

$\alpha 2$ - признак отработки разгонного импульса КТД;

$\alpha 3$ - признак неработоспособности прибора ПВВ1;

$\alpha 4$ - признак неработоспособности прибора ПВВ2.

Термальное описание целевой задачи t.trs.

```
t1:=(α1=0)=>t41+(α1=1)=>t2
t2:=t3->t4
t3:=f001CHf100
t4:=f002CHt5
t5:=t6CHt7
t6:=f101->t8
t8:=(α2=0)=>t9+(α2=1)=>t10
t9:=t11CHt12
t11:=(α3=0)->f005+(α3=1)=>f200
t12:=(α4=0)->f006+(α4=1)=>f201
t10:=t13CHt91
t13:=f003CHf004
t91:=t111CHt112
t111:=(α3=0)=>f015+(α3=1)=>f210
t112:=(α4=0)=>f016+(α4=1)->f211
t7:=t14CHt15
t14:=f102->t16
t16:=f007CHf008
t15:=f103->t17
t17:=t18->t19
t18:=f009CHf010
t19:=((f011->f012)->f014)
endtxt
f005=f015
```

```
f006=f016
f200=f201
t4=t41
end
```

Функциональный базис задачи Ф={f001, f002, f003, f004, f005, f006, f007, f008, f009, f010, f011, f012, f014, f100, f101, f102, f103, f200, f201}.

Множество логических переменных α ={α1, α2, α3, α4}.

Функция выполнимости $F_1 : Ф \to 3^4$ (таблица 5.1).

Таблица 5.1

Базовый элемент	α1	α2	α3	α4	$L=F_1(Ф)$
f001	1	н	н	н	L_1
f002	н	н	н	н	L_2
f003	н	1	н	н	L_3
f004	н	1	н	н	L_4
f005	н	н	0	н	L_5
f006	н	н	н	0	L_6
f007	н	н	н	н	L_7
f008	н	н	н	н	L_8
f009	н	н	н	н	L_9
f010	н	н	н	н	L_{10}
f011	н	н	н	н	L_{11}
f012	н	н	н	н	L_{12}
f014	н	н	н	н	L_{14}
f100	1	н	н	н	L_{100}
f101	н	н	н	н	L_{101}
f102	н	н	н	н	L_{102}
f103	н	н	н	н	L_{103}

f200	н	н	1	н	L_{200}
f201	н	н	н	1	L_{201}

5.3 Множество условий и допустимых вариантов целевой задачи

Множество условий реализации целевой задачи (таблица5.2)

Таблица 5.2

Варианты реализации	Условие реализации			
	α1	α2	α3	α4
ℜ1	0	0	0	0
ℜ2	0	0	0	1
ℜ3	0	0	1	0
ℜ4	0	0	1	1
ℜ5	0	1	0	0
ℜ6	0	1	0	1
ℜ7	0	1	1	0
ℜ8	0	1	1	1
ℜ9	1	0	0	0
ℜ10	1	0	0	1
ℜ11	1	0	1	0
ℜ12	1	0	1	1
ℜ12	1	1	0	0
ℜ14	1	1	0	1
ℜ15	1	1	1	0

ℜ16	1	1	1	1

$\Re0=\{(\alpha_1=н)(\alpha_2=н)(\alpha_3=н)(\alpha_4=н)\}$ - условие безусловного включения терма.

Термальное описание вариантов целевой задачи.

$w_1(t,\Re1)$= (f002 CH (t6 CH t7))= (f002 CH ((f101→(f005 CH f006)) CH t7)),

где t7:=(f102→(f007 CH f008)) CH (f103→((f009 CH f010)→ ((f011 → f012) → f014))),

$w_2(t,\Re2)$= (f002 CH ((f101→(f005 CH f201)) CH t7)),

$w_3(t,\Re3)$= (f002 CH ((f101→(f200 CH f006)) CH t7)),

$w_4(t,\Re4)$= (f002 CH ((f101→(f200 CH f201)) CH t7)),

$w_5(t,\Re5)$= (f002 CH ((f101→((f003 CHf004) CH (f005 CH f006))) CH t7)),

$w_6(t,\Re6)$= (f002 CH ((f101→((f003 CHf004) CH (f005 CH f201))) CH t7)),

$w_7(t,\Re7)$= (f002 CH ((f101→((f003 CHf004) CH (f200 CH f006))) CH t7)),

$w_8(t,\Re8)$= (f002 CH ((f101→((f003 CHf004) CH (f200 CH f201))) CH t7)),

$w_9(t,\Re9)$= ((f001 CH f100) →(f002 CH ((f101→(f005 CH f006)) CH t7))),

$w_{10}(t,\Re10)$ =((f001 CH f100) →(f002 CH ((f101→(f005 CH f201)) CH t7))),

$w_{11}(t,\Re11)$= ((f001 CH f100) →(f002 CH ((f101→(f200 CH f006)) CH t7))),

$w_{12}(t,\Re12)$= ((f001 CH f100) →(f002 CH ((f101→(f200 CH f201)) CH t7))),

$w_{13}(t,\mathfrak{R}13)$= ((f001 CH f100) →(f002 CH ((f101→((f003 CH f004) CH (f005 CH f006))) CH t7)),

$w_{14}(t,\mathfrak{R}14)$= ((f001 CH f100) →(f002 CH ((f101→((f003 CH f004) CH (f005 CH f201))) CH t7)),

$w_{15}(t,\mathfrak{R}15)$= ((f001 CH f100) → (f002 CH ((f101→((f003 CH f004) CH (f200 CH f006))) CH t7)),

$w_{16}(t,\mathfrak{R}16)$= ((f001 CH f100) →(f002 CH ((f101→((f003 CH f004) CH (f200 CH f201))) CH t7)).

В основу приведения всех вариантов к единой временной оси положим принцип одновременного включения всех вариантов, являющийся частным случаем принципа выбора опорного терма. Время его включения есть t_0. Зададим длительности базовых элементов следующей таблицей (таблица 5.3).

Таблица 5.3

f001	f002	f003	f004	f005	f006	f007	f008	f009	f010
10	125	200	200	150	150	600	600	100	200
f011	f012	f014	f100	f101	f102	f103	f200	f201	
100	1800	5840	110	33	96	145	0	0	

На циклограмме (рис 5.1) рассмотрен случай первичного приведения КА в ориентированное положение (α1=1). На ней показано включение КДУ через 110с с момента включения питания. Через 33с с момента включения КДУ включаются построители вертикали (f005, f006) в зависимости от их признака работоспособности (α3, α4). При истинном значении признака отработки разгонного импульса КДУ (α2=1) включаются построители местной вертикали f003, f004. Через 96с с момента включения КДУ включаются датчики угловых скоростей (f007, f008). Через 145с с момента включения КДУ последовательно включаются сигнализатор давления гидроаккумуляторов, сигнализатор давления наддува, режим поиска Земли, по окончании которого начинается участок коррекции длительностью 5480с. При повторном ориентировании ($\alpha 1=0$) отсутствует этап задержки включения КДУ на 110с. В представленном алгоритме оба этапа совмещены на временной оси по моменту включения (t_0). Функциональные элементы f100, f101, f102, f103 определяют относительную задержку моментов включения элементов и сами не выполняются. На циклограмме их длительности изображены прямоугольниками.

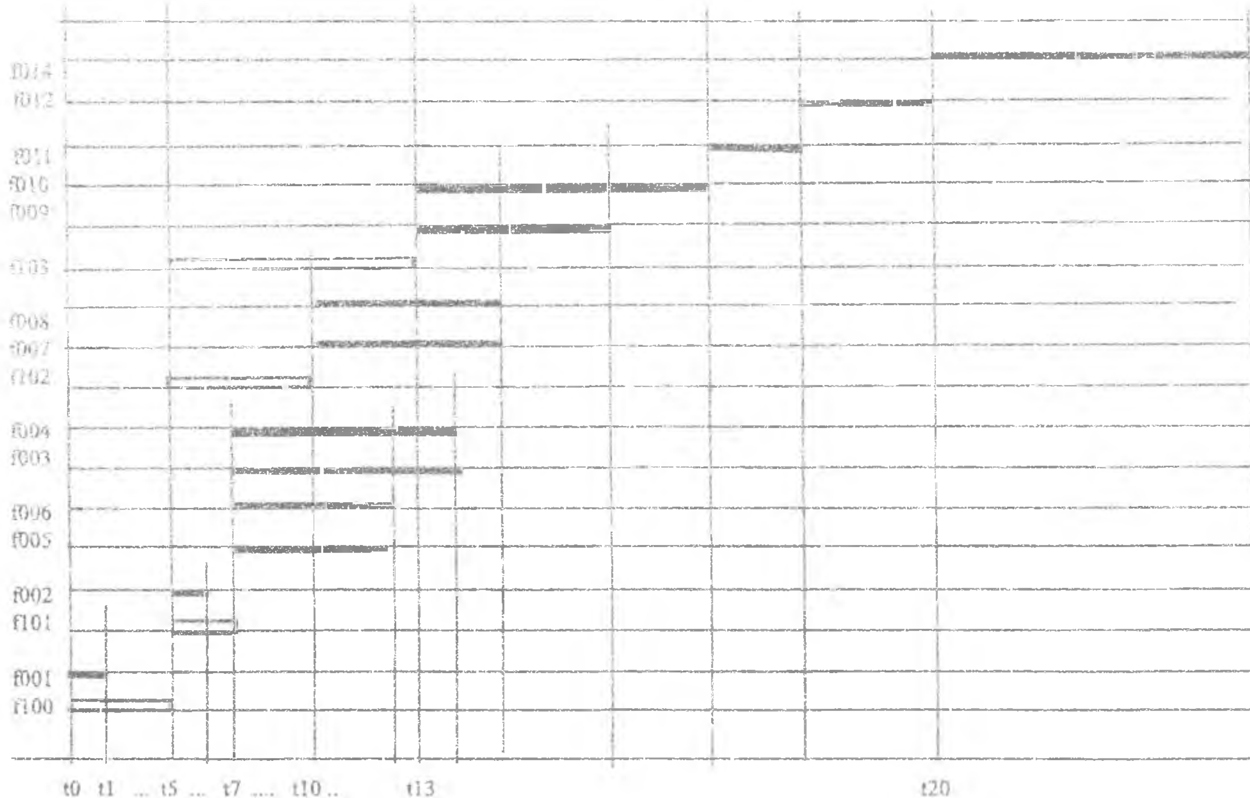

Рис. 5.1 Циклограмма задачи

5.4 Стандартная схема целевой задачи

Стандартную схему целевой задачи опишем множеством ее состояний. Состояние $S(T)(t_i)$ терма $T=T(\Phi)$ в i-й момент времени описывается совокупностью k наборов:

$$
\begin{aligned}
S(T)(t_i) = \{ & \langle \Phi_{j1}, L_{j1}, I_{j1} \rangle, \\
& \langle \Phi_{j2}, L_{j2}, I_{j2} \rangle, \\
& \ldots \\
& \langle \Phi_{jk}, L_{jk}, I_{jk} \rangle \},
\end{aligned}
$$

где к ≥ 1.

Тогда

$$
\begin{aligned}
S_0(t_0) = \{ & \langle f001, L_1=(\alpha 1=1), I_1 \rangle, \\
& \langle f100, L_{100}=(\alpha 1=1), I_{100} \rangle, \\
& \langle f002, L_2=(\alpha 1=0), I_2 \rangle, \\
& \langle f101, L_{101}=(\alpha 1=0), I_{101} \rangle, \\
& \langle f102, L_2=(\alpha 1=0), I_{102} \rangle, \\
& \langle f103, L_{103}=(\alpha 1=0), I_{103} \rangle \}.
\end{aligned}
$$

$$S_1(t_1) = \{ \langle f001, L_1, Out_1 \rangle \}.$$

$$S_2(t_2) = \{ \langle f002, L_2, Out_2 \rangle \}.$$

$$
\begin{aligned}
S_3(t_3) = \{ & \langle f101, L_{101}, Out_{101} \rangle, \\
& \langle f003, L_3=(\alpha 2=1), I_3 \rangle, \\
& \langle f004, L_4=(\alpha 2=1), I_4 \rangle, \\
& \langle f005, L_5=(\alpha 3=0), I_5 \rangle, \\
& \langle f006, L_6=(\alpha 4=0), I_6 \rangle, \\
& \langle f200, L_{200}=(\alpha 3=1), I_{200} \rangle, \\
& \langle f201, L_{201}=(\alpha 4=1), I_{201} \rangle \}.
\end{aligned}
$$

$$
\begin{aligned}
S_4(t_4) = \{ & \langle f102, L_{102}, Out_{102} \rangle, \\
& \langle f007, L_7=(\alpha 1=0), I_7 \rangle, \\
& \langle f008, L_8=(\alpha 1=0), I_8 \rangle \}.
\end{aligned}
$$

$S_5(t_5)=\{<f100, L_{100}, Out_{100}>,$

$<f002, L_2=(\alpha1=1), I_2>,$

$<f101, L_1=(\alpha1=1), I_{101}>,$

$<f102, L_2=(\alpha1=1), I_{102}>,$

$<f103, L_{103}=(\alpha1=1), I_{103}>\}.$

$S_6(t_6)=\{<f002, L_2=(\alpha1=1), Out_2>\}.$

$S_7(t_7)=\{<f101, L_{101}, Out_{101}>,$

$<f003, L_3=(\alpha2=1), I_3>,$

$<f004, L_4=(\alpha2=1), I_4>,$

$<f005, L_5=(\alpha3=0), I_5>,$

$<f006, L_6=(\alpha4=0), I_6>,$

$<f200, L_{200}=(\alpha3=1), I_{200}>,$

$<f201, L_{201}=(\alpha4=1), I_{201}>\}.$

$S_8(t_8)=\{<f103, L_{103}=(\alpha1=0), Out_{103}>,$

$<f009, L_9=(\alpha1=0), I_9>,$

$<f010, L_{10}=(\alpha1=0), I_{10}>\}.$

$S_9(t_9)=\{<f005, L_5, Out_5>,$

$<f006, L_6, Out_6>\}.$

$S_{10}(t_{10})=\{<f102, L_{102}, Out_{102}>,$

$<f007, L_7=(\alpha1=1), I_7>,$

$<f008, L_8=(\alpha1=1), I_8>\}.$

$S_{11}(t_{11})=\{<f003, L_3, Out_3>,$

$<f004, L_4, Out_4>\}$

$S_{12}(t_{12})=\{<f009, L_9, Out_9>\}.$

$S_{13}(t_{13})=\{<f103, L_{103}, Out_{103}>,$

$<f009, L_9=(\alpha1=1), I_9>,$

$<f010, L_{10}=(\alpha1=1), I_{10}>\}$

$S_{14}(t_{14})$ ={ <f010, L_{10}, Out_{10}>,

<f011, L_{11} =(α1=0), I_{11}>}.

$S_{15}(t_{15})$ ={<f009, L_9 , Out_9 >}

$S_{16}(t_{16})$ ={<f011, L_{11}, Out_{11}>,

<f012, L_{12} =(α1=0), I_{12}>}.

$S_{17}(t_{17})$ ={ <f010, L_{10}, Out_{10}>,

<f011, L_{11} =(α1=1), I_{11}>}.

$S_{18}(t_{18})$ ={<f011, L_{11}, Out_{11}>,

<f012, L_{12} =(α1=1), I_{12}>}.

$S_{19}(t_{19})$ ={<f012, L_{12}, Out_{12}>,

<f014, L_{14} =(α1=0), I_{14}>}.

$S_{20}(t_{20})$ ={<f012, L_{12}, Out_{12}>,

<f014, L_{14} =(α1=1), I_{14}>}.

$S_{21}(t_{21})$ ={<f014, L_{14}, Out_{14}>}.

$S_{22}(t_{22})$ ={<f014, L_{14}, Out_{14}>}.

5.5 Функция управления целевой задачей

Функция управления в последовательном методе строится для каждого момента времени.

Управляющий вектор для одного состояния терма, содержащего один базовый элемент <Φ_1, L_1, I_i> в i-й момент времени описывается набором

$$< v(t_i), u(\Phi_1,L_1), mp(\Phi_1) >,$$

где $v(t_i)$ =[t_i].

Для состояния S_0 функцию управления u(f001,f100,f002,f101, f102, f103, $\mathcal{L}_0$) ∈{ вкл. невкл } зададим таблицей 5.4.

$\mathcal{L}_0 = \Re_0 \cup L_1 \cup L_{100} \cup L_2 \cup L_{101} \cup L_{102} \cup L_{103}$.

Таблица 5.4

$\alpha 1$	u(f001,f100,f002,f101,f102,f103, $\angle_0$)
0	вкл(f002), вкл(f101), вкл(f102), вкл(f103)
1	вкл(f001), вкл(f100)

Для состояния S_1 функцию управления u(f001, $\angle_1$) $\in${откл, неоткл} зададим таблицей 5.5. $\angle_1 = \Re_0 \cup L_1$.

Таблица 5.5

$\alpha 1$	u(f001, $\angle_1$)
0	неоткл(f001)
1	откл(f001)

Для состояния S_2 функцию управления u(f002, $\angle_2$) $\in${откл, неоткл} зададим таблицей 5.6. $\angle_2 = \Re_0 \cup L_2$.

Таблица 5.6

$\alpha 1$	u(f002, $\angle_2$)
0	откл(f002)
1	неоткл(f002)

Для состояния S_3 функцию управления u(f003, f004, f005, f006, f101, f200, f201, $\angle_3$) $\in${ вкл, невкл }$\cup${откл, неоткл} зададим таблицей 5.7. $\angle_3 = \Re_0 \cup L_3 \cup L_4 \cup L_5 \cup L_6 \cup L_{101} \cup L_{200} \cup L_{201}$.

Таблица 5.7

$\alpha 1$	$\alpha 2$	$\alpha 3$	$\alpha 4$	u(f003, f004, f005, f006, f101, f200,f201, $\angle_3$)
0	0	0	0	откл(f101), вкл(f005), вкл(f006)
0	0	0	1	откл(f101), вкл(f005), вкл(f201)
0	0	1	0	откл(f101), вкл(f200), вкл(f006)

0	0	1	1	откл(f101), вкл(f200), вкл(f201)
0	1	0	0	откл(f101), вкл(f005), вкл(f006), вкл(f003), вкл(f004)
0	1	0	1	откл(f101), вкл(f005), вкл(f201), вкл(f003), вкл(f004)
0	1	1	0	откл(f101), вкл(f200), вкл(f006), вкл(f003), вкл(f004)
0	1	1	1	откл(f101), вкл(f200), вкл(f201), вкл(f003), вкл(f004)

Для состояния S_4 функцию управления u(f102, f007, f008, $\angle_4$) $\in$\{ вкл, невкл \}$\cup$\{откл, неоткл\} зададим таблицей 5.8.

$\angle_4 = \Re_0 \cup L_{102} \cup L_7 \cup L_8$.

Таблица 5.8

α1	u(f102, f007, f008, $\angle_4$)
0	вкл(f007), вкл(f008), откл(f102)
1	невкл(f007), невкл(f008), неоткл(f102)

Для состояния S_5 функцию управления u(f100, f002, f101, f102, f103, $\angle_5$) $\in$\{ вкл, невкл\}$\cup$\{откл, неоткл\} зададим таблицей 5.9.

$\angle_5 = \Re_0 \cup L_{100} \cup L_2 \cup L_{101} \cup L_{102} \cup L_{103}$.

Таблица 5.9

α1	u(f100, f002, f101, f102, f103, $\angle_5$)
0	невкл(f002), невкл(f101), невкл(f102), невкл(f103), неоткл(f100)
1	вкл(f002), вкл(f101), вкл(f102), вкл(f103), откл(f100)

Для состояния S_6 функцию отключения u(f002, $\angle_6$) ∈{ откл, неоткл } зададим таблицей 5.10. $\angle_6 = \Re_0 \cup L_2$.

Таблица 5.10

α1	u(f002, $\angle_6$)
0	неоткл(f002)
1	откл(f002)

Для состояния S_7 функцию управления u(f003, f004, f005, f006, f101, f200, f201, $\angle_7$) ∈{ вкл, невкл }∪{откл, неоткл} зададим таблицей 5.11. $\angle_7 = \Re_0 \cup L_3 \cup L_4 \cup L_5 \cup L_6 \cup L_{101} \cup L_{200} \cup L_{201}$.

Таблица 5.11

α1	α2	α3	α4	u(f003, f004, f005, f006, f101, f200, f201, $\angle_7$)
1	0	0	0	откл(f101), вкл(f005), вкл(f006)
1	0	0	1	откл(f101), вкл(f005), вкл(f201)
1	0	1	0	откл(f101), вкл(f200), вкл(f006)
1	0	1	1	откл(f101), вкл(f200), вкл(f201)
1	1	0	0	откл(f101), вкл(f005), вкл(f006), вкл(f003), вкл(f004)
1	1	0	1	откл(f101), вкл(f005), вкл(f201), вкл(f003), вкл(f004)
1	1	1	0	откл(f101), вкл(f200), вкл(f006), вкл(f003), вкл(f004)
1	1	1	1	откл(f101), вкл(f200), вкл(f201), вкл(f003), вкл(f004)

Для состояния S_8 функцию управления u(f103, f009, f010, $\angle_8$) ∈{ вкл, невкл }∪{откл, неоткл} зададим таблицей 5.12.

$\angle_8 = \Re_0 \cup L_{103} \cup L_9 \cup L_{10}$.

Таблица 5.12

α1	u(f103, f009, f010, $\angle_8$)
0	вкл(f009), вкл(f010), откл(f103)
1	невкл(f009), невкл(f010), неоткл(f103)

Для состояния S_9 функцию управления u(f005, f006, $\angle_9$) ∈ {откл, неоткл } зададим таблицей 5.13 $\angle_9 = \Re_0 \cup L_5 \cup L_6$.

Таблица 5.13

α1	α3	α4	u(f005, f006, $\angle_9$)
0	0	0	откл(f005), откл(f006)
0	0	1	неоткл(f005), неоткл(f006)
0	1	0	неоткл(f005), неоткл(f006)
0	1	1	неоткл(f005), неоткл(f006)

Для состояния S_{10} функцию управления u(f102, f007, f008, $\angle_{10}$) ∈{ вкл, невкл }∪{откл, неоткл} зададим таблицей 5.14.

$\angle_{10} = \Re_0 \cup L_{102} \cup L_7 \cup L_8$.

Таблица 5.14

α1	u(f102, f007, f008, $\angle_{10}$)
0	невкл(f007), невкл(f008), неоткл(f102)
1	вкл(f007), вкл(f008), откл(f102)

Для состояния S_{11} функцию управления u(f003, f004, $\angle_{11}$) ∈ {откл, неоткл } зададим таблицей 5.15. $\angle_{11} = \Re_0 \cup L_3 \cup L_4$.

Таблица 5.15

$\alpha1$	$\alpha2$	u(f003, f004, $\angle_{11}$)
0	0	неоткл(f003), неоткл(f004)
0	1	откл(f003), откл(f004)
1	0	неоткл(f003), неоткл(f004)
1	1	неоткл(f003), неоткл(f004)

Для состояния S_{12} функцию управления u(f009, $\angle_{12}$) $\in$\{ откл, неоткл \} зададим таблицей 5.16. $\angle_{12} = \Re_0 \cup L_9$.

Таблица 5.16

$\alpha1$	u(f009, $\angle_{12}$)
0	откл(f009)
1	неоткл(f009)

Для состояния S_{13} функцию управления u(f103, f009, f010, $\angle_{13}$) $\in$\{ вкл, невкл \}$\cup$\{откл, неоткл\} зададим таблицей 5.17.

$\angle_{13} = \Re_0 \cup L_{103} \cup L_9 \cup L_{10}$.

Таблица 5.17

$\alpha1$	u(f103, f009, f010, $\angle_{13}$)
0	невкл(f009),невкл(f010), неоткл(f103)
1	вкл(f009), вкл(f010), откл(f103)

Для состояния S_{14} функцию управления u(f010, f011, $\angle_{14}$) $\in$ \{ вкл, невкл \}$\cup$\{откл, неоткл\} зададим таблицей 5.18.

$\angle_{14} = \Re_0 \cup L_{10} \cup L_{11}$

Таблица 5.18

$\alpha1$	u(f010, f011, $\angle_{14}$)

0	откл(f010), вкл(f011)
1	неоткл(f010), невкл(f011)

Для состояния S_{15} функцию управления u(f009, $\angle_{15}$) ∈{ откл, неоткл } зададим таблицей 5.19. $\angle_{15} = \Re_0 \cup L_9$.

Таблица 5.19

α1	u(f009, $\angle_{15}$)
0	неоткл(f009)
1	откл(f009)

Для состояния S_{16} функцию управления u(f011, f012, $\angle_{16}$) ∈ { вкл, невкл }∪{откл, неоткл} зададим таблицей 5.20.

$\angle_{16} = \Re_0 \cup L_{11} \cup L_{12}$.

Таблица 5.20

α1	u(f011, f012, $\angle_{16}$)
0	откл(f011), вкл(f012)
1	неоткл(f011), невкл(f012)

Для состояния S_{17} функцию управления u(f010, f011, $\angle_{17}$) ∈ { вкл, невкл }∪{откл, неоткл} зададим таблицей 5.21.

$\angle_{17} = \Re_0 \cup L_{10} \cup L_{11}$.

Таблица 5.21

α1	u(f010, f011, $\angle_{17}$)
0	неоткл(f010), невкл(f011)
1	откл(f010), вкл(f011)

Для состояния S_{18} функцию управления u(f011, f012, $\angle_{18}$) ∈

{ вкл, невкл }∪{откл, неоткл} зададим таблицей 5.22.

$\angle_{18} = \Re_0 \cup L_{11} \cup L_{12}$

Таблица 5.22

α1	u(f011, f012, $\angle_{18}$)
0	неоткл(f011), невкл(f012)
1	откл(f011), вкл(f012)

Для состояния S_{19} функцию управления u(f012, f014, $\angle_{19}$) ∈ { вкл, невкл }∪{откл, неоткл} зададим таблицей 5.23.

$\angle_{19} = \Re_0 \cup L_{12} \cup L_{14}$

Таблица 5.23

α1	u(f012, f014, $\angle_{19}$)
0	откл(f012), вкл(f014)
1	неоткл(f012), невкл(f014)

Для состояния S_{20} функцию управления u(f012, f014, $\angle_{20}$) ∈ { вкл, невкл }∪{откл, неоткл} зададим таблицей 5.24.

$\angle_{20} = \Re_0 \cup L_{12} \cup L_{14}$

Таблица 5.24

α1	u(f012, f014, $\angle_{20}$)
0	неоткл(f012), невкл(f014)
1	откл(f012), вкл(f014)

Для состояния S_{21} функцию управления u(f014, $\angle_{21}$) ∈{откл, неоткл} зададим таблицей 5.25.

$\angle_{21} = \Re_0 \cup L_{14}$.

Таблица 5.25

α1	u(f014, $\angle_{21}$)
0	откл(f014)

1	неоткл(f014)

Для состояния S_{22} функцию управления u(f014, $\mathcal{L}_{22}$) ∈{откл, неоткл} зададим таблицей 5.26.

$\mathcal{L}_{22} = \Re_0 \cup L_{14}$.

Таблица 5.26

α1	u(f014, $\mathcal{L}_{22}$)
0	неоткл(f014)
1	откл(f014)

Функцию перехода из одного состояния в другое зададим следующим образом:

$t_i = \varphi(t_{i-1})$, i=1,2,. . . , 22

5.6 Текст асинхронного алгоритма. Последовательный метод

Для заданной целевой задачи в рамках последовательной схемы текст алгоритма управления будет состоять из текстов, написанных для каждого состояния и выполняемых в соответствующие асинхронные моменты времени t_i, i=0,1, . . . , 22. Построим тексты для всех состояний

Для состояния $S_0(t_0)$. u(f001,f100,f002,f101,f102,f103, $\mathcal{L}_0$):

ЕСЛИ (α1=1) ТО {ЧТЕНИЕ(f001), ВКЛ(f001), λ(f100)}

НАЧЕ { ЧТЕНИЕ(f002), ВКЛ(f002), λ(f101), λ(f102), λ(f103) }

Примечание. f100, f101, f102, f103- пустые невыполняемые термы.

Для состояния $S_1(t_1)$, u(f001, $\mathcal{L}_1$):

ЕСЛИ (α1=1) ТО {ОТКЛ(f001), ЗАПИСЬ(f001)}

Для состояния $S_2(t_2)$, u(f002, $\angle_2$):

ЕСЛИ (α1=1) ТО {ОТКЛ(f002) ЗАПИСЬ(f002)}.

Для состояния $S_3(t_3)$. u(f003, f004, f005, f006, f101. f200, f201, $\angle_3$):

ЕСЛИ (α1=1) ТО {λ}

ИНАЧЕ{λ(f101),

ЕСЛИ (α4=0) ТО {ЧТЕНИЕ(f006), ВКЛ(f006),

ЕСЛИ (α3=0) ТО {ЧТЕНИЕ(f005), ВКЛ(f005),

ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004) }}

ИНАЧЕ {λ(f200), ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004),ВКЛ(f004)}}}

ИНАЧЕ {λ(f201),

ЕСЛИ (α3=0) ТО {ЧТЕНИЕ(f005), ВКЛ(f005),

ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004) }}

ИНАЧЕ {λ(f200), ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004,)}}}}.

Примечание. f101, f200, f201- пустые, невыполняемые термы.

Для состояния $S_4(t_4)$. u(f102, f007, f008, $\angle_4$):

ЕСЛИ (α1=0) ТО {ЧТЕНИЕ(f007), ВКЛ(f007),

ЧТЕНИЕ(f008), ВКЛ(f008), λ(f102)}.

Остальные состояния $S_i(t_i)$, $i=5,...,22$ определяются аналогично.

Полный текст алгоритма асинхронного управления гермом t.trs в базисе $\hat{\sigma}$ на внутреннем языке системы с запуском в момент времени $t=t_0$ будет иметь следующий вид

1.Инициализация начальных данных.

2.ЕСЛИ ($\alpha 1=1$) ТО {ЧТЕНИЕ(f001), ВКЛ(f001), λ(f100)}

ИНАЧЕ {ЧТЕНИЕ(f002), ВКЛ(f002), λ(f101), λ(f102), λ(f103) }

3.УВИ($\tau=t_1-t_0$)

4.ЕСЛИ ($\alpha 1=1$) ТО {ОТКЛ(f001), ЗАПИСЬ(f001)}.

5.УВИ($\tau=t_2-t_1$)

6.ЕСЛИ ($\alpha 1=1$) ТО {ОТКЛ(f002), ЗАПИСЬ(f002)}.

7.УВИ($\tau=t_3-t_2$)

8.ЕСЛИ ($\alpha 1=1$) ТО {λ}

ИНАЧЕ{λ(f101),

ЕСЛИ ($\alpha 4=0$) ТО {ЧТЕНИЕ(f006), ВКЛ(f006),

ЕСЛИ ($\alpha 3=0$) ТО {ЧТЕНИЕ(f005), ВКЛ(f005),

ЕСЛИ ($\alpha 2=0$) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004) }}

ИНАЧЕ {λ(f200), ЕСЛИ ($\alpha 2=0$) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004),ВКЛ(f004)}}}

ИНАЧЕ {λ(f201),

ЕСЛИ ($\alpha 3=0$) ТО {ЧТЕНИЕ(f005), ВКЛ(f005),

ЕСЛИ ($\alpha 2=0$) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004) }}

ИНАЧЕ {λ(f200), ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004)}}}}.

9.УВИ($\tau=t_4-t_3$)

10.ЕСЛИ (α1=0) ТО {ЧТЕНИЕ(f007), ВКЛ(f007),

ЧТЕНИЕ(f008), ВКЛ(f008), λ(f102)}.

11.УВИ($\tau=t_5-t_4$)

12.ЕСЛИ (α1=0) ТО { λ}

ИНАЧЕ { ЧТЕНИЕ(f002), ВКЛ(f002), λ(f100), λ(f101), λ(f102), λ(f103) }.

13.УВИ($\tau=t_6-t_5$)

14.ЕСЛИ (α1=1) ТО {ОТКЛ(f002), ЗАПИСЬ(f002)}.

15.УВИ($\tau=t_7-t_6$)

16.ЕСЛИ (α1=0) ТО { λ}

ИНАЧЕ{λ(f101),

ЕСЛИ (α4=0) ТО {ЧТЕНИЕ(f006), ВКЛ(f006),

ЕСЛИ (α3=0) ТО {ЧТЕНИЕ(f005), ВКЛ(f005),

ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004) }}

ИНАЧЕ {λ(f200), ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004),ВКЛ(f004)}}}

ИНАЧЕ {λ(f201),

ЕСЛИ (α3=0) ТО {ЧТЕНИЕ(f005), ВКЛ(f005),

ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004) }}

ИНАЧЕ {λ(f200), ЕСЛИ (α2=0) ТО {λ}

ИНАЧЕ {ЧТЕНИЕ(f003), ВКЛ(f003),

ЧТЕНИЕ(f004), ВКЛ(f004)}}}}

17.УВИ($\tau=t_8-t_7$)

18.ЕСЛИ (α1=0) ТО {ЧТЕНИЕ(f009), ВКЛ(f009),

ЧТЕНИЕ(f010), ВКЛ(f010), λ(f103)}.

19.УВИ($\tau=t_9-t_8$)

20.ЕСЛИ (α1=0) ТО {ОТКЛ(f005), ЗАПИСЬ(f005),

ОТКЛ(f006), ЗАПИСЬ(f006)}.

21.УВИ($\tau=t_{10}-t_9$)

22.ЕСЛИ (α1=1) ТО {ЧТЕНИЕ(f007), ВКЛ(f007),

ЧТЕНИЕ(f008), ВКЛ(f008), λ(f102)}.

23.УВИ($\tau=t_{11}-t_{10}$)

24.ЕСЛИ (α1=0) ТО {ОТКЛ(f003), ЗАПИСЬ(f003),

ОТКЛ(f004), ЗАПИСЬ(f004)}.

25.УВИ($\tau=t_{12}-t_{11}$)

26.ЕСЛИ (α1=0) ТО {ОТКЛ(f009), ЗАПИСЬ(f009)}.

27.УВИ($\tau=t_{13}-t_{12}$)

28.ЕСЛИ (α1=1) ТО {ЧТЕНИЕ(f009), ВКЛ(f009),

ЧТЕНИЕ(f010), ВКЛ(f010), λ(f103)}.

29.УВИ($\tau=t_{14}-t_{13}$)

30.ЕСЛИ (α1=0) ТО {ОТКЛ(f010), ЗАПИСЬ(f010),

ЧТЕНИЕ(f011), ВКЛ(f011)}.

31.УВИ($\tau=t_{15}-t_{14}$)

32.ЕСЛИ (α1=1) ТО {ОТКЛ(f009). ЗАПИСЬ(f009)}.

33.УВИ($\tau=t_{16}-t_{15}$)

34.ЕСЛИ ($\alpha 1=0$) ТО {ОТКЛ(f011), ЗАПИСЬ(f011),

ЧТЕНИЕ(f012), ВКЛ(f012)}.

35.УВИ($\tau=t_{17}-t_{16}$)

36.ЕСЛИ ($\alpha 1=1$) ТО {ОТКЛ(f010), ЗАПИСЬ(f010),

ЧТЕНИЕ(f011), ВКЛ(f011)}.

37.УВИ($\tau=t_{18}-t_{17}$)

38.ЕСЛИ ($\alpha 1=1$) ТО {ОТКЛ(f011), ЗАПИСЬ(f011),

ЧТЕНИЕ(f012), ВКЛ(f012)}.

39.УВИ($\tau=t_{19}-t_{18}$)

40.ЕСЛИ ($\alpha 1=0$) ТО {ОТКЛ(f012), ЗАПИСЬ(f012),

ЧТЕНИЕ(f014), ВКЛ(f014)}.

41.УВИ($\tau=t_{20}-t_{19}$)

42.ЕСЛИ ($\alpha 1=1$) ТО {ОТКЛ(f012), ЗАПИСЬ(f012),

ЧТЕНИЕ(f014), ВКЛ(f014)}.

43.УВИ($\tau=t_{21}-t_{20}$)

44.ЕСЛИ ($\alpha 1=0$) ТО {ОТКЛ(f014), ЗАПИСЬ(f014)}.

45.УВИ($\tau=t_{22}-t_{21}$)

46.ЕСЛИ ($\alpha 1=1$) ТО {ОТКЛ(f014), ЗАПИСЬ(f014)}.

5.7 Временная диаграмма алгоритма управления

На рис. 5.2 изображен фрагмент временной диаграммы алгоритма асинхронного управления для случая первичного приведения КА в ориентированное положение ($\alpha 1=1$). Как видно из рисунка, в момент t_0 включаются элементы f001, f100. Через 110с включаются элементы f002, f101, f102, f103. По третьему входу через 33с с момента включения второго входа

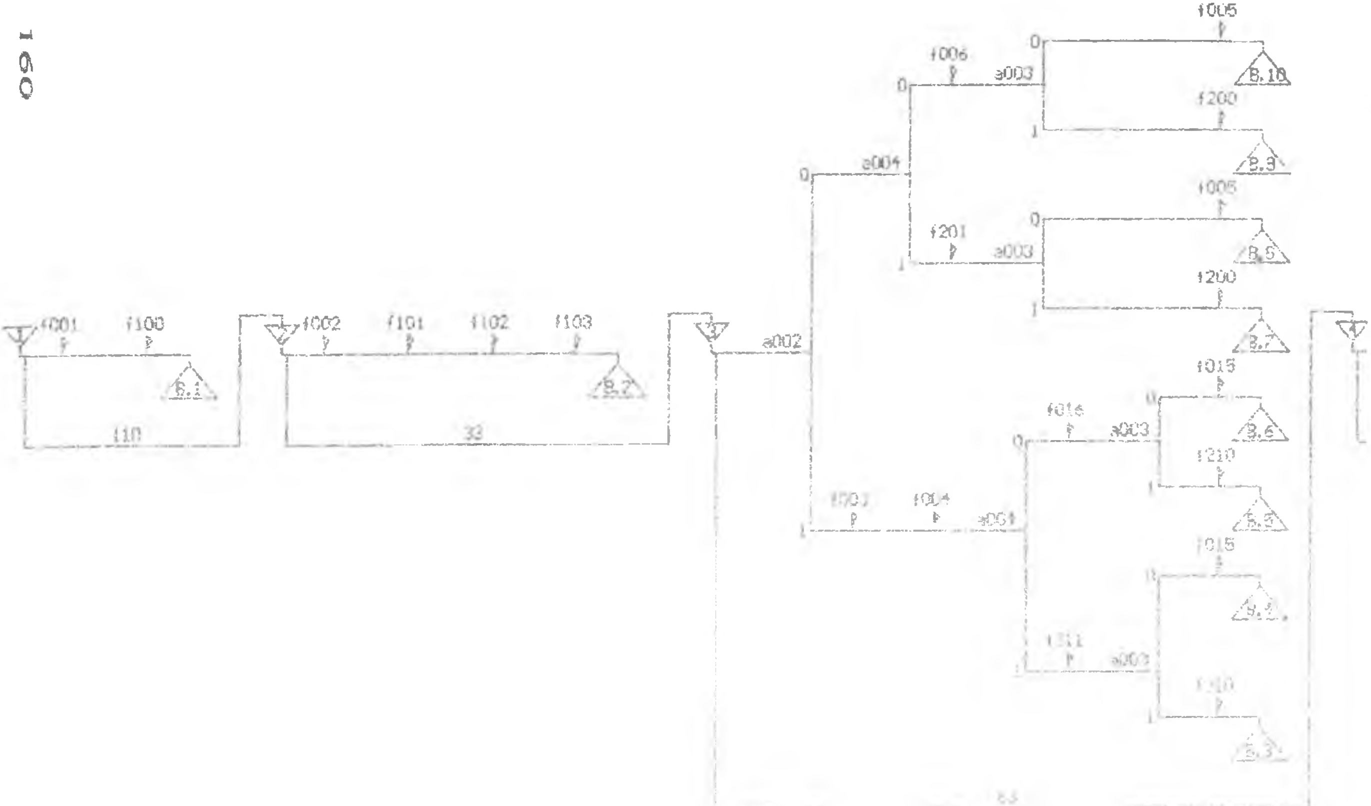

Рис. 5.2 Временная диаграмма алгоритма

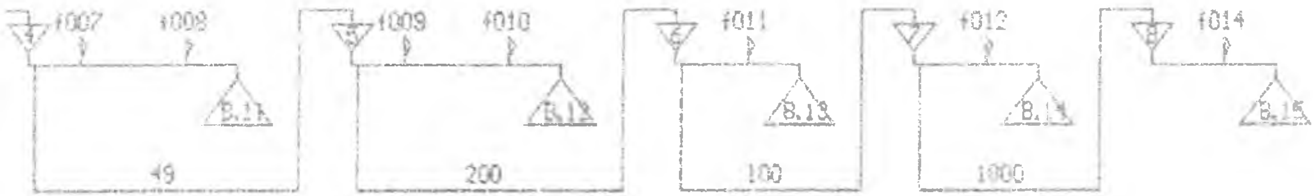

Рис. 5.2 Временная диаграмма алгоритма (продолжение)

выполняется проверка значений логических переменных α2, α4, α3 и включение соответствующих базовых элементов. Всего по алгоритму восемь входов.

5.8 Текст асинхронного алгоритма. Термальный метод

Для термального способа управления построим системную таблицу (таблица 5.27), содержащую описание терма в виде двоичного дерева операндов.

Таблица 5.27

№ зап	Призн. зап.	Код опер	Терм. опер	Время вкл.	Отн смещение	Длит терма	Указ 1 опер-да	Указ 2 опер-да	Указ 3 опер-да	Лог пер.
z_1*)	0	1	+	0	0	[illegible]	u(z_2)	[illegible]	—	α1
z_2	0	1	→	0	110	8195	u(z_3)	u(z_6)	u(z_1)	
z_3	0	1	CH	0	0	116	u(z_4)	u(z_5)	u(z_2)	
z_4	0	2	f001	0	0	10	—	—	u(z_3)	
z_5	0	2	f100	0	0	110	—	—	u(z_3)	
z_6	0	1	CH	110	0	8085	u(z_7)	u(z_8)	u(z_2)	
z_7	0	1	f002	110	0	125	—	—	u(z_6)	α1
z_8	0	1	CH	110	0	8033	u(z_9)	u(z_{26})	u(z_6)	
z_9	0	1	→	110	33	[illegible]	u(z_{10})	u(z_{11})	u(z_8)	
z_{10}	0	2	f101	110	0	33	—	—	u(z_9)	
z_{11}	0	1	+	143	0	200	u(z_{12})	[illegible]	u(z_9)	α2
z_{12}	0	1	CH	143	0	200	u(z_{13})	u(z_{16})	u(z_{11})	
z_{13}	0	1	CH	143	0	200	u(z_{14})	u(z_{15})	u(z_{12})	
z_{14}	0	2	f003	143	0	200	—	—	u(z_{13})	
z_{15}	0	2	f004	143	0	200	—	—	u(z_{13})	
z_{16}	0	1	CH	143	0	150	u(z_{17})	u(z_{20})	u(z_{12})	
z_{17}	0	1	+	143	0	150	u(z_{18})	u(z_{19})	u(z_{16})	α3
z_{18}	0	2	f200	143	0	0	—	—	u(z_{17})	
z_{19}	0	2	f005	143	0	150	—	—	u(z_{17})	
z_{20}	0	1	+	143	0	150	u(z_{21})	u(z_{22})	u(z_{16})	α4
z_{21}	0	2	f201	143	0	0	—	—	u(z_{20})	α4=0
z_{22}	0	2	f006	143	0	150	—	—	u(z_{20})	α4=1
z_{23}	0	1	CH	143	0	150	u(z_{24})	u(z_{25})	u(z_{11})	
z_{24}	0	1	+	143	0	150	u(z_{18})	u(z_{19})	u(z_{23})	α3
z_{25}	0	1	+	143	0	150	u(z_{21})	u(z_{22})	u(z_{23})	
z_{26}	0	1	CH	110	0	8085	u(z_{27})	u(z_{32})	u(z_8)	
z_{27}	0	1	→	110	96	696	u(z_{28})	u(z_{29})	u(z_{26})	
z_{28}	0	2	f102	110	0	96	—	—	u(z_{27})	
z_{29}	0	1	CH	206	0	606	u(z_{30})	u(z_{31})	u(z_{27})	
z_{30}	0	2	f007	206	0	600	—	—	u(z_{29})	
z_{31}	0	2	f008	206	0	600	—	—	u(z_{29})	
z_{32}	0	1	→	110	145	8085	u(z_{33})	u(z_{34})	u(z_{26})	
z_{33}	0	2	f103	110	0	135	—	—	u(z_{32})	
z_{34}	0	1	→	255	200	7940	u(z_{35})	u(z_{38})	u(z_{32})	
z_{35}	0	1	CH	255	0	300	u(z_{36})	u(z_{37})	u(z_{34})	
z_{36}	0	2	f009	255	0	100	—	—	u(z_{35})	
z_{37}	0	2	f010	255	0	200	—	—	u(z_{35})	

z_{38}	0	1	→	455	1900	7740	$u(z_{39})$	$u(z_{42})$	$u(z_{34})$	
z_{39}	0	1	→	455	100	1900	$u(z_{40})$	$u(z_{41})$	$u(z_{38})$	
z_{40}	0	3	1011	455	0	100	—	—	$u(z_{39})$	
z_{41}	0	2	1012	555	0	1800	—		$u(z_{39})$	
z_{42}	0	7	1014	2355	0	5640			$u(z_{38})$	

Полный текст алгоритма асинхронного управления термом t.trs в базисе $\sigma^{M}=\sigma'$ — ВЫЗОВ (имя процесса) на внутреннем языке системы с запуском в момент $t=t_0$ будет иметь следующий вид.

Для терма t.trs.

U(t.trs)(t_0): /Процесс1 (1)/

1. ЕСЛИ (α1=1)

ТО { ВЫЗОВ U(t2) (t_0)}

ИНАЧЕ {ВЫЗОВ U(t41) (t_0)}.

Для терма t2.

U(t2)(t_0): /Процесс1.1 (2)/

1.ВЫЗОВ U(t3)(t_0),

2.УВИ(τ(t3)),

3.ВЫЗОВ U(t4)(t_0 +τ(t3))

Для терма t3.

U(t3)(t_0): /Процесс1.1.1 (3)/

1.ВЫЗОВ U(f001)(t_0)

Для базового элемента f001.

U(f001)(t_0): /Процесс1.1.1.1 (4)/

1. ЧТЕНИЕ (f001)
2. ЕСЛИ (α(f001) =0) =TRUE

   ТО {ВКЛ(f001), α(f001):=(α(f001)+1)mod2}
3. УВИ (τ_1)
4. ЕСЛИ (α(f001) =1) =TRUE

ТО {ОТКЛ (f001), α(f001):=(α(f001)+1)mod2}

5. ЗАПИСЬ (f001)

Для терма t4.

U(t4) (t_0 +τ(t3)): /Процесс1.1.2 (5)/

1.ВЫЗОВ U(f002)(t_0+τ(t3)),

2.ВЫЗОВ U(t5)(t_0 +τ(t3)).

Для базового элемента f002

U(f002)(t_0 +τ(t3)): /Процесс1.1.2.1 (6)/

1.ЧТЕНИЕ (f002),

2.ЕСЛИ (α(f002) =0) =TRUE

ТО {ВКЛ(f002), α(f002):=(α(f002)+1)mod2},

3.УВИ (τ_2),

4.ЕСЛИ (α(f002) =1) =TRUE

ТО {ОТКЛ (f002), α(f002):=(α(f002)+1)mod2},

5.ЗАПИСЬ (f002).

Для терма t5.

U(t5) (t_0 +τ(t3)): /Процесс1.1.2.2 (7)/

1.ВЫЗОВ U(t6)(t_0 +τ(t3)),

2.ВЫЗОВ U(t7)(t_0 +τ(t3)).

Для терма t6.

U(t6)(t_0 +τ(t3)) /Процесс1.1.2.2.1 (8)/

1.ВЫЗОВ U(f101)(t_0 +τ(t3)),

2.УВИ(τ_{101}),

3.ВЫЗОВ U(t8)(t_0 +τ(t3)+τ_{101}).

Для терма t8.

U(t8)(t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.1 (9)/

1. ЕСЛИ (α2=0)

ТО { ВЫЗОВ U(t9) ($t_0 + \tau(t3) + \tau_{101}$)}

ИНАЧЕ {ВЫЗОВ U(t10) ($t_0 + \tau(t3) + \tau_{101}$)}.

Для терма t9.

U(t9) ($t_0 + \tau(t3) + \tau_{101}$): /Процесс1.1.2.2.1.1.1 (10)/

1.ВЫЗОВ U(t11) ($t_0 + \tau(t3) + \tau_{101}$),

2.ВЫЗОВ U(t2) ($t_0 + \tau(t3) + \tau_{101}$).

Для терма t11.

U(t11)($t_0 + \tau(t3) + \tau_{101}$): /Процесс1.1.2.2.1.1.1 (11)/

1. ЕСЛИ ($\alpha5=0$)

ТО { ВЫЗОВ U(f005) ($t_0 + \tau(t3) + \tau_{101}$)}

ИНАЧЕ {ВЫЗОВ U(f200) ($t_0 + \tau(t3) + \tau_{101}$)}.

Для базового элемента f005.

U(f005) ($t_0 + \tau(t3) + \tau_{101}$): /Процесс1.1.2.2.1.1.1.1 (12)/

1.ЧТЕНИЕ (f005),

2.ЕСЛИ (α(f005) =0) =TRUE

ТО {ВКЛ(f005) α(f005):=(α(f005)+1)mod2},

3.УВИ (τ_5),

4.ЕСЛИ (α(f005) = 1) =TRUE

ТО {ОТКЛ (f005). α(f005):=(α(f005)+1)mod2},

5.ЗАПИСЬ (f005)

Для терма t12.

U(t12)($t_0 + \tau(t3) + \tau_{101}$): /Процесс1.1.2.2.1.1.2 (13)/

1. ЕСЛИ ($\alpha4=0$)

ТО { ВЫЗОВ U(f006) ($t_0 + \tau(t3) + \tau_{101}$)}

ИНАЧЕ {ВЫЗОВ U(f201) ($t_0 + \tau(t3) + \tau_{101}$)}.

Для базового элемента f006.

U(f006) ($t_0 + \tau(t3) + \tau_{101}$): /Процесс1.1.2.2.1.1.2.1 (14)/

1.ЧТЕНИЕ (f006),

2.ЕСЛИ (α(f006) =0) =TRUE

ТО {ВКЛ(f006), α(f006):=(α(f006)+1)mod2},

3.УВИ (τ_6),

4.ЕСЛИ (α(f006) =1) =TRUE

ТО {ОТКЛ (f006), α(f006):=(α(f006)+1)mod2},

5.ЗАПИСЬ (f006).

Для терма t10.

U(t10) (t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2 (15)/

1.ВЫЗОВ U(t13) (t_0 +τ(t3)+τ_{101}),

2.ВЫЗОВ U(t91) (t_0 +τ(t3)+τ_{101}).

Для терма t13.

U(t13) (t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.1 (16)/

1.ВЫЗОВ U(f003) (t_0 +τ(t3)+τ_{101}),

2.ВЫЗОВ U(f004) (t_0 +τ(t3)+τ_{101}).

Для базового элемента f003.

U(f003) (t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.1.1 (17)/

1.ЧТЕНИЕ (f003),

2.ЕСЛИ (α(f003) =0) =TRUE

ТО {ВКЛ(f003), α(f003):=(α(f003)+1)mod2},

3.УВИ (τ_5),

4.ЕСЛИ (α(f003) =1) =TRUE

ТО {ОТКЛ (f003), α(f003):=(α(f003)+1)mod2},

5.ЗАПИСЬ (f003).

Для базового элемента f004.

U(f004) (t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.1.2 (18)/

1.ЧТЕНИЕ (f004),

2.ЕСЛИ (α(f004) =0) =TRUE

ТО {ВКЛ(f004), α(f004):=(α(f004)+1)mod2},

3.УВИ (τ_6),

4.ЕСЛИ (α(f004) =1) =TRUE

ТО {ОТКЛ (f004), α(f004):=(α(f004)+1)mod2},

5.ЗАПИСЬ (f004).

Для терма t91.

U(t91) (t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.2 (19)/

1.ВЫЗОВ U(t111) (t_0 +τ(t3)+τ_{101}),

2.ВЫЗОВ U(t112) (t_0 +τ(t3)+τ_{101}).

Для терма t111.

U(t111)(t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.2.1 (20)/

1. ЕСЛИ (α3=0)

ТО { ВЫЗОВ U(f015) (t_0 +τ(t3)+τ_{101})}

ИНАЧЕ {ВЫЗОВ U(f210) (t_0 +τ(t3)+τ_{101})}.

Для базового элемента f015.

U(f015) (t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.2.1.1 (21)/

1.ЧТЕНИЕ (f015),

2.ЕСЛИ (α(f015) =0) =TRUE

ТО {ВКЛ(f015), α(f015):=(α(f015)+1)mod2},

3.УВИ (τ_5).

4.ЕСЛИ (α(f015) =1) =TRUE

ТО {ОТКЛ (f015), α(f015):=(α(f015)+1)mod2},

5.ЗАПИСЬ (f015).

Для терма t112.

U(t112)(t_0 +τ(t3)+τ_{101}): /Процесс1.1.2.2.1.2.2.2 (22)/

1. ЕСЛИ (α4=0)

ТО { ВЫЗОВ U(f016) (t_0 +τ(t3)+$τ_{101}$)}

ИНАЧЕ {ВЫЗОВ U(f211) (t_0 +τ(t3)+$τ_{101}$)}

Для базового элемента f016.

U(f016) (t_0 +τ(t3)+$τ_{101}$). /Процесс1.1.2.2.1.2.2.2.1 (23)/

1.ЧТЕНИЕ (f016),

2.ЕСЛИ (α(f016) =0) =TRUE

ТО {ВКЛ(f016), α(f016):=(α(f016)+1)mod2},

3.УВИ ($τ_6$),

4.ЕСЛИ (α(f016) =1) =TRUE

ТО {ОТКЛ (f016), α(f016):=(α(f016)+1)mod2}

5.ЗАПИСЬ (f016).

Для терма t7.

U(t7) (t_0 +τ(t3)): /Процесс1.1.2.2.2 (24)/

1.ВЫЗОВ U(t14)(t_0 +τ(t3)),

2.ВЫЗОВ U(t15)(t_0 +τ(t3)).

Для терма t14.

U(t14)(t_0 +τ(t3)) /Процесс1.1.2.2.2.1 (25)/

1.ВЫЗОВ U(f102)(t_0 +τ(t3)),

2.УВИ($τ_{102}$),

3.ВЫЗОВ U(t16)(t_0 +τ(t3)+$τ_{102}$).

Для терма t16.

U(t16) (t_0 +τ(t3)+$τ_{102}$): /Процесс1.1.2.2.2.1.2 (26)/

1 ВЫЗОВ U(f007) (t_0 +τ(t3)+$τ_{102}$).

2.ВЫЗОВ U(f008) (t_0 +τ(t3)+$τ_{102}$)

Для базового элемента f007.

U(f007) (t_0 +τ(t3)+$τ_{101}$): /Процесс1.1.2.2.2.1.2.1 (27)/

1.ЧТЕНИЕ (f007),

2.ЕСЛИ (α(f007) =0) =TRUE

ТО {ВКЛ(f007), α(f007):=(α(f007)+1)mod2},

3.УВИ (τ_7),

4.ЕСЛИ (α(f007) =1) =TRUE

ТО {ОТКЛ (f007), α(f007):=(α(f007)+1)mod2},

5.ЗАПИСЬ (f007).

Для базового элемента f008.

U(f008) (t_0 +τ(t3)+τ_{101}) /Процесс1.1.2.2.2.1.2.2 (28)/

1.ЧТЕНИЕ (f008),

2.ЕСЛИ (α(f008) =0) =TRUE

ТО {ВКЛ(f008), α(f008):=(α(f008)+1)mod2},

3.УВИ (τ_8),

4.ЕСЛИ (α(f008) =1) =TRUE

ТО {ОТКЛ (f008), α(f008):=(α(f008)+1)mod2},

5.ЗАПИСЬ (f008).

Для терма t15.

U(t15)(t_0 +τ(t3)) /Процесс1.1.2.2.2.2 (29)/

1.ВЫЗОВ U(f103)(t_0 +τ(t3)),

2.УВИ(τ_{103}),

3.ВЫЗОВ U(t17)(t_0 +τ(t3)+τ_{103}).

Для терма t17.

U(t17)(t_0 +τ(t3)+τ_{103}) /Процесс1.1.2.2.2.2.2 (30)/

1.ВЫЗОВ U(t18) (t_0 +τ(t3)+τ_{103}),

2.УВИ(τ(t18)),

3.ВЫЗОВ U(t19)(t_0 +τ(t3)+τ_{103}+τ(t18)).

Для терма t18.

U(t18) (t_0 +τ(t3)+τ_{103}) /Процесс1.1.2.2.2.2.2.1 (31)/

1.ВЫЗОВ U(f009) (t_0 $+\tau(t3)+\tau_{103}$),

2.ВЫЗОВ U(f010) (t_0 $+\tau(t3)+\tau_{103}$).

Для базового элемента f009.

U(f009) (t_0 $+\tau(t3)+\tau_{103}$): /Процесс1.1.2.2.2.2.2.1.1 (32)/

1.ЧТЕНИЕ (f009),

2.ЕСЛИ (α(f009) =0) =TRUE

ТО {ВКЛ(f009), α(f009):=(α(f009)+1)mod2},

3.УВИ (τ_9),

4.ЕСЛИ (α(f009) =1) =TRUE

ТО {ОТКЛ (f009), α(f009):=(α(f009)+1)mod2},

5.ЗАПИСЬ (f009).

Для базового элемента f010.

U(f010) (t_0 $+\tau(t3)+\tau_{103}$): /Процесс1.1.2.2.2.2.2.1.2 (33)/

1.ЧТЕНИЕ (f010),

2.ЕСЛИ (α(f010) =0) =TRUE

ТО {ВКЛ(f010), α(f010):=(α(f010)+1)mod2},

3.УВИ (τ_{10}),

4.ЕСЛИ (α(f010) =1) =TRUE

ТО {ОТКЛ (f010), α(f010):=(α(f010)+1)mod2},

5.ЗАПИСЬ (f010).

Для терма t19.

U(t19)(t_0 $+\tau(t3)+\tau_{103}$ $+\tau(t18)$) /Процесс1.1.2.2.2.2.2.2 (34)/

1.ВЫЗОВ U(t181) (t_0 $+\tau(t3)+\tau_{103}$ $+\tau(t18)$),

2.УВИ(τ(t181)),

3.ВЫЗОВ U(f014)(t_0 $+\tau(t3)+\tau_{103}+\tau(t18)+\tau(t181)$).

Для терма t181.

U(t181)(t_0 $+\tau(t3)+\tau_{103}$ $+\tau(t18)$) /Процесс1.1.2.2.2.2.2.2.1 (35)/

1.ВЫЗОВ U(f011) ($t_0 + \tau(t3) + \tau_{103} + \tau(t18)$),

2.УВИ(τ_{11})),

3.ВЫЗОВ U(f012)($t_0 + \tau(t3) + \tau_{103} + \tau(t18) + \tau_{11}$).

Для базового элемента f011.

U(f011) ($t_0 + \tau(t3) + \tau_{103} + \tau(t18)$): /Процесс1.1.2.2.2.2.2.2.1.1 (36)/

1.ЧТЕНИЕ (f011),

2.ЕСЛИ (α(f011) =0) =TRUE

ТО {ВКЛ(f011), α(f011):=(α(f011)+1)mod2},

3.УВИ (τ_{11}),

4.ЕСЛИ (α(f011) =1) =TRUE

ТО {ОТКЛ (f011), α(f011):=(α(f011)+1)mod2},

5.ЗАПИСЬ (f011).

Для базового элемента f012.

U(f012) ($t_0 + \tau(t3) + \tau_{103} + \tau(t18) + \tau_{11}$): /Процесс1.1.2.2.2.2.2.2.1.2 (37)/

1.ЧТЕНИЕ (f012),

2.ЕСЛИ (α(f012) =0) =TRUE

ТО {ВКЛ(f012), α(f012):=(α(f012)+1)mod2},

3.УВИ (τ_{12}),

4.ЕСЛИ (α(f012) =1) =TRUE

ТО {ОТКЛ (f012), α(f012):=(α(f012)+1)mod2},

5.ЗАПИСЬ (f012).

Для базового элемента f014.

U(f014) ($t_0 + \tau(t3) + \tau_{103} + \tau(t18) + \tau_{11}$): /Процесс1.1.2.2.2.2.2.2.2 (38)/

1.ЧТЕНИЕ (f014),

2.ЕСЛИ (α(f014) =0) =TRUE

ТО {ВКЛ(f014), α(f014):=(α(f014)+1)mod2},

3.УВИ (τ_{14}),

4.ЕСЛИ (α(f014) =1) =TRUE

ТО {ОТКЛ (f014), α(f014):=(α(f014)+1)mod2},

5 ЗАПИСЬ (f014).

Всего 38 процессов.

6 Технология автоматизированного проектирования алгоритмов асинхронного управления

Данная глава содержит описание CASE-технологии для автоматизированного проектирования алгоритмов асинхронного управления систем, работающих в реальном времени.

Все большее и большее применение в мире находят различные методы автоматизации программирования [14,28] или CASE-технологии. Данные методики создаются в расчете на значительное сокращение затрат времени, труда, материальных затрат при разработке современного программного обеспечения, повышение надежности программного продукта, удобство его восприятия и модификации. Развитие сложных технических систем и их автоматизация привели к необходимости создания программных комплексов объемом в сотни тысяч и миллионы команд. Их написание возможно только при использовании тех или иных технологий автоматизированного программирования. Таким образом, разработка методов и средств автоматизации программирования является одной из основных проблем современного периода – периода перехода к новым информационным технологиям [2,5,16]. Для разработки такого программного средства, как Microsoft Excel for Windows, при использовании традиционной техники программирования необходимы трудозатраты, оцениваемые в пятьсот человеко-лет [22]. В случае же написания алгоритмов управления бортовой аппаратурой, исполняемых в реальном времени на борту летательного аппарата, предъявляются повышенные требования к надежности программного продукта. При возникновении ошибки в работе такого алгоритма возможны катастрофические последствия, неприемлемые потери.

Известно несколько основополагающих подходов к построению систем автоматизированного программирования. Приведем основные из них:

-создание систем, облегчающих в рамках традиционной парадигмы программирования проведение рутинных процедур (комплексные оболочки редактор -компилятор -компоновщик, интерфейсы универсальных текстовых редакторов с различными трансляторами и др. [13]);

-сведение программирования к построению программы из набора готовых модулей из библиотек;

-программирование с применением различных непроцедурных подходов (объектно -ориентированное, логическое, функциональное, ориентированное на стек и др) [19];

-методы, основанные на объединении предыдущих двух подходов, предлагающие при объектно-ориентированном программировании библиотеки классов (Turbo Vision, OWL в системе Borland C++ 3.x,4.x и т.д.);

-системы с графическим интерфейсом, заменяющим либо дополняющим текстовое описание, такие, например, как Visual Basic или Application Expert в Borland C++ 4.x),

-системы программирования, ориентированные на синтаксис [29] либо на запись программы в каком-то специфическом представлении (Е-язык,]|-схемы и др.) [17],

-системы, объединяющие в себе несколько вышеперечисленных подходов,

-методы разработки систем реального времени [21,23,25].

В то же время одной из наиболее интересных отраслей CASE - технологий является разработка средств для автоматизации нетрадиционного программного обеспечения, такого, как параллельные вычисления или микропрограммы для центральных процессоров ЭВМ [3,15,22]. Стандартом при этом становится введение графического интерфейса с пользователем. Подобные разработки ведутся в основном за рубежом [28,29].

Представляется очень важным ответ на вопрос, какая же из CASE-технологий является самой перспективной. Аналитические обзоры на эту тему [28], показывают, что применяемые подходы к процессам анализа и проектирования настолько различны, что их сравнение зачастую неправомочно.

В разделе описывается CASE-подход для автоматизированного проектирования алгоритмов асинхронного управления целевой задачей и, как следствие, технической системой, решаемой целевую задачу либо их совокупность. В частности, такой подход применим к проектированию алгоритмов асинхронного управления бортовой аппаратурой космических аппаратов.

6.1 Задача проектирования алгоритмов асинхронного управления

Определение. Под процессом проектирования (ПП) будем понимать преобразование исходного описания проектируемого объекта в его результирующее описание. Исходное описание алгоритма асинхронного управления включает в себя: термальное описание целевой задачи и его отображение на множество команд управления [11].

Исходное Результирующее

ПП (ААУ) : →

описание ААУ описание ААУ

Под результирующим описанием ААУ будем понимать модель ААУ, полученную из термального описания целевой задачи. Модель ААУ будем представлять парой объектов: временной диаграммой и блок-схемой.

Как видно из этого определения, проектирование алгоритма управления есть процесс преобразования термального описания целевой задачи в набор некоторых структур наперед заданного вида.

6.2 Задача автоматизированного проектирования алгоритмов асинхронного управления

Под задачей автоматизированного проектирования ААУ будем понимать некоторое отображение всех элементов процесса проектирования (ПП) на программы и данные, выполнение которых реализует процесс проектирования.

Решить задачу автоматизированного проектирования означает построить программный комплекс, являющийся образом процесса проектирования и позволяющий решать проектные задачи [4,12,16,24].

Исходное и результирующее описания ААУ будем отображать в определенного типа данные. Сам процесс проектирования отобразим в совокупность программ, реализующих технологические операции отдельных этапов проектирования ААУ.

Программный комплекс, содержащий средства предметно-ориентированного описания исходных и результирующих данных, вместе с заданным на нем порядком выполнения технологических операций назовем информационной технологией проектирования алгоритмов асинхронного управления [12].

6.3 Технология автоматизированного проектирования

В основе современных информационных технологий в качестве базовых понятий используются абстрактные типы данных и понятие объекта. При создании сложных программных комплексов, к которым относится и система автоматизированного проектирования управляющих алгоритмов, традиционные методы написания и отладки программ становятся малоэффективными. Описание данных и последовательности операторов над ними образует некий монотонный одноуровневый процесс, малоструктурированный, трудночитаемый и долго отлаживаемый. При этом разобраться в нем другому программисту зачастую очень трудно. Разработка сложных проектов традиционными методами равносильна проектированию с "нулевого" уровня, без учета существующих наработок. Новый подход предполагает большую структурированность проекта.

Переход к новой технологии определяется переходом от понятия "данное" к понятию "объект". Как известно, для формируемых в программе объектов определяется их тип и над ними производятся наперед заданные действия, совокупность которых определяется семантикой решаемой задачи. При этом производимые действия могут выполняться над любыми данными соответствующего типа. Как правило, данные со сложной структурой реализовывались некоторыми базовыми типами, действия над которыми должны быть расширены с учетом сложности структуры. Выполнение семантических преобразований базовыми средствами над данными со сложной структурой приводило к перегрузке соответствующих процедур. Чтобы развязать эти противоречия, введено понятие абстрактных данных. Его смысл заключается в создании средств, позволяющих для данных любой структуры определять собственный тип. Для вновь созданного типа определяются операции, выполняемые с данными только этого типа. Поименованное данное и его тип назовем объектом. Имя данного есть имя объекта. Совокупность таких объектов и операций по преобразованию данных в пределах одного типа назовем классом объектов.

Выделение некоторых наборов данных и операций над ними в отдельные классы существенно структурирует текст программы и называется инкапсуляцией. Свойство инкапсуляции класса скрывает от "внешнего" мира отдельные поля данных и позволяет обращаться к объекту только через операции класса. Создание сложных объектов через уже объявленные, простые, называется повторным использованием объектов или свойством наследования.

Исходя из определения класса, его объектам ставятся в соответствие операции, их обрабатывающие. В их описании заложены знания о типе объекта и его компонент. При формировании сложных объектов путем наследования классов в них войдут новые поля с новыми типами данных. Естественно, в новом классе под новые поля данных появятся новые операции. Не меняя имени функции, учтем в новом классе обработку дополнительных полей. Такое нагружение операции можно продолжить одновременно с построением новых классов - потомков и называется оно свойством полиморфизма операции.

Инкапсуляция, наследование, полиморфизм- три основных принципа объектно - ориентированной технологии программирования. Как следствие, выполнение таких принципов приводит к модульности, надежности и возможности многократного использования программ.

Наличие "хорошо" спроектированных классов или "правильно" выбранных объектов, что равносильно, позволяет легко формировать из них различные модули, выражающие законченную, функционально полную часть проекта. Их легко можно изменять, дополнять, удалять. "Хорошо" спроектированные объекты обладают минимальным набором связей между собой. Они связываются только тогда, когда это необходимо. Таким образом, сводится к минимуму вероятность возникновения побочных эффектов, повышается надежность программ. При изменении объекта не требуется удаление всего класса, достаточно создать его потомок с измененными свойствами. Это потребует отладки только дополнительных операций класса, без проверки всех частей проекта, в которых встречается этот класс.

Свойство многократного использования программ с целью "не начинать" проект с "нулевого" уровня также вытекает из "правильного" выбора классов и свойства модульности. Правильный выбор базовых классов существенно опирается на глубокое понимание смысла решаемой задачи. Он трудно формализуем и от него зависит структура всего проекта.

Опишем основные классы объектов рассматриваемого проекта средствами прагматического метода абстрагирования.

В спецификации программы прагматического метода абстрагирования дается явное определение типа данных, содержащее описание данных и операций. Каждая операция оформляется отдельной процедурой, в которой по умолчанию тип обрабатываемых данных предполагается известным, и процедура обрабатывает данные только этого типа.

6.3.1 Классы проекта

Класс Ф_Б - элементов функционального базиса. Объекты этого класса интерпретируются как физические приборы, работающие во времени, в логическом пространстве. Каждый объект задается парой: имя объекта $Ф_i$ (идентификатор), длительность его работы t_i (тип целый). Одни и те же объекты этого класса могут участвовать в решении различных целевых задач, выполняемых системой.

Операции класса: создать объект, удалить объект.

Класс L_P - логических переменных. Объекты этого класса являются значениями унарных, бинарных и т.д. предикатных функций, определяющих условия включения элементов функционального базиса либо их пар и т.д. в работу. Например, по умолчанию можно ввести множество логических переменных, каждая из которых определяет текущее состояние элемента функционального базиса

{ включен / не включен}.

Задание логической переменной:

$$
\text{имя}(\alpha_i) = \begin{cases} 1, & (\text{условие1}), \\ 0, & (\text{условие0}). \end{cases}
$$

Условие1/0 определяется на множестве произвольной природы, которое заранее известно исходя из семантики задачи.

Операции класса: создать объект, удалить объект.

Класс $\mathfrak{B}$ - логических выражений. Объект этого класса есть логическое выражение, полученное из значений логических переменных класса L_P с помощью операций булевой алгебры. Объекты этого класса определяют динамические термы. В частности, логическим выражением может быть и значение одной логической переменной.

Операции класса есть операции булевой алгебры на множестве логических переменных, определенных в классе L_P. Каждая логическая переменная α_i может принимать значение из множества { 0, 1, н}, где α_i = н означает независимость включения $Ф_i$ от значения данной логической переменной. Доопределим их для значения 'н' - независимое выполнение функционального элемента:

$(\alpha_i = 0) \cup (\alpha_i = н) = (\alpha_i = н)$,

$(\alpha_i = 0) \cap (x_i = н) = (x_i = 0)$.

Аналогично

$(\alpha_i = 1) \cup (\alpha_i = н) = (\alpha_i = н)$,

$(\alpha_i = 1) \cap (\alpha_i = н) = (x_i = 1)$}.

Класс ТЕРМ по определению есть строка символов. Символами в строке могут быть операнды, операции, разделители. Операндом является имя базового функционального элемента класса Ф_Б вида $Ф_i$, либо имя терма вида T_j. В качестве разделителей применяются скобки (,) {,},[,], знак :=, а также ',' и '.'.

ОПЕРАЦИИ

{ CH - бинарная операция совпадения по началу двух термов, ее результатом будет новый терм,

CK - бинарная операция совпадения по концу двух термов; ее результатом будет новый терм,

→ - бинарная операция следования одного терма за другим; ее результатом будет новый терм,

-унарная операция навешивания логического выражения на терм; ее результатом является новый динамический терм };

Пример. $T_1 := Ф_1$ CH $Ф_2$,

$T_2 := Ф_3$ CK T_1.

$T_1 := (\alpha=1) => Ф_4 + (\alpha=0) => T_2$.

Взаимосвязанная совокупность объектов этого класса, удовлетворяющих правилу подстановки, описывает одну целевую задачу. Входящее в нее подмножество функциональных элементов есть подмножество объектов класса Ф_Б. Входящее в нее подмножество логических выражений есть подмножество объектов класса Б.

Класс T_OGR - ограничений в целевой задаче. Один объект описывает запрещенные ситуации на одновременное выполнение некоторых пар функциональных элементов. Объектом этого класса является таблица, одна строка которой описывает запрещенную пару ($Ф_i$, $Ф_j$).

Операции класса: создать объект, удалить объект, добавить строку, удалить строку.

Класс D_T - целевых задач (дерево терма). Всякой взаимосвязанной совокупности термов класса ТЕРМ, описывающих одну целевую задачу, ставится в соответствие один объект этого класса. Фактически это внутрисистемное представление целевой задачи. По определению объект этого класса есть СПИСОК, состоящий из записей вида:

{ операция	типа символ,
имя логической переменной	типа символ,
указатель на запись 1-го операнда	типа ссылка,
указатель на запись 2-го операнда	типа ссылка,
длительность терма	типа целое };

Множество операций

$$\{ СН, СК, -> \} \cup \{+\} \cup \{Ф_i, i= 1,...,n\}$$

состоит из операций над термами на временной оси, т.е. { СН, СК, ->}, операции перехода по логическому выражению (+) и элементов функционального базиса {$Ф_i$, i= 1,...,n}. Если операция есть элемент первого множества, то данная запись описывает составной терм, а ее операнды1 и 2 являются указателями на новые записи и отсылают к их описанию. Для операции перехода (+) второе поле указывает имя логической переменой, по которой происходит условный переход. При этом по умолчанию переход по указателю на запись операнда1 происходит при истинном значении логической переменной и по указателю на запись операнда2 в противном случае. Если операция есть имя элемента функционального базиса, то поля с операндами и именем логической переменной пусты.

Примечание. Элементы функционального базиса {$Ф_i$, i= 1...,n} не являются операциями над термами, однако такая их интерпретация позволяет иметь единую информационную структуру представленыя целевой задачи.

ОПЕРАЦИИ

{ Параметризация по логическому условию – унарная операция, выделяющая из дерева терма поддерево, удовлетворяющее заданному логическому условию;

Выделение полного логического выражения для элемента функционального базиса}.

Класс TABL_WAR (таблица вариантов). Объектом этого класса является таблица вариантов выполнения целевой задачи в логическом пространстве. Одной целевой задаче ставится в соответствие один объект этого класса. Одна строка таблицы описывает один вариант и состоит из вектора значений всех логических переменных, участвующих в задаче. Число строк таблицы соответствует числу вариантов задачи.

Операции класса: создать объект, удалить объект, добавить строку, удалить строку.

Класс Ф_L динамических объектов. Один объект класса представляет собой таблицу, строкой которой является один динамический объект ($Ф_i$, α), задаваемый именем функционального элемента $Ф_i$ и логическим условием его включения ($\alpha_i \in \{ 0,1, н\}$, $i=1,2,...,n$). Два динамических объекта (две строки таблицы) отличаются друг от друга, если они отличаются хотя бы одной компонентой.

Операции класса: создать объект, удалить объект, добавить строку, удалить строку.

Класс WAR_R (варианты реализации). Каждый объект этого класса соответствует одному варианту реализации целевой задачи. Тогда одной целевой задаче соответствует столько объектов этого класса, сколько строк у соответствующего объекта класса TABL_WAR. Объект класса представляет собой СПИСОК, состоящий из записей вида:

{ операция типа целый,
операнд1 типа указатель на новую запись,
операнд2 типа указатель на новую запись}.

Здесь множество операций есть объединение

$$\{ СН, СК, -> \} \cup \{Ф_i, i= 1,...,n\}$$

операций над термами на временной оси, т.е. {СН, СК, ->}, и элементов функционального базиса $\{Ф_i, i= 1,...,n\}$. Если операция есть элемент первого множества, то данная запись описывает составной терм, а ее операнды1 и 2 являются указателями на новые записи и отсылают к их описанию. Если операция есть имя элемента функционального базиса, то поля с операндами пусты.

Операции класса: создать объект, удалить объект, сканировать объект, выделить подтерм с заданной операцией, выделить элементы функционального базиса, входящие в объект.

Класс WAR_RT (варианты реализации во времени). Объект этого класса есть расширение объекта класса WAR_R. Он представляет собой СПИСОК, состоящий из записей вида:

{ операция типа целый,
операнд1 типа указатель на новую запись,
операнд2 типа указатель на новую запись,
$t_{вкл}$ типа целое,
$t_{см}$ типа целое}.

Здесь $t_{вкл}$ - время включения терма, описанного в этой записи в собственной шкале;

$t_{см}$ - время смещения включения второго операнда относительно первого в собственной шкале.

Унарные операции класса:

-создать объект,

-удалить объект,

-сканировать объект,

-выделить подтерм с заданной операцией,

-выделить элементы функционального базиса, входящие в объект,

-рассчитать время включения всех базовых элементов, входящих в терм,

-определить длительность работы терма,

-выделить опорный базовый элемент на всех вариантах целевой задачи.

Бинарная операция:

-сравнить длительности двух вариантов одного терма.

n-арные операции:

-синхронизировать все варианты одного терма,

-выделить все состояния системы (все моменты включения элементов системы).

Класс WXOD. Один объект этого класса описывает одно состояние системы или один момент времени (параметризация целевой задачи по времени). В один объект входят все базовые элементы с их логическими условиями, включаемые в единый момент времени. Объект представляет собой СПИСОК, каждая запись которого имеет следующий смысл:

{ $t_{вкл}$ типа целое,

n типа целое,

u типа ссылка в список T_WXOD }.

Здесь $t_{вкл}$ - время включения,

n - число базовых элементов, включаемых одновременно в момент $t_{вкл}$ (в общем случае это элементы из разных вариантов),

u - ссылка в список T_WXOD, содержащий базовые включаемые элементы и их условия включения.

Список T_WXOD состоит из записей длины n. Одна запись состоит из двух полей, имеющих следующий смысл:

{ $Ф_i$ - имя базового функционального элемента, тип символьный,

α - логическое условие включения $Ф_i$, тип булевский }.

Операции класса: создать объект, удалить объект, выделить одно состояние.

Класс S_P_AL (системное представление алгоритма). Объектами этого класса являются управляющие алгоритмы, представленные в некотором внутреннем формате, позволяющем легко формировать выходные документы. Объект класса описывается следующими системными таблицами: ST7, ST8, ST9, ST10, ST11.

i-я строка таблицы ST7 описывает i-й вход. Между входами одного AAУ существует связь по управлению. Вызов на выполнение i-го входа может осуществляться либо с предыдущих входов, либо из других AAУ.

Структура i-й строки таблицы ST7:

{поле1 - номер входа алгоритма, из которого запускается данный i- й вход, тип поля - целый;

поле2 - число входов алгоритма, запускаемых данным i - м входом, тип поля - целый,

поле3 - указатель на строку таблицы ST8, начиная с которой приводится список номеров запускаемых входов, тип поля - ссылка;

поле4 - указатель на строку таблицы ST9, начиная с которой описывается алгоритм управления для i-го входа, тип поля - ссылка}.

Первые три поля описывают взаимодействие различных входов алгоритма по управлению.

Системная таблица ST8 образует списковую структуру, вход в которую производится по указателю из таблицы ST7 поле3.

Системная таблица ST9 состоит из последовательности списковых структур, каждая из которых описывает один вход и имеет структуру алгоритма управления. Вход в отдельный список производится по указателю из таблицы ST7 поле4.

Список для i-го входа ААУ состоит из записей фиксированной длины. Каждая запись содержит указатели для перехода к новым записям. Одна запись содержит семь полей, имеющих следующий смысл:

{поле1 - число функциональных элементов на описываемом линейном участке, тип поля целый; поле2 - указатель на строку таблицы ST10, начиная с которой приводится список имен функциональных элементов на описываемом линейном участке, тип поля - ссылка;

поле3 - признак ветвления данного линейного участка, тип поля - целый;

поле4 - указатель на строку таблицы ST9, являющуюся продолжением списка по ложному значению логической переменной, тип поля - ссылка;

поле5 - указатель на строку таблицы ST11, содержащей имя логической переменной и признак ее ложного значения, тип поля - ссылка;

поле6 - указатель на строку таблицы ST9, являющейся продолжением списка по истинному значению логической переменной, тип поля - ссылка; поле7 - указатель на строку таблицы ST11, содержащей имя логической переменной и ее значение}.

Системная таблица ST10 образует списковую структуру, вход в которую производится по указателю из таблицы ST9 поле2. Один список описывает все функциональные элементы, встречающиеся на одном линейном участке алгоритма. Одна запись списка описывает один функциональный элемент со всеми его реквизитами.

Элементом списковой структуры таблицы ST11 является одна запись, содержащая имя логической переменной и ее значение. Вход на элемент таблицы ST11 производится по указателю из таблицы ST9 поле5 и поле7.

Операции класса:

-создать объект,

-удалить объект,

-выделить i -й вход,

-выделить линейный участок i - го входа, удовлетворяющий условию $\alpha_i = 0$,

-выделить все входы, запускаемые i - м входом,

-выделить входы, запускающие i - й вход.

6.3.2 Технологические операции

Опишем основные технологические операции, выполняемые в процессе создания асинхронного алгоритма. Каждой технологической операции поставим в соответствие класс объектов либо процедуру, преобразующую данные одного класса в другой. Так как каждая технологическая операция обладает определенной функциональной завершенностью, то независимо от ее формы представления определим ее через входные и выходные данные и неформальное выполнение.

Все технологические операции можно разделить на две группы:

-формируемые пользователем. Это операции ввода основных объектов, исходных данных и ограничений в задаче, а также операции вывода выходных документов;

-выполняемые автоматически. Это операции по преобразованию данных при переходе от целевой задачи к ее функции управления и эквивалентные преобразования над различными формами представления алгоритма управления с целью его полного и однозначного построения.

6.3.2.1 Формирование каталога элементов функционального базиса (Технологическая операция ТО1)

Область входных данных: перечень функциональных элементов с указанием их идентификатора и длительности их выполнения.

Область выходных данных: каталог функциональных элементов КФ или множество объектов класса Ф_Б.

Данная технологическая операция ТО1 формирует каталог базовых элементов КФ и является операцией класса Ф_Б - создать объект:

ТО1: Ф_Б → Ф_Б.

Каждому элементу класса ставится в соответствие идентификатор, используемый при описании целевой задачи. Каталог функциональных элементов единствен. Его элементы полностью или частично могут участвовать в различных проектах. Под проектом будем понимать совокупность данных, участвующих в проектировании ААУ для одной целевой задачи.

6.3.2.2 Определение набора логических переменных (Технологическая операция ТО2)

Область входных данных: перечень логических переменных; каждая переменная задается идентификатором и своим определением:

$$\alpha_i = \begin{cases} 1, \text{если} < \text{условие}> = \text{TRUE}, \\ 0, \text{в противном случае} \end{cases}$$

Область выходных данных: каталог логических переменных α_i, $i=1,\ldots,n$ или множество объектов класса L_P.

Операция ТО2 формирует каталог логических переменных и относится к классу L_P - создать объект:

$$\text{TO2: L\_P} \rightarrow \text{L\_P}.$$

Логические переменные вводятся для определения динамических термов. Они могут навешиваться как на элементы функционального базиса, так и на их совокупности. Каждый набор определяется внутри проекта. В частности, набор может быть пустым.

< Условие > в определении логической переменной может быть объектом произвольной природы: временем включения функционального элемента, результатом сравнения вычисленных и эталонных данных, признаком взаимодействия двух функциональных элементов, например, включение элемента $Ф_2$ возможно только после отключения элемента $Ф_1$.

Каждому элементу класса ставится в соответствие идентификатор, используемый при описании целевой задачи.

Каталог логических переменных единствен. Его элементы полностью или частично могут участвовать в различных проектах.

6.3.2.3 Формирование термального описания целевой задачи (Технологическая операция ТО3)

Область входных данных: неформальное описание целевой задачи.

Область выходных данных: множество взаимосвязанных термов, удовлетворяющих правилу подстановки и описывающих одну целевую задачу.

Операция ТО3 формирует множество термов и относится к классу ТЕРМ - создать объект:

$$ТО3: ТЕРМ \rightarrow ТЕРМ.$$

Она позволяет пользователю, разработчику ААУ, описать целевую задачу по определенным синтаксическим правилам Т=Т(Ф,L).

6.3.2.4 Трансляция термального описания целевой задачи (Технологическая операция ТО4)

Область входных данных: множество взаимосвязанных термов, удовлетворяющих правилу подстановки и описывающих одну целевую задачу или множество объектов класса ТЕРМ.

Область выходных данных: системное представление целевой задачи или объект класса D_T.

Данная операция транслирует взаимосвязанную совокупность объектов класса ТЕРМ в объект класса D_T:

$$ТО4: ТЕРМ \rightarrow D\_T.$$

6.3.2.5 Определение ограничений в задаче (Технологическая операция ТО5)

Область входных данных: множество пар ($Ф_i$, $Ф_j$), удовлетворяющих условию неналожения.

Область выходных данных: системная таблица неналожений S_N или объект класса T_OGR.

Данная операция формирует бинарное отношение на множестве функциональных элементов:

$$ТО5: T\_OGR \subseteq (Ф\_Б)^2 ,$$

где Ф_Б - класс объектов функционального базиса

Множество ограничений вводится пользователем и описывает пары функциональных элементов, на одновременное выполнение которых наложен запрет. Множество ограничений может быть и пустым. Ограничения над элементами являются более сильными требованиями, чем термальные операции, т.е. если два элемента связаны термальной операцией и одновременно на них наложено ограничение, и эти два отношения не согласованы, то предпочтение в реализации отдается ограничению.

6.3.2.6 Построение функции выполнимости (Технологическая операция ТО6)

Область входных данных: объекты класса D_T.

Область выходных данных: системная таблица ST1, в которой каждому функциональному элементу $Ф_i$ ставится в соответствие условие его включения L_j или объект класса Ф_L.

Операция ТО6 для каждого функционального элемента, входящего в терм целевой задачи, выделяет полное условие его выполнимости. $ТО6: D\_T \rightarrow Ф\_L$.

6.3.2.7 Построение вариантов реализации УА в логическом пространстве (Технологическая операция ТО7)

Область входных данных: объекты класса D_T

Область выходных данных: системная таблица STL, содержащая множество всех вариантов реализации управляющего алгоритма или объект класса WAR_R.

Данная операция формирует все варианты реализации управляющего алгоритма в логическом пространстве.

$$TO7: D\_T \rightarrow WAR\_R.$$

Вариант реализации УА определяется набором значений логических переменных, входящих в терм. Два варианта реализации называются различными, если они отличаются значением хотя бы одной логической переменной. Фиксация одного варианта соответствует одной реализации УА. Операция выполняется автоматически сразу же после выполнения технологических операций ТО1, ТО2, ТО3, ТО4.

На объекте класса D_T она выделяет альтернативные условия реализации терма и выписывает соответствующие этим условиям варианты. Множество всех реализаций определяет допустимое поведение алгоритма управления в логическом пространстве.

6.3.2.8 Построение вариантов реализации УА во временном пространстве (Технологическая операция ТО8)

Область входных данных: объекты класса WAR_R, правила интерпретации функциональных элементов и операций над ними на временной оси.

Область выходных данных: объекты класса WAR_RT.

Данная операция каждому функциональному элементу задает время его включения относительно начала j-й реализации.

$$TO8: WAR\_T \to WAR\_RT$$

6.3.2.9 Синхронизация всех вариантов УА (Технологическая операция ТО9)

Область входных данных: объекты класса WAR_RT.

Область выходных данных: объекты класса WAR_RT.

Данная технологическая операция приводит все варианты к единой временной шкале и является операцией класса WAR_RT:

$$TO9: WAR\_RT \to WAR\_RT.$$

Операция ТО9 синхронизирует все реализации одного терма на временной оси, пересчитывает моменты включения элементов относительно начала терма.

6.3.2.10 Выделение всех состояний системы (F_4) (Технологическая операция ТО10)

Область входных данных: объекты класса WAR_RT.

Область выходных данных: последовательность моментов времени включения УА.

Операция ТО10 формирует последовательность моментов включения базовых элементов для всего терма:

$$TO10: WAR\_RT \to \{ t_i\},\ i = 1,2, ...,n.$$

Каждому моменту времени она ставит в соответствие подмножество включаемых в этот момент функциональных элементов из различных реализаций:

$$F_4(\{ ST4[j] \}) = ST5,$$

где $ST5(i) = < t_i, \{ \Phi_{iq}\} >.$

$\{ Ф_{iq} \}$ - множество функциональных элементов из различных реализаций, включаемых в i-й момент времени относительно начала работы терма.

6.3.2.11 Параметризация терма по времени (F_5) (Технологическая операция ТО11)

Область входных данных: объекты класса WAR_RT, объекты класса Ф_L, последовательность моментов включения $\{ t_i \}$, i = 1,2, ...,n.

Область выходных данных: объекты класса WXOD.

Данная технологическая операция для фиксированного момента времени выделяет подмножество функциональных элементов, принадлежащих разным вариантам и включаемых в один момент времени:

$$TO11: WAR\_RT \times Ф\_L \rightarrow WXOD.$$

Данная операция для функциональных элементов по их условиям выполнимости строит функцию управления для каждого момента времени:

$$F_5(ST5(i), ST1) = ST6(i), i=1,...,l,$$

где l- число моментов времени включения и/или отключения функциональных элементов;

ST6 - системная таблица, реализующая функцию управления для фиксированного момента времени; i-м элементом таблицы ST6(i) является пара множеств:

$< РФ_i, PL_j >$, где $РФ_i$ - подмножество функциональных элементов, включаемых одновременно, и таких, что каждый элемент удовлетворяет условию PL_j либо его части, $РФ_i \in \mathcal{B}(Ф)$, $\mathcal{B}(Ф)$ - множество всех подмножеств функциональных элементов, включаемых одновременно, $PL_j \in \mathcal{B}(\alpha)$, $\mathcal{B}(\alpha)$ -

булеан логических переменных α, зафиксированных в i-й момент времени.

6.3.2.12 Построение алгоритма управления для фиксированного момента времени (F_5) (Технологическая операция ТО12)

Область входных данных: системная таблица ST6, i-й элемент которой описывает функцию управления для i-го момента времени.

Область выходных данных: системные таблицы ST7, ST8, ST9, ST10, ST11.

ТО12: WXOD $\rightarrow$ S_P_AL.

Введем определение i-м входом ААУ назовем ААУ для i-го фиксированного момента времени.

i-я строка таблицы ST7 описывает i-й вход. Между входами одного ААУ существует связь по управлению. Вызов на выполнение i-го входа может осуществляться либо с предыдущих входов, либо из других ААУ.

i-я строка таблицы ST7 состоит из четырех полей. Первые три поля описывают взаимодействие различных входов алгоритма по управлению: первое поле - номер входа ААУ, из которого запускается данный вход; второе поле - число входов ААУ, запускаемых данным входом; третье поле - указатель на строку таблицы ST8, начиная с которой приводится список номеров запускаемых входов. Четвертое поле содержит указатель на строку таблицы ST9, начиная с которой описывается алгоритм управления для i-го входа.

Системная таблица ST8 образует списковую структуру, вход в которую производится по указателю из таблицы ST7 поля 3.

Системная таблица ST9 образует списковую структуру, сохраняющую структуру алгоритма управления. Вход в отдельный список производится по указателю из таблицы ST7 поле4. Список для i-го входа ААУ состоит из записей фиксированной длины. Каждая запись содержит указатели для перехода к новым записям. Одна запись содержит семь полей, имеющих следующий смысл. Поле1 - число функциональных элементов на описываемом линейном участке; поле2 - указатель на строку таблицы ST10, начиная с которой приводится список имен функциональных элементов на описываемом линейном участке; поле3 - признак ветвления данного линейного участка; поле4 - указатель на строку таблицы ST9, являющейся продолжением списка по ложному значению логической переменной; поле5 - указатель на строку таблицы ST11, содержащей имя логической переменной и признак ее ложного значения; поле6 - указатель на строку таблицы ST9, являющейся продолжением списка по истинному значению логической переменной; поле7 - указатель на строку таблицы ST11, содержащей имя логической переменной и признак ее истинного значения.

Системная таблица ST10 образует списковую структуру, вход в которую производится по указателю из таблицы ST9 поле2. Один список описывает все функциональные элементы, встречающиеся на одном линейном участке алгоритма. Одна запись списка описывает один функциональный элемент со всеми его реквизитами.

Элементом списковой структуры таблицы ST11 является одна запись, содержащая имя логической переменной и ее значение. Вход на элемент таблицы ST11 производится по указателю из таблицы ST9 поле5 и поле7.

Данная технологическая операция для фиксированного момента времени переводит функцию управления в соответствующий алгоритм

Системные таблицы ST7, ST8, ST9, ST10, ST11 полностью определяют алгоритм асинхронного управления.

6.3.3 Технологическая цепочка

Совокупность технологических операций, объединенных некоторым порядком их выполнения, образует технологическую цепочку автоматизированного проектирования УА.

Укрупненно система автоматизированного проектирования УА изображена на рис. 6.1. Предложенная методология реализована в программной системе "ГРАФКОНТ" [30 - 32].

Входной многоязыковый процессор системы "ГРАФКОНТ" определяет взаимодействие пользователя с системой. Он включает в себя технологические операции ввода пользователем исходных данных:

-процессор ввода элементов функционального базиса (F_f);

-процессор ввода и определения логических переменных (F_l);

-процессор ввода ограничений в задаче ($F_{огр}$);

-процессор ввода и трансляции описания целевой задачи (F_0).

Он реализован как двух проходный транслятор. На первом проходе проводится лексический и синтаксический анализы, формируется структура типа дерево терма. На структуре типа дерево (второй проход) проводится семантический анализ.

Взаимодействие процессоров показано на рис. 6.2. Первые три процессора независимы и могут вызываться пользователем на выполнение в любом порядке. Процессор ввода описания целевой задачи реализует контекстно- свободную грамматику и транслирует его в дерево терма D_T. Полученные при этом

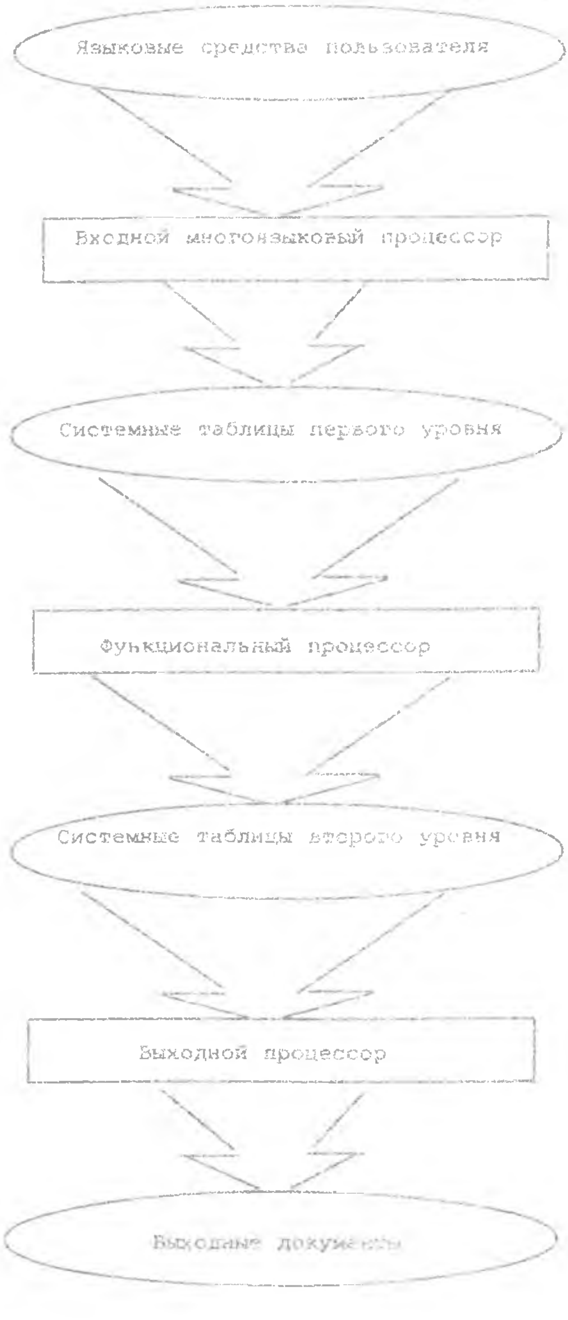

Рис.6.1Схема автоматизированного проектирования

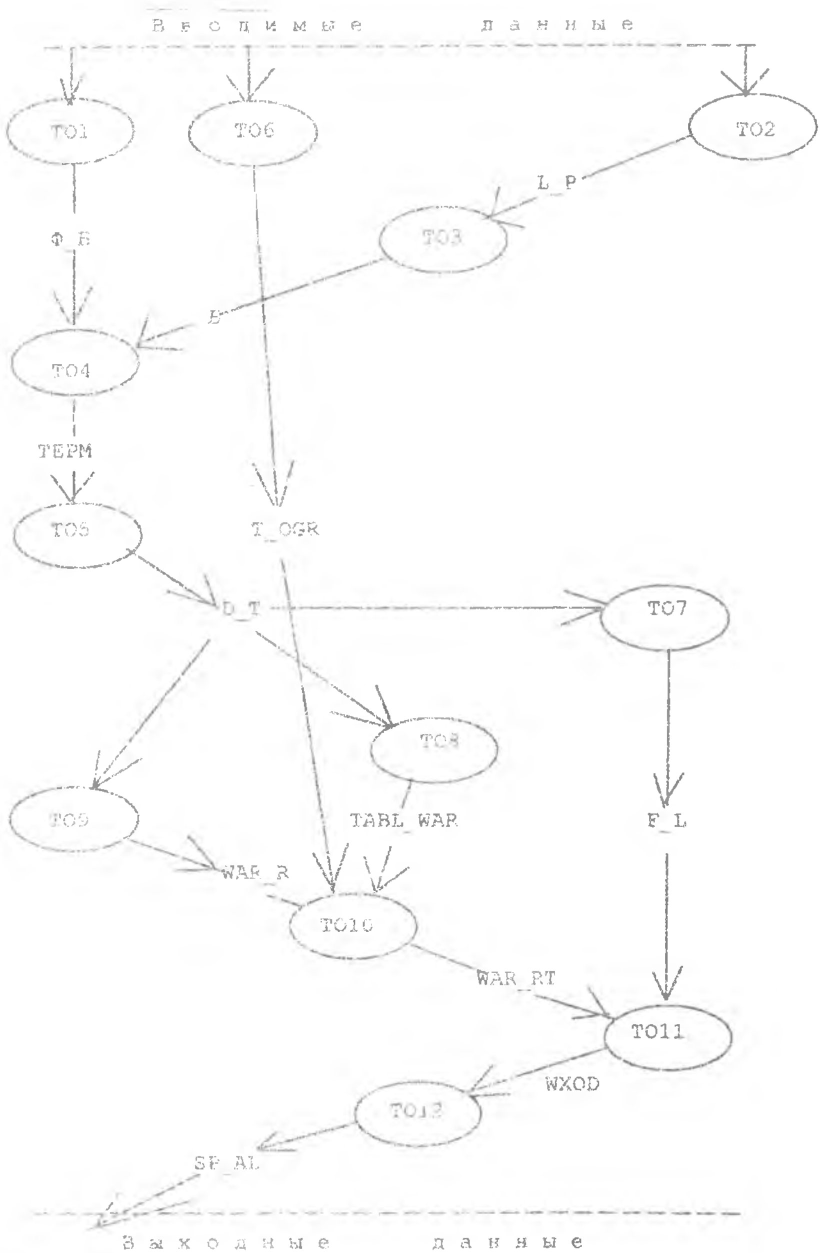

Рис.6.2 Схема взаимодействия технологических операций

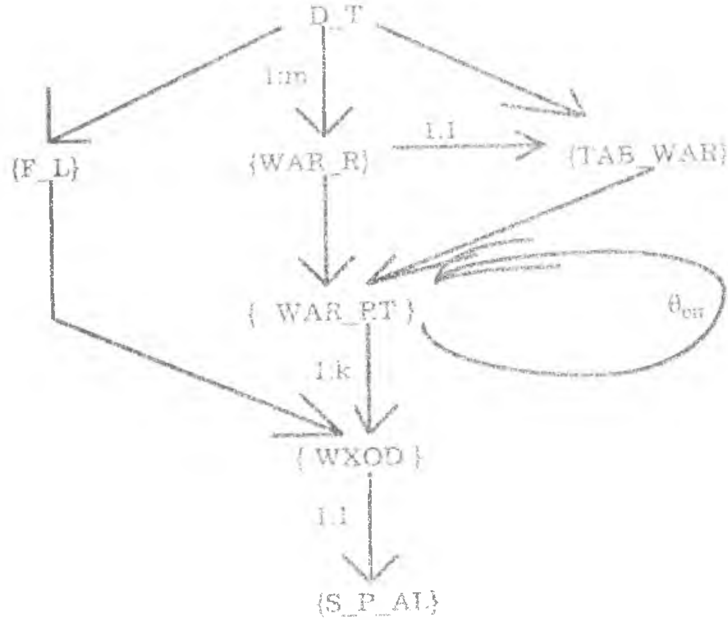

Рис.6.3. Порядок выполнения технологических операций функционального процессора

системные таблицы 1-го уровня: каталог функциональных задач, каталог логических переменных, таблица ограничений, дерево терма - являются исходными данными для функционального процессора.

Функциональный процессор состоит из технологических операций, выполняемых в автоматическом режиме, и изображен на рис. 6.3. По сути, это линейная последовательность технологических операций, образующих полный комплекс преобразований УА в его многовходовую модель, являющейся внутрисистемным представлением УА. Она является основой получения выходных документов: временной диаграммы УА и его блок- схемы.

Выходные документы УА получаются средствами выходного процессора в интерактивном режиме по запросу пользователя. Технологические операции выходного процессора: формирование временной диаграммы, формирование блок схемы - могут выполняться и в автоматическом режиме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Авдуевский В.С., Успенский Г.Р. Народнохозяйственные и научные космические комплексы -М.: Машиностроение, 1985. 416с.
2. Арсеньев Ю.Н., Журавлев В.М. Проектирование систем логического управления на микропроцессорных средствах - М.: Высшая школа, 1991. 319с.
3. Вальковский В.А. Распараллеливание алгоритмов и программ Структурный подход -М.: Радио и связь, 1989. 175с.
4. Вязгин В.А., Федоров В.В. Математические методы автоматизированного проектирования -М.: Высшая школа, 1989. 184с.
5. Гайсарян С.С., Ластовецкий А.Л. Алгебраическая модель неймановских языков программирования // Программирование. 1984. №6. С.12-22.
6. Горбатов В.А., и др. САПР систем логического управления - М.: Энергоатомиздат, 1988. 231с.
7. Ершов Ю.Л., Палютин Е.А. Математическая логика -М.: Наука, гл. ред. физ. -мат. литературы, 1979. 320с
8. Закревский А.Д. Алгоритмы синтеза дискретных автоматов - М.: Наука, 1971. 153с.
9. Калентьев А.А. Исчисление управляющих алгоритмов // "Математические методы и модели в САПР": Межвуз. сб. науч. трудов. -Самара,1991. С.4-10.
10. Калентьев А.А. Технология описания бортовой аппаратуры под задачу синтеза ее алгоритмов управления // Российская науч.- техн. конф. "Информационно-управляющие и вычислительные комплексы на основе новых технологий". -СПб, 1992.

11.Калентьев А.А., Тюгашев А.А. Построение текста управляющего алгоритма на базе многовходовой модели // 8-й Всероссийский семинар с международным участием по управлению движением и навигации летательных аппаратов, РАН, секция научного совета по проблемам управления движением и навигацией, -Самара, 1997.
12.Калентьев А.А. Построение закона функционирования системы, состоящей из n приборов // 8-й Всероссийский семинар с международным участием по управлению движением и навигации летательных аппаратов, РАН, секция научного совета по проблемам управления движением и навигацией, -Самара, 1997.
13.Касьянов В.Н., Поттосин И.В. Методы построения трансляторов -Новосибирск: Наука, Сиб. отд.-ние, 1986. 344с.
14.Козлов Д.И., Аншаков Г.П., Мостовой Я.А., Соллогуб А.В. Управление космическими аппаратами зондирования Земли. Компьютерные технологии -М.: Машиностроение, 1998. 368с.
15.Котов В.Е. Введение в теорию схем программ -Новосибирск: Наука, Сиб. отд.-ние, 1978. 256с.
16.Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов -М.: Энергоатомиздат, 1984. 192с.
17.Липаев В.В., Колин К.К., Серебровский Л.А. Математическое обеспечение управляющих ЦВМ -М.: Советское радио, 1972. 256с.
18.Мальцев А.И. Алгебраические системы -М.: Наука, гл. ред. физ.- мат. литературы, 1970. 392с.
19.Семантика языков программирования: Сб. статей, -М., Мир, 1980. 394с.

20.Соллогуб А.В., Аншаков Г.П., Данилов В.В. Космические аппараты систем зондирования поверхности Земли / Под ред. Д.И. Козлова -М.: Машиностроение, 1993. 367с.

21.Сушков Б.Г. Проблемы проектирования вычислительных систем реального времени// Теория и реализация систем реального времени: Сб. тр. ВЦ АН СССР, -М., 1984. С.4-12.

22.Трахтенгерц Э.А. Программное обеспечение параллельных процессов//АН СССР. Ин-т проблем управления -М.: Наука, 1987. 271с

23.Хоар Ч. Взаимодействующие последовательные процессы - М.: Мир, 1989. 264с.

24.Юдицкий С.А., Магерут В.З. Логическое управление дискретными процессами: модели, анализ, синтез -М.: Машиностроение, 1987. 175с.

25.Янг С. Дж. Алгоритмические языки реального времени -М.: Мир, 1985. 275с.

26.Янов Ю.И. О логических схемах алгоритмов// Проблемы кибернетики. -М.: Физматгиз, 1958. Вып.1. С.75-127.

27.Kalentjev A.A. Conversion term- based description of target task to time sequence // Processing of fourth Ukraine - Russia - China symposium on space science and technology, vol. II, 1996, p.844-846.

28.A parallel object- oriented language. Dessign and semantic foundation, America P.H.M., Rutten J.J.P // Reports Cent. Math. and Computer Sciences - 1989 pp 1- 44.

29.Priorities in process algebras / Cleaveland Rance, Henneysey Matthew // Inf. and Computers-1990-87-N1-2 pp 58-77.

30.Система «ГРАФКОНТ». Подсистема автоматизированной разработки алгоритмов Руководство программиста. // Отраслевой фонд алгоритмов и программ / Калентьев А.А.,

Мостовой Я.А., Никанорова Н.С., Николаев Ю.А., Ендуткина Л.И., Тюгашев А.А. -Самара: НТЦ «Наука», 1995. 20с.
31. Система «ГРАФКОНТ». Подсистема автоматизированной разработки программ. Руководство программиста. // Отраслевой фонд алгоритмов и программ / Калентьев А.А., Мостовой Я.А., Никанорова Н.С., Николаев Ю.А., Ендуткина Л.И., Тюгашев А.А. -Самара: НТЦ «Наука», 1996. 28с.
32. Система «ГРАФКОНТ». Подсистема сопровождения алгоритмов. Руководство программиста. // Отраслевой фонд алгоритмов и программ / Калентьев А.А., Мостовой Я.А., Никанорова Н.С., Николаев Ю.А., Ендуткина Л.И., Тюгашев А.А., -Самара: НТЦ «Наука», 1997. 23с.

Научное издание

Калентьев Анатолий Алексеевич

Автоматизированный синтез алгоритмов асинхронного управления техническими системами с множеством дискретных состояний

Корректор Т.К. Кретинина
Компьютерный набор и верстка Т.К. Кретинина

Лицензия ЛР № 020301 от 30.12.96г.

Подписано в печать 14.05.98г. Формат 60×84 $^1/_{16}$.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 12,09. Усл. кр.- отт. 12,21. Уч.- изд. л. 13,0.
Тираж 1000 экз. Заказ 85.

Самарский государственный аэрокосмический университет имени академика С.П. Королева 443086 Самара, Московское шоссе, 34.

ИПО Самарского государственного аэрокосмического университета имени академика С. П. Королева. 443001 Самара, ул. Молодогвардейская, 151.