

## БАЗОВЫЙ АЛГОРИТМ СИНТЕЗА ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР РЕАЛЬНОГО ВРЕМЕНИ

Clause is devoted to problems of synthesis of the computing structure, realizing the set algorithm, represents the important direction in computer science. In clause on the basis of the analysis of known methods of designing of complex technical systems perspective directions of researches for decisions of the given problem are offered. Conversation will be conducted about computing systems of the real time, working in a mode of real time – a mode of data processing at which interaction of computing systems with external in relation to them to processes are carried out during the moments defined in the speed of course of these processes. Let's note, that the requirement of realization of a mode of real time causes at designing such computing systems first of all maintenance of time parities between their components as bases of construction of such computing systems. Therefore the offered methods will be directed first of all on the decision of this problem while questions of spatial construction of computing systems of real time can become a theme of the further researches in this branch.

**Введение.** Проектирование сложных объектов, под которыми понимаются любые объекты, законы, функционирование которых допускает декомпозицию на структурные составляющие, предполагает предварительное создание моделей этих объектов с целью оценки их параметров и выбора оптимальных структур. Ответственная стадия – выбор структуры осуществляется в три этапа: анализ технического задания; анализ прототипов и на этой основе создание и анализ блок-схемы проектируемого объекта; выбор оптимального варианта.

Если третий этап решения проблемы может быть полностью формализован и на основе формализации автоматизирован, то на двух первых этапах, которые условно можно назвать системным анализом и синтезом структурной модели проектируемого объекта, используются два традиционных подхода – эвристическое и так называемое поисковое моделирование. К первому относятся неформализованные методы, основанные на анализе всех известных вариантов, выборе на основе опыта и интуиции нескольких, наиболее приемлемых для решаемой задачи вариантов и создании требуемой модели, по возможности, с использованием некоторых общих методов решения изобретательских задач. Ко второму – формализованные методы, основанные на строгом математическом аппарате и в силу этого поддающиеся алгоритмизации и компьютерной реализации. Такие методы на сегодняшний день разработаны для ограничения класса объектов, таких как комбинационные схемы и конечные автоматы, алгоритмы, вычислительные системы реального времени, некоторые классы электрических цепей.

**Алгоритм синтеза вычислительных систем реального времени.** Система реального времени – это система, которая должна ответить на внешнее воздействие в пределах конечного и указанного периодов времени. Корректность системы реального времени зависит не

только от логического результата вычисления, но также и от времени, в течение которого результаты получены.

Различают системы жесткого и мягкого реального времени. Системы жесткого реального времени – это такие системы, где абсолютно обязательно, чтобы система ответила в пределах указанного крайнего срока. Системы мягкого реального времени – это такие системы, где время ответа важно, но система будет все еще функционировать правильно, если крайние сроки иногда нарушаются. В системе жесткого или мягкого реального времени компьютер обычно соединяется непосредственно с некоторым физическим оборудованием и предназначен для контроля или управления действием того оборудования. Ключевой особенностью всех этих приложений является роль компьютера как компонента обработки информации в пределах большой технической системы.

Система реального времени как аппаратно-программный комплекс включает в себя датчики, регистрирующие события на объекте, модули ввода-вывода, преобразующие показания датчиков в цифровой вид, пригодный для обработки этих показаний на компьютере, и, наконец, компьютер с программой, реагирующей на события, происходящие на объекте [1].

Работа в режиме реального времени – это особая работа, когда необходимо отслеживать реально протекающий физический (химический) процесс и запаздывание с управлением может привести к катастрофе [2].

Наиболее трудными и дорогостоящими задачами в решении программ автоматизации технологических процессов являются проблемы обеспечения реального времени. Традиционный подход, ориентированный на использование унифицированных средств вычислительной техники, является неэффективным по причинам избыточности оборудования и временных затрат на реакцию системы на изменяющуюся внешнюю обстановку в случае экстремальных

ситуаций. Результатом анализа экстремальных ситуаций явился вывод о необходимости создания встроенных систем реального времени, к которым принято относить приборы и установки, в которых специализированный микропроцессор управляет технологическим процессом, обработкой информации, процессом принятия решений управления.

Типичными примерами являются станки с числовым программным управлением, электронные автоматические станции, системы военного назначения и др. Трудности реализации этих систем связаны с необходимостью обработки информации и принятия решений в реальном времени. Встроенные системы должны обладать такими техническими характеристиками, как малое энергопотребление, повышенная надежность, достаточная скорость реакции на внешнюю обстановку. Многообразие решаемых задач приводит к необходимости разработки формальных методов проектирования вычислительных структур специального назначения и создания на их основе средств автоматизации проектирования таких систем [3].

Известные системы автоматизированного проектирования позволяют решать задачи синтеза цифроаналоговых, логических структур, трассировки, компоновки, выпуска технической документации. В то же время отсутствуют средства автоматизации структурного и функционального проектирования, что связано, в первую очередь, с отсутствием до недавнего времени математических методов решения задачи функционального проектирования систем с нетрадиционной архитектурой.

Введение условия реализуемости пути алгоритма ориентирует проектируемые вычислительные структуры на функционирование в реальном масштабе времени, позволяет на одном из промежуточных этапов проектирования сделать вывод о реализуемости вычислительной структуры на выбранном функциональном устройстве, является основой для решения проблемы синхронизации проектируемой структуры и получения исходных данных для построения блоков управления, дает возможность решать ряд вопросов об использовании памяти, распределяет автоматически в процессе синтеза аппаратные и программные ресурсы системы между компонентами решаемой задачи.

Метод синтеза вычислительных структур реального времени позволит синтезировать вычислительные структуры в условиях жестких требований к временным параметрам алгоритмов на основе практически любых математических моделей, заданных аналитическим выражением во временной области или алгоритмом ее реализации, обеспечить высокие технические показатели проектируемых вычислительных структур, в первую очередь, загруженно-

сти и ускорения за счет распараллеливания и конвейеризации его путей, автоматизировать процесс синтеза в вычислительных системах реального времени [4].

Например, пусть существует набор функциональных устройств, позволяющий в совокупности выполнять все операции алгоритма, реализующего модель (1). В качестве таких устройств в зависимости от степени детализации алгоритма могут выступать элементы вычислительной техники (счетчики, сумматоры, триггеры) или функционально законченные блоки и устройства (арифметические устройства, функциональные модели, микропроцессоры, специализированные вычислительные устройства, процессоры ЭВМ, матричные процессоры, периферийные устройства ЭВМ и т. п.).

По принципу организации вычислительных процессов функциональные устройства подразделяются на простые и конвейерные.

Функциональные устройства называются простыми, если время выполнения на нем любой операции определено априори и никакая последующая операция не может начать выполняться раньше момента окончания предыдущей.

Функциональные устройства называются конвейерными, если время выполнения на нем любой операции может начать выполняться лишь через один такт после начала выполнения предыдущей.

По назначению функциональные устройства также делятся на две группы: основные, выполняющие операции алгоритма, и служебные, обеспечивающие эффективную работу основных функциональных устройств. Ко второй группе относятся каналы ввода-вывода, коммутаторы и т. п.

Отдельные ветви алгоритма могут быть реализованы с помощью конвейерных вычислителей. При этом речь идет об организации взаимодействия ряда устройств, а не об их внутренней структуре. Если в конвейерных вычислителях все операции являются безусловными, т. е. момент начала каждой операции не зависит от момента окончания другой, то конвейерные вычислители называются безусловными конвейерными вычислителями. Обычно при работе безусловного конвейерного вычислителя принято использовать синхронный режим работы [5].

Проблема автоматизации одного из важнейших этапов проектирования сложных систем (структурного и функционального проектирования) может быть решена путем создания программных средств, реализованных на базе основных положений теории синтеза вычислительных систем реального времени (ВСПВ) [4]. На основе данной теории предлагается алгоритм, который может в дальнейшем стать осно-

вой для создания автоматизированной системы функционального проектирования ВСПВ.

Вычислительная структура реального времени реализует вычислительный процесс общего вида

$$Y = F(X, Z, C), \quad (1)$$

где  $Y, X, Z$  – множества выходных, входных, промежуточных переменных;  $C$  – множество констант. При этом выполняется хотя бы одно из требований:

1. Скорости обработки переменных  $\varphi_i = X \cup Y \cup Z$ ,  $i = 1, I$  должны соответствовать заданным соотношениям

$$\tau_{\varphi_i} \leq \Delta t(\varphi_i), \quad (2)$$

где  $\tau_{\varphi_i}$  – время выполнения операции формирования переменной  $\varphi_i$ ;  $I$  – количество вершин графа алгоритма реализации процесса (1);  $\Delta t(\varphi_i)$  – фиксированный интервал времени (шаг дискретизации), который задает периодичность обработки данных для получения значения переменной  $\varphi_i$ , в частности, для входных данных это период поступления на соответствующий вход вычислительной структуры очередного дискретного отсчета переменной  $\varphi_i$ ; при этом  $\forall j, l \subset \{i\}$ :

$$\Delta t(\varphi_i) = \Delta t(\varphi_i)_{\min} k_n, \quad k_n = \overline{1, K},$$

$$\Delta t(\varphi_i) = \min_i \Delta t(\varphi_i),$$

где  $K$  – мощность множества интервалов  $\Delta t(\varphi_n)$ ,  $n \in \{i\}$ :

$$r, s \subset \{n\} : \Delta t(\varphi_r) \neq \Delta t(\varphi_s).$$

2. При известных моментах  $\{t(x_n)\}$ ,  $x_n \in X$  поступления на входы структуры ряда переменных  $x = \{x_n\}$  заданы моменты  $\{t'(Y_m)\}$  потребности в ряде выходных переменных  $y = \{y_m\}$ ,  $y_m \in Y$ . Это требование можно определить в виде соотношений

$$t_{\phi}(y_m) = t(x_n) \leq \Delta T(y_m, x_n), \quad (3)$$

где  $t_{\phi}(y_m)$  – момент завершения формирования переменной  $y_m$ ;  $\Delta T(y_m, x_n)$  – разностный интервал, по завершению которого после подачи на соответствующий вход переменной  $x_n$  должно быть получено значение  $y_m$ :

$$\Delta T(y_m, x_n) = t'(Y_m) - t(x_n).$$

3. Время реализации алгоритма вычислительной структурой  $T_a$  не должно превышать заданного значения:

$$T_a \leq T_{aufg}. \quad (4)$$

При случайном характере временных параметров вычислительного процесса в правой части выражений (2)–(4) используется значение нижней границы существования соответствующего переменного параметра времени.

Для реализации модели (1) необходимо наличие совокупности функциональных устройств,

программа работы которых связана с указанием последовательности операций над входными и промежуточными переменными. Под функциональным устройством будем понимать в зависимости от степени детализации алгоритма элементы вычислительной техники (сумматоры, вычислители, устройства для выполнения специальных функций) или функционально законченные блоки или устройства (унифицированные или специализированные процессоры, сопроцессоры, матричные процессоры и т. д.). Все функциональные устройства делятся на основные и служебные. Основными являются функциональные устройства, которые непосредственно выполняют операции алгоритма согласно математической модели вычислительного процесса (1). К служебным функциональным устройствам относятся устройства, не выполняющие непосредственно операций в соответствии с моделью (1), но наличие которых необходимо для нормальной организации вычислительного процесса в системе. Примерами таких устройств являются каналы связи, адаптеры, коммутаторы, мультиплексоры, преобразователи уровней сигналов, формирователи импульсных или других типов сигналов и т. д.

Отметим, что вычислительную структуру удобно представлять в виде графа, вершинам которого соответствуют функциональные устройства, а дугам – информационные связи. Тогда задача синтеза вычислительной структуры сводится к преобразованию исходного графа алгоритма вычислительного процесса в соответствии с некоторой последовательностью процедур с целью получения графа, который трансформируется в вычислительную структуру, удовлетворяющую определенным требованиям.

Модель вычислительного процесса (1) определяется в виде графа алгоритма  $G = (V, E)$ , множество вершин  $V$  которого соответствует множеству операций отдельных фрагментов алгоритма, множество  $E$  ребер – информационным связям.

Граф  $G = (V, E)$  может быть описан матрицей инцидентностей, которая имеет  $I$  вершин,  $J$  дуг и элементы  $a_{ij}$ .

Другим распространенным способом описания топологии графа является матрица смежности. Матрицей смежности называется квадратная матрица  $n \times n$  с элементами  $b_{ij}$ .

Алгоритм решения задачи предполагает выполнение следующих процедур:

1. Формирование графа вычислительного алгоритма.

2. Определение полных путей графа вычислительного алгоритма (путь, связывающий одну из вершин первого яруса с одной из конечных вершин графа).

3. Назначение уровней временной иерархии вершинам графа вычислительного алгоритма.

Уровнем временной иерархии подмножества вершин графа называется их приоритет, соответствующий требуемой скорости обработки данных каждой из вершин данного подмножества.

4. Назначение функциональных устройств вершинам графа базовой структуры («нагруженный» граф), формирование векторов назначения.

5. Формирование вектора реализации (элемент  $\tau_i$  равен времени выполнения операции, отождествленной с  $i$ -й вершиной функционального устройства, назначенного в соответствии с вектором назначения).

6. Формирование графа алгоритма с буферной памятью для систем, рассчитанных на использование средств с различными уровнями временной иерархии.

7. Создание усеченных путей уровней временной иерархии (усеченные пути формируются из полных путей в порядке снижения уровня временной иерархии посредством удаления вершин, включенных в уже сформированные пути).

8. Определение конвейеризируемых путей и ступеней конвейера.

9. Нахождение множества свертываемых вершин (для неконвейеризируемых путей).

10. Построение вычислительного графа алгоритма (добавление вершин при необходимости введения служебных устройств (мультиплексоры)).

11. Проверка реализуемости вычислительной структуры реального времени (выполнение условия реализуемости пути графа алгоритма в РМВ).

12. Формирование вектора временной развертки.

13. Формирование вектора требований к памяти для хранения промежуточных результатов (с регистровыми файлами, индексы положительных элементов данного вектора являются номерами дуг, требующих выполнения операции добавления вершины, которой назначается блок памяти).

14. Анализ реализуемости вычислительной структуры (наличие отрицательных элементов в данном векторе указывает на нереализуемость ВС на выбранном векторе назначения).

15. Разработка графа вычислительной структуры.

16. Построение функционала временной развертки (определяет программу взаимодейст-

вия функциональных устройств, составляющих вычислительную систему, на одном цикле ее функционирования).

17. Выбор структуры из множества работоспособных синтезированных структур, наилучшим образом удовлетворяющей заданным критериям оптимальности.

В результате реализации алгоритма, основанного на теории синтеза вычислительных систем реального времени, будут сформированы:

– граф вычислительной структуры, транспонируемый в структурную схему проектируемой системы, и спецификация его вершин;

– вектор назначения, который может стать основой создания принципиальных схем;

– функционал временной развертки, являющийся основой синхронизации проектируемой системы и создания средств управления данной системой; представление процессов формирования управляющих сигналов в виде функционала временной развертки позволит создавать управляющие сигналы для управления системами и технологическими процессами;

– такт вычислительной структуры, определяющий частоту следования тактовых импульсов.

**Заключение.** На базе предложенного алгоритма может быть создан пакет прикладных программ, использование которого позволит автоматизировать процесс функционального проектирования сложных технических систем и их компонент.

## Литература

1. Кобайло, А. С. Основы теории синтеза вычислительных структур реального времени / А. С. Кобайло. – Минск: БГУИР, 2001.

2. Складаров, В. А. Синтез автоматов на матричных БИС / В. А. Складаров. – Минск: Наука и техника, 1984.

3. Анкудинов, Г. И. Синтез структуры сложных объектов: Логико-комбинаторный подход / Г. И. Анкудинов. – Л.: Изд-во Ленинград. ун-та, 1986.

4. Кобайло, А. С. Единичные функции для математических моделей вычислительных процессов / А. С. Кобайло; РСНР. – Минск, 1997. – Деп. в Бел ИСА.

5. Шпаковский, Г. И. Архитектура параллельных ЭВМ / Г. И. Шпаковский. – Изд-во «Университетское», 1989.