

УДК 681.324

**Ю.К. Апраксин, проф., канд. техн. наук;**

**Т.В. Волкова, доцент, канд. техн. наук**

*Севастопольский национальный технический университет*

*Студгородок, г.Севастополь, Украина, 99053*

*E-mail: tatyana\_volkova.ukr.net*

## **АППАРАТНАЯ РЕАЛИЗАЦИЯ ПРОТОКОЛОВ РАСПРЕДЕЛЕННЫХ ТЕХНИЧЕСКИХ СИСТЕМ**

*Предлагается методика преобразования спецификаций протокольных автоматов в аппаратную среду функционирования децентрализованной системы управления распределенными техническими системами.*

Применение конечноавтоматной модели для построения формальных спецификаций протоколов распределенных систем имеет много положительных моментов, главный из которых — достаточно мощная математическая и алгоритмическая поддержка процессов анализа и синтеза специфицированных объектов [1-3]. Однако процесс технической реализации классической модели конечного автомата в конкретную физическую среду наталкивается на ряд трудностей. Основная проблема заключается в том, что эта модель игнорирует понятие физического времени [1], в связи с чем при каждой конкретной реализации протокола возникает необходимость существенного изменения исходной спецификации протокола. Кроме того динамика изменения состояний реального протокола связана как с внешней, так и с внутренней синхронизацией, что приводит к неоднозначности понятия «состояние».

Предлагается единая методика реализации протоколов в аппаратную среду функционирования, основанная на двухуровневой интерпретации конечноавтоматной модели.

Протокол распределенной технической системы обычно определяется как правило взаимодействия некоторых выделенных протекающих в системе процессов. Каждый такой процесс будем называть протокольным объектом (ПО). Считаем также, что взаимодействующие между собой ПО и среда взаимодействия образуют систему протокольного взаимодействия (СПВ). Взаимодействие в СПВ осуществляется путем обмена сообщениями между ПО. Отличительной особенностью распределенных технических систем (РТС) от других сложных систем является непредсказуемость зарождения, последовательности и времени пребывания каждого сообщения в системе [1]. Поэтому поведение каждого ПО можно представить только в так называемом нерегулярном событийном времени, т.е. время отсчитывается лишь в моменты, когда в ПО происходят события, изменяющие его состояние. Смена состояний ПО осуществляется под воздействием ряда внутренних для конкретной СПВ событий, причем каждое такое событие опре-

деляется как состоянием самого ПО, так и состоянием других ПО, взаимодействующих с ним.

Каждое состояние ПО с точки зрения процедуры взаимодействия может быть отнесено к одному из двух классов (внешние и внутренние состояния). К классу внешних состояний отнесем состояния, которые определяют динамику всей СПВ. Они обязательно должны быть определены общей спецификацией СПВ. Другими словами, взаимодействие между ПО может быть осуществлено только тогда, когда они находятся во внешних состояниях. К внутренним состояниям ПО причислим состояния, которые никак не сказываются на функционировании других ПО. Эти состояния важны лишь с точки зрения реализации собственных управленческих функций.

Значительное функциональное разнообразие ПО, высокие требования к качеству их функционирования и времени проектирования требуют наличия автоматических процедур реализации ПО в ту или иную среду функционирования. Это возможно только при наличии формальных средств спецификации ПО, способных адекватно описать особенности СПВ.

В предположении аппаратной реализации ПО (доведения абстрактной модели ПО до ее конкретного воплощения в среду функционирования в виде цифрового автомата), наиболее адекватной реальности с точки зрения авторов является конечноавтоматная модель, а средством формального описания ПО – язык регулярных выражений [3]. Предлагаемая спецификация ПО позволяет получать компактные формальные описания без перечисления конкретных состояний автомата. Чтобы спецификация ПО отражала не только динамику функционирования СПВ, но и поведение конкретного ПО при выработке управленческих решений (внешние и внутренние состояния), в качестве алфавита языка спецификаций предлагается использовать алфавит  $Z = \{z_k | z_k = \langle \langle EC_k \rangle, \langle LA_k \rangle \rangle\}$ . Каждая буква алфавита идентифицирует событие  $z_k$ , наступление которого определяется предикатом условий  $EC_k$ , а  $LA_k$  характеризует внутреннюю динамику ПО в случае наступления этого события.

В общем случае цифровой автомат, являющийся реализацией ПО (протокольный автомат – ПА), в соответствии со структурной теорией автоматов рассматривается как система двух взаимодействующих автоматов: управляющего (УА) и операционного (ОА). Взаимосвязь УА и ОА в процессе реализации возможных для ПА функций изображена на рисунке 1. Управляющий автомат обеспе-

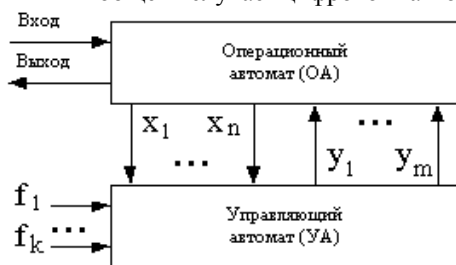


Рисунок 1 - Взаимодействие управляющего и обрабатывающего блоков в цифровом автомате

чивает выработку необходимых последовательностей управляющих сигналов из множества  $Y = \{y_1, y_2, \dots, y_m\}$ . Генерируемые УА последовательности предопределяются поступающими на его вход сигналами  $X = \tilde{x}_1 \tilde{x}_2 \dots \tilde{x}_n$ ;  $\tilde{x}_i \in \{0, 1\}$  из ОА, несущими информацию об особенностях реализации каждого элементарного действия на ОА.

Внешний управляющий сигнал  $f_i \in \{f_1, f_2, \dots, f_k\}$ , требующий выполнения той или иной процедуры преобразования данных, инициирует выработку УА определенной последовательности управляющих сигналов, а значит, и выполнение ОА соответствующей последовательности элементарных действий. Конкретизируя внутреннюю структуру УА, выделяют комбинационную часть и память. Комбинационная часть представляет собой комбинационную схему, реализующую систему булевых функций. Память в зависимости от способа реализации УА представляется либо регистровой схемой (автомат с жесткой логикой), либо специальным постоянным запоминающим устройством (автомат с программируемой логикой).

Структура управляющего блока цифрового автомата изображена на рисунке 2.

В структурном плане ОА представляет собой совокупность операционных элементов, связанных между собой информационными шинами, предназначенными для взаимной передачи обрабатываемых слов. Следует отметить, что структура ОА существенно зависит не

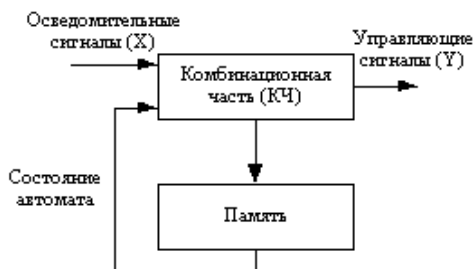


Рисунок 2 - Структура управляющего блока цифрового автомата

только от выбранных алгоритмов выполнения операций по преобразованию данных, но и от способов реализации каждого элементарного действия на операционных элементах. Значит, при проектировании обрабатывающих устройств в первую очередь следует решать задачу выбора наиболее подходящего варианта структурной организации, удовлетворяющей заданному критерию оптимальности.

Анализ алфавита спецификации ПА и способа описания его функционирования показал, что наилучшим практическим вариантом ПА является трехкомпонентная структура «Базовый УА – Процедурный УА – ОА».

Управляющий блок ПА в этом случае представляется композицией двух автоматов – базового УА (БУА) и процедурного УА (ПУА). Совместное поведение этих автоматов можно прокомментировать следующим образом. Общей входной инструкцией для композиции автоматов является

$\langle EC_k \rangle \cdot \langle LA_k \rangle$ . Воспринимая  $\langle EC_k \rangle$ , БУА переходит в состояние  $s_i \in S_1$ , где  $S_1$  – множество внешних состояний ПО. Каждая допустимая пара  $(s_i, \langle LA_k \rangle)$  является входным воздействием для ПУА, который, изменяя свои состояния  $(s_j^{t+1} = \Phi((s_i^t \times LA_{jk}), s_j^t))$ ,  $s_j^t \in S_2$ , где  $S_2$  – внутренние состояния ПО,  $t$  – индекс внутренней синхронизации), генерирует выходные сигналы  $y_j$ , иницирующие ОА на выполнение конкретных действий по преобразованию данных и формированию сообщений для информационного обмена. Относительно малая мощность множества  $S_1$  позволяет в качестве структурной модели БУА взять автомат с жесткой логикой.

Разнообразие действий, составляющих процедуры  $LA_{jk}$  ( $j=1,2,\dots$ ), и необходимость модернизации алгоритмов реализации этих процедур, во-вторых, позволяют рекомендовать реализацию ПУА в виде автомата с программируемой логикой, память которого хранит программы реализации  $\{LA_k\}$ .

Предлагаемая структурная схема ПА изображена на рисунке 3.

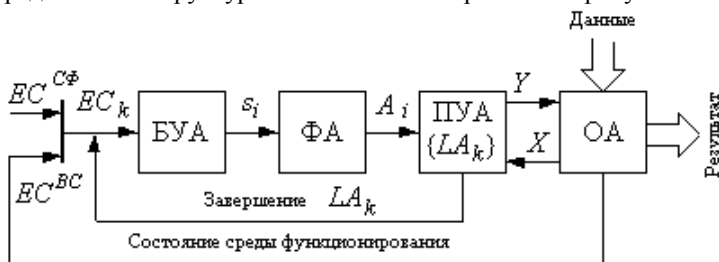


Рисунок 3 – Структурная схема протокольного автомата

Предикат  $EC_k$  наступления протокольного события  $z_k$  формируется по вектору состояния среды функционирования  $EC^{CФ}$  и вектору состояния внешней среды  $EC^{BC}$ . В соответствии с функцией переходов БУА изменяет свои состояния от какого-либо начального до конечного, если это протокол одnorазового использования, либо находится в бесконечном цикле, если автомат находится в постоянной работе. Код состояния  $s_i$  базового автомата формирователем адреса (ФА) преобразуется в адрес  $A_i$  программы, записанной в памяти ПУА. Последний совместно с ОА выполняет процедуру  $LA_k$ , сопоставленную в языке спецификаций предикату условий  $EC_k$ .

Операционный автомат в структурном плане может быть реализован как цифровое устройство с закрепленными по операционным элементам микрооперациями.

Предложенная интерпретация конечноавтоматной модели, язык спецификации протоколов и структура управляющего блока ПА позволили снять

перечисленные ранее сложности и разработать единую методику реализации протоколов в аппаратную среду функционирования.

Наблюдающаяся в настоящее время тенденция усложнения СПВ характеризуется как увеличением числа взаимодействующих ПО, так и усложнением функций, возлагаемых на ПО. Это обстоятельство затрудняет процесс проектирования ПА, спецификация которого базируется на единой конечно-автоматной модели. По мнению авторов, перспективным представляется подход, при котором модель ПО определяется композицией конечных автоматов. Такой подход в сочетании с предложенной методикой реализации ПА может дать существенную экономию как времени проектирования, так и аппаратно-программных средств.

#### ***Библиографический список***

1. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / Под. ред. В.И. Варшавского. — М.: Наука, 1986. — 400 с.
2. Поспелов Д.А. Ситуационное управление: теория и практика / Д.А. Поспелов. — М.: Наука, 1986. — 288 с.
3. Апраксин Ю.К. Приложение теории регулярных событий к анализу и синтезу протоколов распределенных систем / Ю.К. Апраксин // АСУ и приборы автоматики: Всеукраинский межвед. науч.-техн. сб. — Харьков. — 2000. — Вып. 111. — С. 3–9.

***Поступила в редакцию 11.03.2003 г.***