

УДК 681.5:656.257

И.А. ФУРМАН, М.Л. МАЛИНОВСКИЙ, А.Ю. АЛЛАШЕВ, С.Я. БОВЧАЛЮК

*Харьковский национальный технический университет сельского хозяйства
им. Петра Василенко, Украина*

ОПЫТ И ПЕРСПЕКТИВЫ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ АЛГОРИТМОВ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ ОБЪЕКТАМИ КРИТИЧЕСКОГО ПРИМЕНЕНИЯ

Представлена история развития методов и средств построения логических управляющих автоматов параллельного действия, рассмотрены особенности их структурной организации, концепция разработки методологии и инструментальных средств их программирования, перспективы их использования в системах критического применения

ПЛК параллельного действия, БЛП-контроллеры, TVP-технология, язык программирования CycloGraf, метод цикломатного программирования

Введение

Проблемой распараллеливания процессов обработки информации человечество занимается очень давно. Достаточно сказать, что уже в древние века представители известного племени Майя в Южной Америке при вычислениях использовали систему, которую в наше время называют системой счисления в остаточных классах (СОК), где операции выполняются не над числами, а над их остатками от деления на выбранные основания, и где в отличие от позиционных систем счисления (ПСС) отсутствует понятие переноса, а, следовательно, и все связанные с ним сложности и неприятности.

Рассмотрим теперь историю развития методов и средств реализации алгоритмов логического управления в недалеком прошлом (с конца 19-го – начала 20-го веков). С указанного времени практически все системы логического управления строились по принципу "жесткой" логики, на релейно-контактной аппаратуре и лишь к началу 60-х годов прошлого столетия эти системы перестали удовлетворять предъявляемым к ним требованиям, как в части надежности функционирования, так и в части способности к реконфигурации при развитии или модернизации управляемого объекта.

"Жесткую" логику сменила программируемая, что позволило не только корректировать алгоритмы управления без изменения электрического монтажа, но и вообще осуществить революцию в технологии проектирования - заменить процесс разработки систем логического управления их проектной компоновкой. При этом основная доля усилий стала приходиться на разработку программного обеспечения. Однако, никто не акцентировал внимание на том, что при переходе к программируемой логике мы вынуждены были перейти от параллельной (на релейных схемах) реализации алгоритмов управления к последовательной. Этот недостаток в значительной мере компенсируется быстродействием современных микропроцессоров и поэтому практически все технологическое оборудование управляется с помощью программируемых логических контроллеров (ПЛК) и промышленных (промышленных) компьютеров (ИК), которые являются управляющими автоматами последовательного действия. И те и другие приспособлены к работе в производственных условиях и отличаются лишь тем, что ПЛК ориентированы на непосредственное управление агрегатами в режиме "жесткого" реального времени, а ИК более эффективно решают задачи обработки и визуализации больших объемов информации на верхних уровнях интегрированных систем автоматизации.

Массовое практическое применение ПЛК выявило не только безусловные их преимущества в сравнении с другими средствами управления, но и существенные недостатки, основным из которых является уже указанный выше последовательный принцип обработки информации, что приводит к известному противоречию между фантастическим быстродействием современных аппаратных средств (тактовая частота уже составляет десятки ГГц) и малой длиной машинных слов (макс. - 64, да и те за 2 такта). При этом оказывается, что быстродействие обслуживания контролируемых входов контролера существенно зависит от их количества. Кроме того, микропроцессорные ПЛК за счет большого количества последовательно исполняемых команд при управлении объектами критического применения (в атомной энергетике, на железнодорожном транспорте и в метрополитенах, в летательных аппаратах) зачастую не обеспечивают необходимого уровня надежности и достоверности обработки данных и поэтому для непосредственного управления агрегатами и установками в указанных отраслях используются крайне ограниченно. Здесь следует заметить, что до сих пор с помощью микропроцессорных средств без использования двойного дублирования и (или) мажоритирования не удается достичь интенсивности появления опасного отказа 10^{-13} , которая обеспечивается в релейных системах ж/д автоматики и к тому же без резервирования.

Исторический аспект

В работах И. В. Прангишвили, А. Д. Закревского, Э. А. Якубайтиса, С. И. Баранова, В. А. Складорова и др. авторов еще в 60-70 г. прошлого века были показаны значительные преимущества регулярных микроэлектронных структур при решении заданий логического управления. Сначала появилась монография И. В. Прангишвили по однородным микроэлектронным структурам, затем А. Д. Закревским было сформулирована абстрактная модель параллельного логического автомата, а затем уже Э. А. Якубайтис в своих

работах наглядно показал принципиальную возможность построения на основе регулярных микроэлектронных структур (в частности – ПЛМ) сверхбыстродействующих ("однотактных") логических автоматов параллельного действия. В 80-е годы прошлого века в институте вычислительной техники Академии наук Латвийской ССР, возглавляемом Якубайтисом, велись работы по созданию компьютерной сети Академий наук Союзных республик и в которой в качестве быстродействующих коммутирующих устройств предполагалось использовать логические автоматы на основе ПЛМ. В 90-х годах прошлого века в институте ВНИИТЭлектромаш (г. Харьков) была сделана попытка разработать логические управляющие автоматы параллельного действия как функционально законченные устройства, в результате чего были созданы и внедрены в производство на многих предприятиях бывшего СССР первые в мировой практике промышленные модели ПЛК параллельного действия (или параллельные ПЛК – ППЛК).

По мнению авторов, представляет определенный интерес история создания во ВНИИТЭлектромаше этих контроллеров. В указанном институте разрабатывались технология и оборудование для производства асинхронных электродвигателей массовых серий. Номенклатура создаваемых агрегатов, технологических модулей, автоматических линий и технологических комплексов была довольно большой и в отделах – разработчиках систем автоматики и программного управления имела место проблема: как унифицировать применяемые средства и системы управления, в том числе особо остро стоял вопрос: как упорядочить, как формализовать процесс разработки технических заданий (ТЗ) на системы управления с тем, чтобы по крайней мере уменьшить то значительное число ошибок в проектировании, которые возникали по причине нечеткого формулирования задания.

С целью разрешения указанной проблемы была сделана попытка формализации процесса составле-

ния ТЗ с использованием в качестве первичного описания языка технологических циклограмм в табличном виде, где строка таблицы соответствует рабочему такту (шагу) цикла и где в каждой строке, в одной ее части записывается совокупность (вектор) команд управления исполнительными механизмами, а в другой – комбинация состояний, в которые должны прийти датчики при срабатывании механизмов.

Предложенный подход очень быстро прижился по трем причинам. Во-первых, ТЗ стало компактным (вместо 30-50 стр. один лист формата макс. А1), во-вторых, ТЗ стало намного легче составлять – просто заполнять таблицу определенной формы, а в-третьих это ТЗ стало наглядным для контроля и выявления ошибок, при этом главным инструментом являлась обычная метровая линейка.

Затем специалисты по разработке систем программно-логического управления обратили внимание на то, что упомянутая выше таблица по существу представляет собой матричную структуру, подобную структуре микроэлектронного ПЗУ. Вот тогда и возникла идея: а что, если это подобие и вообще свойства регулярности, присущие технологическим агрегатам дискретного циклического действия, технологическим языкам описания алгоритмов управления такими объектами и матричным микроэлектронным структурам, попытаться как-то рационально сочетать и использовать на практике. Реализация на практике указанной идеи и привела к созданию и промышленному применению уже упомянутых ПЛК параллельного действия.

Указанные ПЛК имели безусловные преимущества в быстродействии и надежности перед ПЛК последовательного действия, программировались непосредственно на технологическом языке пользователей, но были реализованы на микросхемах малой и средней степени интеграции и поэтому имели ограниченные функциональные возможности и не обеспечивали их программирования посредством автоматизированных инструментальных средств.

В последние годы для надежной реализации логических функций все чаще стали применять программируемые логические интегральные схемы (ПЛИС) как перспективную элементную базу, в которой в полной мере реализованы возможности и преимущества регулярных микроэлектронных структур. Причем для управления особо ответственными объектами, например, для включения аварийных защит на АЭС рекомендуется к применению "жесткая" логика, реализованная программным путем на ПЛИС. На первый взгляд - парадокс: 21 век и возврат к "жесткой" логике. Однако, при более серьезном рассмотрении – это использование самой надежной и на сегодняшний момент реализации логических алгоритмов, но с применением новейших технологий ПЛИС.

Методология и принципы

Рассмотрим далее методологию и основные принципы реализации программируемой логики на основе ПЛИС с использованием управляющих автоматов параллельного действия и параллельных ПЛИС-контроллеров как функционально законченных устройств.

В основу математической модели логического управляющего автомата параллельного действия положено матричное представление как управляемых объектов (при этом рассматриваются технологические агрегаты дискретного циклического действия), так и алгоритмов управления ними. Так, например, циклограмма функционирования технологического агрегата (ТА) может быть задана с помощью двух прямоугольных булевых матриц конечных размеров: матрицы состояний управляемых механизмов и матрицы состояний контролируемых датчиков.

$$\|A\| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{s1} & a_{s2} & \dots & a_{sk} \end{vmatrix} \quad \|C\| = \begin{vmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{s1} & c_{s2} & \dots & c_{sm} \end{vmatrix}$$

А структура логического управляющего автомата, реализованного на основе такой модели, в

терминах теории цифровых автоматов представляет собой сочетание комбинационного автомата, используемого для выбора подпрограммы, подлежащей реализации в i -й момент времени, и последовательного автомата Мура, обеспечивающего последовательную (по шагам технологического цикла) и параллельную на каждом шаге отработку управляющей подпрограммы (рис. 1).

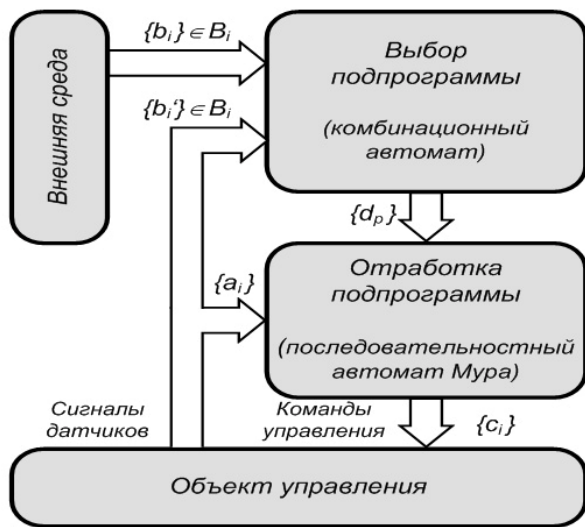


Рис. 1. Структура управляющего автомата

Указанная структура имеет ряд особенностей:

1. В отличие от ПЛК с классической архитектурой, в которых опрос датчиков и отработка управляющей программы осуществляется последовательно, в данном контроллере производится явно параллельно, за 1 такт, что и позволяет многократно повысить быстродействие контроллера как управляющего устройства промышленного назначения. Однотактный принцип работы позволяет эффективно использовать образующуюся временную избыточность и осуществлять, например, при оптимальной величине такта в 1 мс 16-32-х кратный опрос состояний входов, что не менее, чем на порядок позволяет повысить достоверность обработки потока входной информации.

2. Управление и синхронизация процесса отработки управляющих программ осуществляется не от тактового генератора, а потоком состояний входов,

то есть потоком, который образуют в асинхронной последовательности интервалов дискретного автоматного времени комбинации состояний управляемого объекта и внешней среды.

3. Используемый принцип управления позволяет строить из параллельных ПЛК сети произвольной конфигурации без введения дополнительной синхронизации и без применения каких-либо протоколов.

Дальнейшее развитие архитектуры ПЛК параллельного действия связано с их распространением в область управления объектами критического применения, где к элементной базе предъявляются наиболее жесткие требования в части обеспечения функциональной безопасности и надежности. С учетом современных требований к средствам и системам управления, разработка безопасных ПЛК параллельного действия, которые в дальнейшем получили название БЛП-контроллеров, требовала решения трех основных задач: 1) разработка архитектуры управляющих автоматов для БЛП-контроллеров (БЛП-автоматов); 2) разработка HDL-моделей, которые могли бы использоваться как библиотечные ресурсы (макрофункции); 3) разработка методологии и инструментальных средств проектирования программного обеспечения для БЛП-контроллеров.

Первая задача решалась в двух направлениях, которые определяются исходными (базовыми) моделями управляющих автоматов. В качестве 1-й модели использовался традиционный конечный автомат, который при реализации на ПЛИС также обеспечивает параллельную однотактную реализацию логических функций. В качестве 2-й исходной модели использовалась описанная выше модель ПЛК параллельного действия. В результате была получена совокупность абстрактных и структурных моделей БЛП-автоматов, которые, в отличие от известных, обладают следующими особенностями:

1. Каждая модель описывает управляющее устройство как набор компонентных автоматов, каж-

дый из которых выполняет функции одного из каналов обработки информации;

2. Сравнение результатов обработки информации различными компонентными автоматами и выбор «более безопасного» состояния в БЛП-автоматах М-типа выполняется с учетом предыдущего состояния автомата;

3. Входные и выходные состояния БЛП-автоматов задаются не в виде вектора дискретных сигналов, а в виде сигналов, в которых признаком их состояния являются динамические свойства, например частота;

4. Структурные модели БЛП-автоматов учитывают выполнение функции многократной проверки правильности принятия управляющих решений.

Основными задачами разработки HDL-моделей являются: выполнение компьютерного моделирования и анализ функционирования БЛП-автоматов; получение текстового описания стандартных компонентов, на основе которых строятся БЛП-автоматы, а также непосредственно архитектуры БЛП-автоматов с целью их дальнейшего практического использования при разработке программного обеспечения для БЛП-контроллеров. Полученные HDL-модели представляют собой параметризованные модули, которые могут быть настроены на выполнение заданных функций путем изменения ряда параметров. К таким параметрам, в частности, относятся: количество входов-выходов БЛП-автомата, кратность и период проверки правильности принятия управляющих решений, настройка входных и выходных функциональных преобразователей, преобразующих сигналы из импульсных в постоянные и из постоянных в импульсные, настройка модулей, выполняющих сравнение результатов обработки информации различными каналами и т.д.

Разработка методологии и инструментальных средств проектирования программного обеспечения

для БЛП-контроллеров базировалась на следующей концепции:

1. Разрабатываемый язык, инструментальные средства и технология их применения должны быть существенно более простыми, наглядными и доступными непрофессиональным пользователям, чем существующие на сегодняшний день и ориентированными в большей степени на специалиста-технолога, нежели на профессионального программиста.

2. Язык, рассчитанный на непрофессионального пользователя, должен удовлетворять требованию психологической естественности, основанной на максимальном приближении языка программирования к языку пользователей технологического оборудования. Такая психологическая естественность предполагает, прежде всего, устранение различий между языком, на котором пользователь формирует алгоритм управления, и языком, на котором собственно осуществляется программирование. Т.е. алгоритмический язык становится в данном случае языком программирования.

3. Максимальная разгрузка языка от машинных атрибутов и сокращение количества используемых элементов и конструкций.

В соответствии с данной концепцией была предложена так называемая TVP-технология – технологическое визуальное программирование, которая позволяет непосредственно инженеру-технологу, как носителю экспертных знаний об объекте автоматизации, в естественной для него форме задать алгоритм управления на высоком уровне абстракции в виде технологической табличной модели. В рамках данной технологии по разработанной и отлаженной табличной модели управления технологическим оборудованием специализированный транслятор автоматически генерирует программное обеспечение на выбранном языке программирования, например на языках описания аппаратуры.

Ключевой задачей, которая решалась при создании TVP-технологии являлась разработка эффективного языка программирования (в дальнейшем получившего название CycloGraf), ориентированного на пользователя, не имеющего навыков разработки программного обеспечения. Язык CycloGraf поддерживает два метода проектирования программного обеспечения: метод структурного проектирования ПО, который за счет значительного сокращения количества и упрощения конструкций языка программирования обеспечивает уровень наглядности программ, доступный технологу, не имеющему навыков программирования (таким образом достигается существенное повышение эффективности взаимодействия программиста и технолога), а также метод цикломатного проектирования ПО, который обладает всеми свойствами технологического программирования, предполагающего возможность составления программ непосредственно технологом без участия программиста.

Цикломатное программирование сводится к формированию пространства состояний объекта управления в процессе анализа его работы путем заполнения двух взаимосвязанных таблиц, одна из которых описывает последовательность выполняемых операций для каждого микроцикла, а вторая – условия переходов к этим микроциклам. Такое описание алгоритмов управления технологическими объектами является наглядным, первичным по своей природе и не требует навыков программирования. Однако область эффективного использования цикломатного программирования ограничивается относительно несложными задачами управления объектами явно выраженного циклического действия.

Метод структурного программирования на языке CycloGraf, обладая простой формой представления алгоритмов работы функциональных блоков, позволяет реализовать цифровые устройства практически любой сложности, но, одновременно, проигрывает в наглядности описания цикломатному проектирова-

нию ПО. В связи с этим, структурное программирование на языке CycloGraf эффективно при решении сложных и нестандартных задач и требует от пользователя определенных навыков построения цифровых устройств. При этом следует отметить, что описанные конструкции языка CycloGraf обладают явными преимуществами по сравнению с известными средствами проектирования ПО для ПЛИС, которые проявляются как при подготовке ПО, так и при его сопровождении, анализе и коррекции, а табличная форма представления алгоритмов делает их доступными для понимания технологами, не имеющими навыков программирования.

Заключение

Перспективы развития архитектуры ПЛК параллельного действия и методологии проектирования на их основе систем логического управления, по мнению авторов, связаны с их адаптацией к конкретным применениям: управления объектами систем централизации на ж.-д. транспорте, управления системой аварийной защиты на АЭС и т.д. При этом многие технические решения, как в части аппаратной, так и программной реализации, должны быть приведены к типовым, что существенно повысит эффективность использования параллельных контроллеров на основных этапах жизненного цикла систем управления: разработки, сертификации, проектирования, эксплуатации и модернизации.

Литература

1. Фурман И.А., Краснобаев В.А., Малиновский М.Л., Панченко С.В. Контроллеры и процессоры с параллельной архитектурой. – Х, 2006. – 416 с.

Поступила в редакцию 31.01.2008

Рецензент: д-р техн. наук, проф. В.А. Краснобаев, Харьковский национальный технический университет сельского хозяйства им. Петра Василенко, Харьков.