

УДК 681.58

Кардаш Д. И.

## БЫСТРОДЕЙСТВУЮЩИЙ ПРОГРАММИРУЕМЫЙ ЛОГИЧЕСКИЙ КОНТРОЛЛЕР НА ОСНОВЕ ПРОГРАММИРУЕМОЙ ЛОГИКИ

*В статье приводится новый способ построения программируемых логических контроллеров на основе использования FPSLIC интегральных схем. Предлагается замена управляющих программ контроллеров аппаратной реализацией автоматов, построенных на базе программируемой логики. Рассматривается пример программы в автоматнo-ориентированной парадигме программирования. Предложена методика использования быстродействующих логических контроллеров на базе FPSLIC интегральных схем.*

В условиях повышения требований к качеству процессов управления существует задача сокращения длительности цикла управления, осуществляемого с помощью специализированных ЭВМ. При этом, в качестве конкурирующего требования оказывается стремление сократить как время разработки, так и реконфигурации электронных систем управления.

Уровень современных технологий создания микропроцессорной техники позволяет создавать современные микроконтроллеры большой производительности. Однако их использование при построении базового элемента систем управления промышленными системами – программируемого логического контроллера (ПЛК) сталкивается с рядом ограничений.

Во-первых, ПЛК должен обладать определенной долей универсальности. Это достигается за счет использования специальных системных мониторов (операционных систем), предназначенных для исполнения управляющих программ, загружаемых по информационному интерфейсу ПЛК. Во-вторых, ПЛК должен быть серийным устройством, так как именно это и позволяет сократить время подготовки, модернизации производства, повысить надежность и улучшить иные технические характеристики контроллера.

При существующей практике построения систем управления, программирование идет по экстенсивному пути создания программ. Это следует из того, что многие современные технологические процессы довольно инертны и не требуют малого времени управляющего цикла. В качестве примера можно привести продукцию одного из мировых лидеров автоматизации производства – фирмы *Siemens Automatic*. Так контроллеры серии S7-300 обеспечивают минимальное время выполнения одного цикла программы равное 10 мс. Учитывая теорему Котельникова, можно подсчитать, что частоты протекающих в объекте управления процессов не должны превышать 50 Гц. В том случае, если возникает потребность уменьшения длительности данного цикла, для использования предлагается серия контроллеров S7-400, проигрывающих по массогабаритным,

стоимостным (4000 евро, против стоимости S7-300 порядка 500 евро) и эксплуатационным характеристикам. Требование обеспечения универсальности ПЛК S7-400 приводит к тому, что большая часть возможностей контроллера в этом случае не используется, а применение высокопроизводительного ПЛК S7-400 может оказаться неэффективным.

Под экстенсивностью написания программ в данном контексте понимается программная реализация тех функций, аппаратная реализация которых была бы более эффективна (например, программный опрос входящих сигналов вместо реализации аппаратного прерывания или блока быстрого ввода). Реализация подобных функций в виде аппаратных блоков производится, однако, это лишь частичное решение по повышению производительности.

Создание управляющей ЭВМ в виде специфичного электронного изделия порождает необходимость таких шагов, как разработка аппаратуры, реализация вычислителя и его программного обеспечения. Все эти шаги неминуемо приводят к увеличению времени подготовки производства, что в условиях динамизма современного мира недопустимо.

Другим из подходов является реконфигурируемая логика и построение систем на кристалле (*SoC – System on Chip*). Основными преимуществами такого подхода являются миниатюрность и универсальность получаемых решений. Построение *SoC* названо стратегической задачей микроэлектроники [1].

Изначально эта задача решалась с помощью «классической» программируемой логики, программируемых логических интегральных схем (ПЛИС, *PLD – Programming Logic Devices*). В этом случае внутренняя аппаратная топология микросхемы определяется с помощью сторонних программирующих устройств и в ряде случаев является однократно записываемой. Следующим шагом в развитии программируемой логики явилось создание микросхем *FPGA (Field Programmable Gate Array)*. За внутреннюю структуру в них отвечает специальная оперативная память, которая загружается из внешней конфигурационной памяти каждый раз по включении напряжения питания. При использовании *FPGA* возникла возможность реконфигурации электронных схем в процессе работы, но сохранилась потребность во внешнем программаторе.

Сложность и функциональность современных ПЛИС позволяет реализовывать модули, решающие разнообразные функциональные задачи, вплоть до реализации процессорных ядер. Однако в рамках рассматриваемой задачи построения быстродействующего ПЛК создание процессорных ядер с помощью ПЛИС видится излишним. Создание микроконтроллеров или микропроцессоров промышленно с применением жесткой архитектуры видится более эффективным. На ПЛИС рационально реализовывать лишь специализированные функции. Реализация *SoC* с помощью только микросхем *FPGA* так же наталкиваются на некоторые препятствия, изложенные в [2].

В качестве современного способа построения систем на кристалле можно указать технологию *FPSLIC (Field Programmable System Level Integration Circuits)* [3]. В рамках одной интегральной микросхемы размещаются не только элементы программируемой логики (*FPGA*), но и процессорное ядро с требуе-

мым набором периферийных модулей и необходимая внедренная периферия (например, аппаратная поддержка распространенных интерфейсов связи). При этом создается возможность реконфигурации в процессе работы без дополнительных программирующих устройств.

Так компания *Atmel* производит серию микросхем *FPSLIC*® II на основе ядра *AVR*, являющихся первым семейством динамически реконфигурируемых систем на кристалле. Процесс реконфигурации управляется встроенным *AVR*-процессором и реконфигурационным контроллером. Динамически реконфигурируемые *SoC FPSLIC* II интегрируют 8-разрядный *AVR*-процессор со статическим ОЗУ для хранения программы/данных, аппаратным умножающим устройством, периферийными устройствами и ядром *FPGA*. В силу этого предлагается использовать *MC FPSLIC* для построения быстродействующих ПЛК.

Одной из необходимых составляющих быстродействующего ПЛК является использование перспективной парадигмы программирования – автоматного программирования. При ее использовании программа или её фрагмент осмысливается как модель формального автомата. При применении данной парадигмы при построении непосредственно программ для вычислителя, данный автомат является абстракцией. Использование же в качестве составной части *SoC FPGA* модуля дает возможность аппаратной реализации такого автомата.

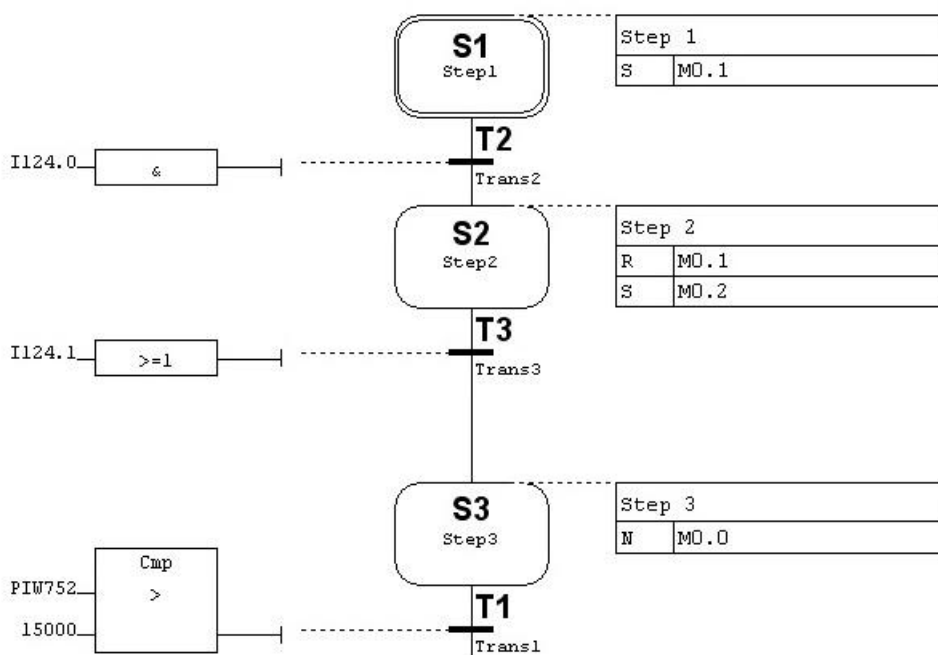


Рисунок 1 – Пример программы на языке *S7-GRAPH*

В задачах автоматизации производства автоматное программирование используется и в настоящее время. Так, фирма *Siemens Automatic* имеет в составе используемого программного обеспечения такие пакеты, как *S7-GRAPH* и *S7-HIGRAPH*, для создания управляющих программ (синхронного и асинхронного типа соответственно) ПЛК в виде специализированных сетей Петри [4]. Данная форма представления удобна и практически используется при определении процессов управления производственными системами. Пример програм-

мы на языке *S7-GRAPH* приведен на рис. 1. Каждой из вершин приписываются выполняемые действия (изменяемые переменные), а каждому переходу поставлены условия его осуществления (значения переменных, входов или результаты сравнения многобитных переменных).

Данная форма представления предлагается в качестве прототипа для написания программ конфигурации *FPGA* модуля предлагаемого ПЛК. При этом реализация управляющей программы будет проходить в виде аппаратно реализуемого автомата, что позволит значительно повысить его производительность.

На рис. 2 приведен способ использования программно-аппаратного комплекса с использованием предлагаемого быстродействующего ПЛК.



Рисунок 2 – Применение быстродействующего ПЛК на базе *FPSLIC*

Его работу можно представить в виде последовательности следующих шагов:

Шаг 1. Пользователь создает автоматную модель управляющей программы (подобную представленной на рис. 1).

Шаг 2. С помощью программы конфигурации аппаратуры полученная на первом шаге модель преобразуется в текст программы конфигурации на языке *VHDL* или *Verilog* [5].

Шаг 3. С помощью инструментальной программы конфигурации *FPGA* полученная программа конфигурации адаптируется для использования в быстродействующем ПЛК. В качестве такой программы может выступать программа *System Designer* фирмы *Atmel* [6], либо специальное программное обеспечение.

Шаг 4. Через информационный интерфейс (обычную промышленную сеть: *Profibus*, *CAN*, *ModBus* и т.п.) конфигурационная информация поступает на внутренние модули, осуществляющие функции приема/передачи между микропроцессорным ядром и окружающей средой.

Шаг 5. Микропроцессорное ядро создает в конфигурационной памяти образ автомата, который впоследствии реализуется в блоке программируемой логики.

Шаг 6. Созданный в блоке *FPGA* автомат через аналогово-цифровые, цифро-аналоговые преобразователи и электронику входов/выходов осуществляет управление объектом.

Если учесть, что средняя тактовая частота работы *FPGA* блока микросхем *FPSLIC* составляет порядка 100 МГц, время отклика созданного таким образом управляющего автомата составит 10 нс. (без учета времени преобразования аналоговых сигналов). Сравнение такого времени и времени отклика существующих ПЛК (на примере *S7-300*) в 10 мс. дает основание говорить о существенном повышении качества (производительности) работы электронной системы управления.

Анализируя предлагаемую схему ПЛК можно сделать предложение о том, что бы в состав функциональных блоков, доступных микропроцессорному ядру, либо блоков, подключаемых к *FPGA*, были добавлены аналоговые преобразователи. Это позволит реализовать *SoC* с большей эффективностью.

Таким образом, можно сделать вывод о том, что предлагаемый способ построения ПЛК позволит существенно повысить быстродействие управляющих электронных устройств, сохранив при этом универсальность применения и возможность гибкой реконфигурации (смены управляющего автомата).

#### Список литературы

1. Кривченко, И. Системная интеграция в микроэлектронике – *FPSLIC* [Текст] / И. Кривченко // *Chip News*. – 2000. – № 3. – С. 4-10, № 4 С. 62-64.

2. Королев, Н. Atmel *FPSLIC* – элементная база 21 века. [Текст] / Н. Королев // *Chip News*. – 2001. – № 1. – С. 14-17.
3. *FPSLIC™ (AVR with FPGA)* [Электрон. ресурс] / *Atmel*. – Режим доступа: <http://www.atmel.com/products/fpslic/default.asp>
4. *SIMATIC S7-GRAPH V5.3* для S7-300/400. Программирование систем последовательного управления. Руководство [Электрон. ресурс] / *SIEMENS*. – Редакция 02/2004 A5E00290656-01. – Режим доступа: <http://www.automation.siemens.com>
5. Суворова, Е. А., Шейнин, Ю. Е. Проектирование цифровых систем на *VHDL* [Текст] / Е. А. Суворова, Ю. Е. Шейнин. – СПб.: *BHV*, 2003. – 560 с.
6. *FPSLIC (AVR with FPGA) - Tools & Software* [Электрон. ресурс] / *Atmel*. – Режим доступа: [http://www.atmel.com/dyn/products/tools.asp?family\\_id=627](http://www.atmel.com/dyn/products/tools.asp?family_id=627)