

УДК 621.03

Ю.Б. ЮРЧЕНКО

НПП Хартрон-АРКОС, Україна

## СИНХРОННЫЕ «АЛГОРИТМ-В-АППАРАТУРЕ» СТРУКТУРЫ В УПРАВЛЯЮЩИХ СИСТЕМАХ БОРТОВЫХ КОМПЛЕКСОВ КРИТИЧЕСКОГО ПРИМЕНЕНИЯ: ПРОГРАММНО-КОНСТРУКТОРСКАЯ НЕОБХОДИМОСТЬ И ВОЗМОЖНОСТИ АППАРАТУРНОЙ РЕАЛИЗАЦИИ

Проведена оценка реализуемости заданных требований к вычислительной системе в рамках доступных аппаратных средств, а так же путей оптимизации гарантоспособных архитектур на базе применения программно-параллельных «алгоритм-в-аппаратуре» систем при синтезе комплексов для особо тяжелых условий эксплуатации с длительным сроком активного функционирования. Обоснована целесообразность и проанализирована эффективность трансформации задачи из алгоритмическо-программной в алгоритмическо-аппаратную, а также применения синхронизации для контроля в резервированных структурах вычислительных систем.

**встроенная система, отказоустойчивость, реальное время, реконфигурируемая архитектура, уровень целостности**

### Введение

**Проблема производительности и гарантоспособности встроенных управляющих систем.** При построении бортовых вычислительных комплексов автоматических систем управления критического применения особо важной задачей есть контролируемость истинности функционирования [1 – 3] и своевременность адекватной реакции при сбоях или отказах [4]. Построение системной структуры специальных вычислителей на основе готовых блоков [5] ориентировано на выполнение основных задач контроля и поддержания отказоустойчивости центральным процессором. Принимая во внимание, что вычислительный и управляющий процессы должны исполняться в масштабе реального времени с жестким внутри-тактовым планированием задач, а вычислительная производительность процессора конечна и пропорциональна энергопотреблению (а равно и тепловыделению), то выделяемое время на исполнение контролирующего (а при необходимости восстанавливающего) процесса минимально. Кроме того, есть необходимым параллельное исполнение процесса поддержания отказоустойчивости системы с процессами

основной задачи [6, 7] для недопущения лавинообразного обвала резерва в реконфигурируемой архитектуре [4, 5].

Сочетание требований высокой системной производительности и крайних конструктивных ограничений обусловило направление применения электронных компонент класса Industry [1] при синтезе вычислительных систем бортовых комплексов ракет-носителей и космических аппаратов, а опыт применения однокристалльных микроконтроллеров [2] подтвердил возможности реализации хорошо зарекомендовавших себя базовых архитектур с привлечением встроенных систем. Однако при повышении требований к комплексам и возможностей вычислительной производительности используемых процессорных элементов, увеличивается количество обслуживаемых абонентов, и, как следствие, повышается нагрузка на внешний интерфейс процессора и собственно процессор, так как производится внутри-тактовое перераспределение между вычислительным временем и временем по обслуживанию драйверов обмена. При этом скорость физических интерфейсов конечна, информационный протокол обмена заранее задан

изготовителем аппаратуры периферийного абонента и обязателен в исполнении, а поддержание его возлагается на драйвер вычислительной системы. Это приводит к увеличению непроизводительно затрачиваемого процессором времени на обслуживание драйверов ввода-вывода и при дальнейшем повышении сложности системы, в свою очередь, к необходимости дополнительного повышения его вычислительной мощности.

**Цель исследования:** повышение показателя, определяемого отношением интегральной производительности резервированной вычислительной системы к производительности используемого процессора без ухудшения параметров гарантоспособности.

### Анализ подходов

Развитие технологии компонент программируемой логики фирм Actel, ALTERA и др. в сочетании с высокоуровневой поддержкой полного цикла проектирования кристалла позволяет синтезировать конечные автоматы с жесткой и полужесткой логикой даже на элементах FPGA (Field Programmable Gates Array) малой и средней интеграции [6]. Применение высоко- и сверхвысокоинтегральных FPGA позволяет организовать внутри одного чипа не только элементы логики, но и массивы RAM, ROM, а также узлы процессорных ядер. Внедрение в поддержку проектирования языка Verilog высокоуровневого формализованного описания логики функционирования [8] дает реальную возможность синтезировать программно-аппаратные драйверные модули интерфейсов. Таким образом, каждый алгоритм взаимодействия с абонентом, выделяясь как частично независимый модуль, излагается в операторах языка Verilog описания конфигурации FPGA, а не операторах языка программирования процессорного модуля. Последующие этапы полного цикла проектирования кристалла FPGA заканчивают трансформацию драйвера из алгоритмически-программного в алгоритмически-аппаратный, а интерфейс с процессорным модулем представляет типовое

подключение элементов памяти, где каждому из абонентов выделено соответствующее адресное пространство для информационных массивов.

Учитывая, что в комплексах рассматриваемого класса применяется жесткое потактовое планирование исполнения процессов, а так же анализируя общесистемные требования и возможности абонентов в соответствии с их интерфейсами, точки времени выдачи команд на включение исполнения драйверов обмена определяемы заранее с точностью до готовности принятых данных к непосредственному использованию в головных системных алгоритмах. Такой подход применим при обслуживании абонентов - датчиков. Что же касается абонентов - исполнительных механизмов, то инициация запроса на обмен возможна непосредственно после занесения выдаваемой информации процессорным модулем в буферную память аппаратного драйвера.

При этом целесообразно инициацию требований на обмен, выдаваемых процессором по внутритактовой сетке времени, перенести на аппаратный уровень, внедрив в системную службу времени вычислительно-управляющего комплекса, а обмена с приемом запросов на прерывание по готовности абонентов выполнять аппаратным драйвером непосредственно после их получения, исключая необходимость двойного перевода процессора между уровнями задач. Конвейер приема-выдачи информационных потоков, синтезированный подобным образом, полностью высвобождает процессор от непроизводительных накладных расходов.

### Оценка производительности

Системное улучшение производительности можно оценить следующим показателем:

$$k = \frac{1 + \sum_{n=1}^{N_a} (t_n + T_n)}{1 + N_a T_{RAM}},$$

где  $N_a$  – количество обслуживаемых абонентов;  $t_n$  – аппаратные затраты на время прохождения информации по линии интерфейса  $n$ ;  $T_n$  – затраты

процессором реализуемого драйвера обслуживания линии интерфейса  $n$ ;  $T_{RAM}$  - время доступа к массиву данных драйвера с аппаратно реализованным обслуживанием.

Хотя, с одной стороны, драйвер «алгоритм-в-аппаратуре» функционирует без непосредственного участия процессорного модуля, что позволяет достичь ощутимых результатов преимущества реализации параллелизма процессов приема, обработки и выдачи информации, с другой стороны, образуется дополнительный аппаратный уровень в иерархии целостности [5] с необходимостью его контроля и обслуживания в отказоустойчивой системе.

Принимая во внимание, что при построении отказоустойчивых комплексов используются резервированные структуры, для контроля состояния каналов с горячим резервированием применимо аппаратно-логическое сравнение идентичной информации.

Применяя методы синхронизации процессов [2] в аппаратно-программных модулях, процесс обнаружения сбоя или отказа в узловых точках синтезируемой вычислительно-управляющей системы, возможно, не представляет особой сложности. С точки зрения гарантоспособности целесообразно внедрить аппаратный драйвер в уровень поддержания целостности синтезируемой системы и встроить в него средства самоконтроля.

Оценка предложений проведена при определении необходимой производительности процессорного модуля (MIPS CPU) вычислительной системы с системной производительностью (MIPS system) обработки информации в 4,5 MIPS (Million Instructions Per Second).

Проведены оценочные сеансы приема данных по интерфейсу RS485 с процессорной поддержкой для различных времен ( $\Psi$ ) базовых тактов опроса, а также с варьированием количества однотипных абонентов ( $N_a$ ) и идентичных протоколов обмена.

Подключение одиночного абонента не потребовало необходимости применения предъявления особых требований к процессору (рис.

1). Однако подключение каждого последующего абонента вызывало необходимость резкого повышения производительности процессора, особенно при высокой частоте (до 100Гц) опроса абонентов (рис. 2).

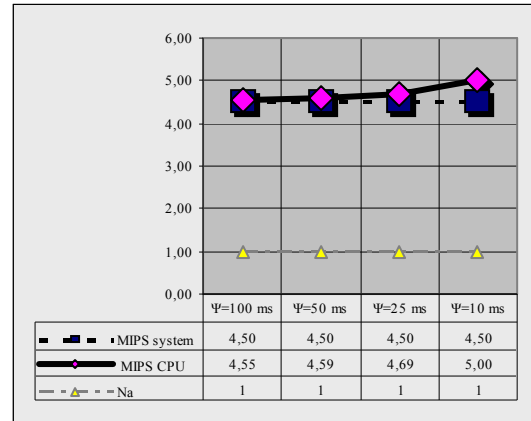


Рис. 1. Требуемое повышение производительности процессора для сохранения прежнего вычислительного уровня системы при программной поддержке драйвером одного абонента и различного периода опросов

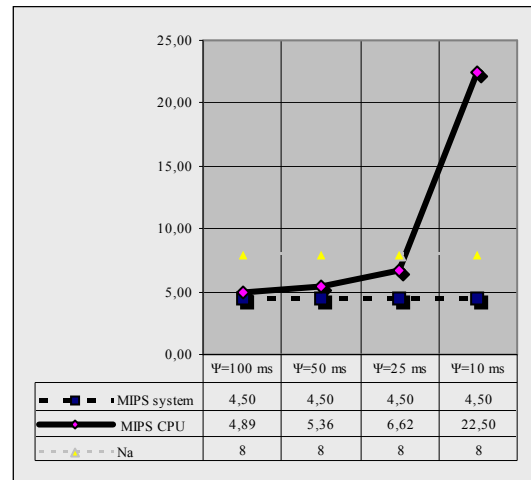


Рис. 2. Требуемое повышение производительности процессора для сохранения прежнего вычислительного уровня системы при программной поддержке драйвером восьми абонентов и различного периода опросов

Применение синтезированного в среде пакета MAX+plusII [9] чипа «алгоритм-в-аппаратуре» драйвера, совместно с информационным мостом на скоростной двух-портовой памяти позволило подтвердить эффективность реализации замещения программно-процессорного построения драйвера его аппаратным функциональным аналогом (рис. 3).

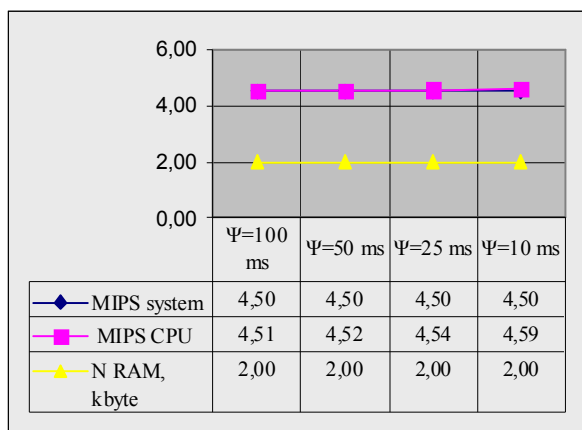


Рис. 3. Требуемое повышение производительности процессора для сохранения прежнего вычислительного уровня системы при аппаратном параллельном драйвере обслуживания 16-ти абонентов и различного периода опросов

### Выводы

1. Трансформация части алгоритмов из программного в аппаратный уровень возможна и целесообразна. Реализуется средствами проектирования чипов программируемой логики.

2. Применение предлагаемой параллельной архитектуры «алгоритм-в-аппаратуре» значительно повышает эффективность использования процессорных модулей в управляющих системах, увеличивает интегральную производительность системы без привлечения сверхмощных процессоров.

3. Мониторинг состояния аппарата драйвера реализуется в контексте поддержания целостности системы; достоверность контроля повышается при синхронизме с системной службой времени и горячем резервировании. Однако актуальным остается вопрос контроля истинности выполнения непосредственно алгоритма, исполненного аппаратно. Обусловлено это тем, что традиционная пошаговая отработка процессором аналогично принудительному моделированию прохода по всем ветвям алгоритма в среде отладчика невозможна.

### Литература

1. Харченко В.С., Юрченко Ю.Б. Анализ структур отказоустойчивых бортовых комплексов при использовании электронных компонент Industry // Технология и конструирование в электронной аппаратуре. – 2003. – № 2. – С. 3-10.

2. Юрченко Ю.Б. Повышение надежности бортовых управляющих комплексов путем построения мажоритированных структур на основе аппаратной синхронизации однокристалльных микроконтроллеров // Космічна наука і технологія. Додаток. – К.: НАНУ. – 2004. – Т. 10, № 1. – С. 41-49.

3. Algirdas A. Toward Systematic Design of Fault-Tolerant Systems. – [Электрон. ресурс]. – Режим доступа:

[www.md.kth.se/RTC/SC3S/papers/avizienis.pdf](http://www.md.kth.se/RTC/SC3S/papers/avizienis.pdf).

4. Kim H., Shin K.G. Evaluation of fault tolerance latency from real-time application's perspectives // Computers, IEEE Trans. – 2000. – Vol. 49, Issue 1. P. 55-64.

5. GUARDS: a generic upgradable architecture for real-time dependable systems / D. Powell, J. Arlat, L. Beus-Dukic ets. // Parallel and Distributed Systems, IEEE Trans. – 999. – Vol. 10, Issue 6. – P. 580-599.

6. Prizant J. High speed communicator for fault tolerant systems Digital Avionics Systems Conference // 1998. Proceedings 17<sup>th</sup> DASC. The AIAA/IEEE/SAE. – 1998. – Vol. 1, P. D25/1 -D25/7.

7. Randell Brain: System Structure for Software Fault Tolerance. – [Электрон. ресурс]. – Режим доступа:

[www.cs.ncl.ac.uk/research/pubs/inproceedings/papers/341.pdf](http://www.cs.ncl.ac.uk/research/pubs/inproceedings/papers/341.pdf).

8. Язык Verilog. Описание. – [Электрон. ресурс]. – Режим доступа: [www.actel.com](http://www.actel.com).

9. Пакет MAX+plusII. – [Электрон. ресурс]. – Режим доступа: [www.altera.com](http://www.altera.com).

Поступила в редакцию 22.02.2006

**Рецензент:** д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Харьков.