

УДК 004.415.2.031.43

## ЭФФЕКТИВНАЯ РЕАЛИЗАЦИЯ И КОНТРОЛЬ ИНТЕРВАЛОВ ВРЕМЕНИ В СПЕЦИАЛИЗИРОВАННЫХ ЦИФРОВЫХ СИСТЕМАХ

Н. Б. Преображенский, Ю. А. Холопов, Д. А. Гаврилов

Институт точной механики и вычислительной техники имени С.А.Лебедева  
Российской академии наук, 119991, Москва, Ленинский проспект, д.51

Статья поступила в редакцию 30 июля 2019 г.

**Аннотация.** На примере реализации цифровых систем управления, обсуждается необходимость перераспределения функций обработки данных между программными и аппаратными компонентами систем. Показаны возможные решения, обеспечивающие значительное повышение интероперабельности компонент цифровых систем реального времени. Отмечен ряд преимуществ аппаратно-программной реализации взаимодействия с внешними устройствами системы.

**Ключевые слова:** циклическая обработка информации, функции программной и аппаратной обработки данных, аппаратное взаимодействие с внешними устройствами.

**Abstract.** Digital real-time system, investigated in our work, executes cyclically repeating data processing. Disadvantages of software implementation of such systems on the example of digital control systems are given. We propose a new distribution of functions between software and hardware components of the system. Possible hardware solutions for interaction with external devices of such a system - sensors and actuators are discussed. Hardware support can significantly speed up data exchange with multiple peripherals. The interoperability of real-time digital systems components is increased.

**Key words:** cyclic information processing, functions of software and hardware data processing, hardware interaction with external devices.

### 1. Введение

Сложившаяся практика использования универсальных компьютеров в качестве вычислительного ресурса во встраиваемых цифровых системах

реального времени (обработка цифровых информационных потоков, системы технического зрения, цифровое управление объектами различной сложности), кроме кажущихся удобств от программной реализации алгоритмов, обладает рядом недостатков, на которые, почему-то не принято обращать внимания. Эти недостатки связаны, прежде всего, с тем, что в архитектуре универсальной вычислительной машины отсутствуют средства эффективной и точной обработки и учета длительности интервалов времени, что значительно ограничивает интероперабельность компонент системы, в которую встраивается универсальная вычислительная машина. Рассмотрим эти проблемы на примере цифровой системы управления (ЦСУ) некоторым объектом, где соблюдение цикличности обработки информации об объекте и необходимость контроля времени очевидны.

## **2. Недостатки программно реализуемой цифровой системы**

Взаимодействие вычислительного ресурса с объектом управления осуществляется посредством подключенных к ним датчиков, считывающих характеристики объекта и окружающей среды, а также исполнительных устройств, подключенных к компьютеру и передающих управляющие воздействия на объект. Датчики и исполнительные устройства (актуаторы) подключаются к компьютеру через специальные схемы сопряжения (согласования). Все эти устройства (датчики и актуаторы) являются внешними (ВУ) по отношению к компьютеру и управляются программным путем. Согласованный обмен командами и данными между универсальным компьютером и внешним устройством осуществляется, как правило, программно, посредством запросов, порождаемых процессором и ответов от ВУ, о готовности которых ВУ сигнализирует процессору, через механизм прерываний. Рассмотрим в общем виде затраты времени на выполнение прерывания. В компьютере, под управлением операционной системы (ОС), выполняется прикладная программа – программа управления объектом. В ходе выполнения программы осуществляется обращение к внешнему устройству, измеряющему некий параметр объекта управления.

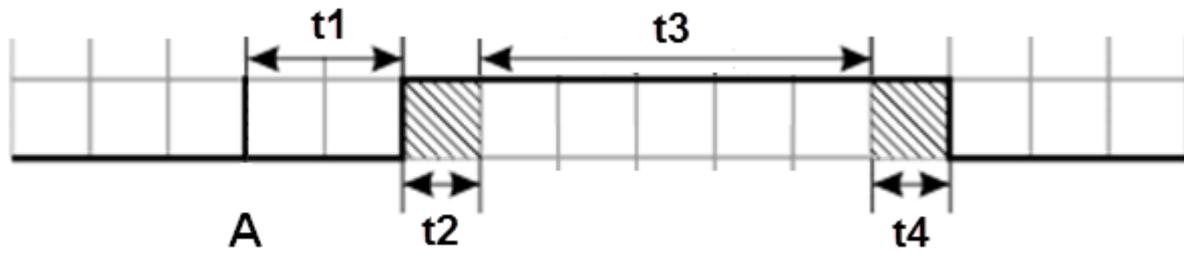


Рис. 1. Временная диаграмма выполнения прерывания.

На рисунке 1 представлена временная диаграмма, показывающая отрезок времени выполнения программы с прерыванием, в ходе которого осуществляется обмен информацией с внешним устройством (датчиком).

$A$  – момент времени формирования запроса на выполнение прерывания;

$t1$  – интервал времени ожидания момента, когда реально можно остановить выполнение программы и обработать прерывание;

$t2$  – интервал времени выгрузки контекста исполняемой программы и загрузки программы выполнения прерывания;

$t3$  – интервал времени непосредственно обработки прерывания – время взаимодействия с внешним устройством;

$t4$  – интервал времени, в течение которого идет восстановление контекста прерванной программы и передача управления, по адресу её продолжения.

Необходимый системе управления параметр объекта будет получен для обработки программой с задержкой  $T$ , равной сумме времен показанных интервалов:

$$T = t1 + t2 + t3 + t4 \quad (1)$$

Заметим, что в реальной системе управления используются десятки, а иногда и сотни внешних устройств (датчиков и актуаторов), посредством которых компьютер связан с объектом (и окружающей средой). Все эти устройства должны быть задействованы в каждом, повторяющемся с заданной частотой, цикле управления. Поэтому, сокращение и точная оценка времени взаимодействия ЦСУ с объектом, а именно, оценка времени выполнения работ с внешними устройствами имеет важное значение. Уменьшение суммарного

времени взаимодействия процессора с ВУ может быть достигнуто несколькими путями:

- Использование возможностей ускорения обработки прерываний, предоставляемых специализированными операционными системами реального времени (ОС РВ), мы можем значительно сократить время ожидания  $t1$ , а так же обеспечить сокращение времени обработки контекста программы – интервалы времени  $t2$  и  $t4$ .
- Выбирая более современный компьютер, с высокой частотой, мы сокращаем время выполнения команд в нем – очевидно, уменьшается  $t1$ ,  $t2$  и  $t4$ .
- Используя многоядерный вычислитель, мы можем организовать обработку прерываний на одном из ядер без манипуляций с контентом программы (это не так просто, как кажется), при этом интервалы времени  $t1$ ,  $t2$  и  $t4$  могут значительно уменьшиться.

Приведенные выше оценки показывают, что выбор более современного быстродействующего компьютера и операционной системы способны уменьшить суммарное время выполнения прерывания. Собственно, по этому пути обычно и идут разработчики программных средств ЦСУ.

Отметим, что величина времени ожидания  $t1$  существенно зависит от момента возникновения запроса на прерывание (А) и выполняемых компьютером в данный момент процессов. Даже для ОС РВ оценка интервала времени  $t1$  имеет разброс в несколько десятков машинных тактов - от возможного минимального, до возможного максимального значения.

Однако значительную часть суммарного времени обращения к внешнему устройству составляет величина  $t3$ , которая зависит от характеристик внешнего устройства. Эта величина не может быть точно оценена программистом и заметно уменьшена за счет улучшения характеристик компьютера и операционной системы.

Взаимодействие компьютера с объектом управления осуществляется во временном интервале – цикле управления, который определяется (вычисляется,

назначается в ходе проектирования ЦСУ) исходя из задачи управления и характеристик объекта. Как было показано выше, обеспечить точное во времени взаимодействие с внешними устройствами средствами программного управления не представляется возможным. Даже если обращения к внешним устройствам будут сгруппированы в определенных участках программы управления, поддержаны стандартной аппаратной коммутацией, все равно опрос их будет вестись последовательно. Получение характеристик объекта управления, выдача управляющих команд на исполнительные устройства будут не одновременны и «размазаны» по времени выполнения цикла управления, будут иметь различные по величине смещения (по времени) относительно фазы начала очередного цикла. Сосредоточим внимание на организации циклического ввода данных.

### **3. Возможности аппаратной поддержки интероперабельности**

Для корректного восприятия состояния объекта управления предпочтительным является одномоментное получение всего множества его характеристик, своего рода моментальный снимок в каждом цикле. Как было показано выше, организовать одновременное считывание множества характеристик объекта, то есть программным путем одномоментно опросить множество датчиков, не представляется возможным. Здесь совершенно уместна аналогия с фотоснимком – необходимо «заснять» состояние объекта, чтобы обеспечить максимальную интероперабельность, при дальнейшем выполнении расчетов. Время выполнения опроса программным способом является в общем случае суммой времен опросов всех датчиков.

$$T_{опр} = \sum T_i \quad (2)$$

При этом, так как  $T_i$  - длительность опроса каждого датчика не может быть строго фиксирована, то и общее время выполнения опроса может быть оценено в определенных пределах (больше - меньше). Таким образом, программный способ организации опроса датчиков не позволяет получить «мгновенный

снимок» состояния объекта, не только в связи с последовательным порядком их опроса, но и не гарантирует стабильных задержек в получении данных.

Для произвольного вычислительного процесса, обращения к внешним устройствам, как правило, не привязаны к какому-то расписанию, более того, это время обращений как бы выносится за скобки счетной части решения задачи. В задачах цифрового управления с опроса всех датчиков начинается каждый цикл. Поэтому функционал опроса является обязательным, многократно повторяющимся, однотипным действием, предшествующим вычислениям, выполняемым в каждом цикле. Оптимальная длительность цикла в общем случае определяется инерционными свойствами объекта управления. Опрос датчиков, выполнение вычислений, передача управляющих воздействий на исполнение и их отработка объектом управления суммарно должны укладываться в длительность цикла. Именно показанная выше нестабильность задержек во времени порождает необходимость при написании программы управления сложным объектом выполнять программным путем дополнительные действия по контролю времени.

Существенно уменьшить время, затрачиваемое на опрос датчиков, сделать опрос датчиков одновременным, а не последовательным можно, обеспечив аппаратную поддержку этой процедуры. Предположим, что в ЦСУ получение данных от всех датчиков одновременно выполняется по специальной команде. Рассмотрим возможность расширения состава команд компьютера дополнительной командой одномоментного опроса некоторого заранее известного фиксированного множества внешних устройств. Аппаратная реализация данной команды предполагает создание специализированного многоканального коммутатора, который за фиксированный интервал времени (один или несколько тактов) осуществляет опрос и последующую передачу данных в компьютер. Предполагаемое аппаратное решение, в зависимости от особенностей реализуемого проекта может иметь несколько различных аппаратно-алгоритмических реализаций.

Вариант 1. Предложенное выше расширение состава команд процессора, например, может быть реализовано аппаратно в виде IP-модуля, выполняющего команду параллельно организуемого обмена. Реализация команды может предполагать запись данных в специальный регистровый буфер или доступ к регистровому файлу или памяти процессора. Подобный подход фактически предполагает целевое проектирование периферийной подсистемы процессора, и может затрагивать проектирование других его узлов.

Вариант 2. Предлагаемый IP-модуль может быть реализован в виде независимой сопроцессорной схемы, внешней по отношению к процессору. В системе команд современных микропроцессоров, например, в ARM, присутствует специально выделенная команда обращения к сопроцессору. При этом функциональная сущность сопроцессора не фиксирована, а организация обмена данными с ним предполагается через регистровое пространство или через общую память. Такое аппаратное решение не затрагивает базовые возможности процессора, взятого за основу разработки. Однако заложенные в базовую архитектуру решения могут при проектировании сопроцессора потребовать выполнения определенных интерфейсных правил.

Вариант 3. Многие современные универсальные процессоры имеют канал быстрого прямого обмена внешнего устройства с памятью процессора. Представим многоканальный коммутатор, который в автоматном режиме опрашивает все подключенные к нему внешние устройства и формирует массив (пакет) для последующей передачи непосредственно в память процессора. Такое аппаратное решение является внешним по отношению к вычислительному ресурсу и совершенно не затрагивает базовый процессор. Событие запуска для циклической работы коммутатора может порождаться однократным в каждом цикле обработки программным прерыванием от процессора. При определенных условиях работы такой системы можно представить и независимое асинхронное автоматическое функционирование коммутатора, не требующее программного прерывания от процессора.

На ранних этапах развития вычислительной техники производительность вычислительного устройства обоснованно считалась основным показателем решения той или иной задачи. Работа по подготовке, предобработке и постобработке данных, как правило, выносилась за скобки затрат на выполнение задачи. Расширение прикладных областей применения вычислительных устройств, развитие и востребованность встраиваемых решений в значительной степени изменили взгляд на назначение, специализацию и способ использования цифровой аппаратуры.

Описанные выше проблемы программной обработки данных, затруднения при соблюдении временных соотношений процесса обработки свойственны не только системам цифрового управления. Эти проблемы, в той или иной мере, приходится решать при проектировании систем реального времени: систем потоковой обработки, циклического контроля, анализа и сжатия данных [1], систем автоматического обнаружения и сопровождения объекта интереса в видеосигнале, систем технического зрения (СТЗ) [2], везде, где существует необходимость циклического взаимодействия с комплексом внешних устройств и фиксированная во времени алгоритмическая обработка полученных данных.

#### **4. Заключение**

Идея реализации специализированного препроцессора подготовки данных не нова, однако в контексте задач проектирования встраиваемых систем, систем потоковой обработки данных, цифровых систем управления аппаратная поддержка эффективного взаимодействия с периферией без участия процессора в научной периодике практически не рассматривается. Представленные здесь в общих чертах варианты аппаратной поддержки циклической обработки данных в ЦСУ и СТЗ, опробованные на ряде моделей и стендовых реализаций [2, 3], позволили сделать вывод о преимуществах данного подхода:

- Аппаратное решение позволяет структурировать задачу, архитектурно разделив ее на слабо зависимые функционально связанные между собой блоки со стандартизованными интерфейсными связями.
- Значительно упрощается написание, отладка программы обработки, так как из нее практически удалены все компоненты проверки готовности ВУ, контроля времени и порядка обращений к внешним устройствам, контроля длительности этапов цикла обработки, обработка прерываний и т.п. Программа содержит только вычислительную часть – реализацию расчетной части алгоритмов управления.
- Снижается требование к скорости процессора, так как на него возлагается только вычислительная задача, которая по сложности значительно меньше, чем в случае классической программно реализуемой ЦСУ.
- Минимизация числа прерываний (или полный отказ от их использования) снижает требования к используемой операционной системе.
- Аппаратное разделение процедур работы с периферией и вычислений позволяет независимо настраивать и наладивать узлы и компоненты цифровых систем, упрощает модификацию алгоритмов расчета, упрощает масштабирование системы.

### Литература

1. Гаврилов Д.А., Павлов А.В., Щелкунов Д.Н. Аппаратная реализация сжатия динамического диапазона цифровых изображений на ПЛИС Xilinx. Журнал радиоэлектроники [электронный журнал] . 2018. № 10. Режим доступа: <http://jre.cplire.ru/jre/oct18/6/text.pdf> DOI 10.30898/1684-1719.2018.10.6
2. Ч.Н. Дам, Ю.А. Холопов, Н.Б. Преображенский. Аппаратная реализация системы приема и передачи видеопотока с использованием алгоритма Витерби. // Труды 57-й научной конференции МФТИ: «Актуальные проблемы фундаментальных и прикладных наук в современном

информационном обществе». Радиотехника и кибернетика. - М.: МФТИ, 2014. -200с. - С.45. - ISBN 978-5-7417-0544-5

3. Ю.А. Холопов, Ле Ба Чунг, Нгуен Тхань Чунг. Согласованная информационная среда для высокодинамичной системы управления // Мехатроника, автоматизация, управление. 2016. № 12(17). С. 816-820. ISSN 1684-6427.

**Для цитирования:**

Н. Б. Преображенский, Ю. А. Холопов, Д. А. Гаврилов. Эффективная реализация и контроль интервалов времени в специализированных цифровых системах. Журнал радиоэлектроники [электронный журнал]. 2019. № 8. Режим доступа: <http://jre.cplire.ru/jre/aug19/12/text.pdf>  
DOI 10.30898/1684-1719.2019.8.12