

Мультипроцессорная информационная система на базе ПЛИС для контроля положения элементов манипулятора с управляемым изгибом¹

Богданов Д.Р.* , Даринцев О.В.**,**

* Институт механики им. Р.Р. Мавлютова УНЦ РАН, Уфа

** Уфимский государственный авиационный технический университет, Уфа

Детально рассмотрены ключевые моменты методики аппаратно-программного обеспечения приема и обработки информации, поступающей с МЭМС-датчиков положения в информационную систему манипулятора, построенного на базе звеньев с управляемым изгибом. Приведены отличия в процедуре построения мультипроцессорных информационных систем, базирующихся на новых программируемых интегральных логических схемах большой емкости, предоставляющих возможность реализации в них *soft*-процессоров. Показаны результаты качественного сравнения решений, полученных на основе схем с применением цифровых автоматов и на базе схем с *soft*-процессорами. В качестве примера рассмотрена структура разрабатываемой мультипроцессорной информационной системы и предложены варианты ее аппаратной и структурной реализации.

Ключевые слова: манипулятор с управляемым изгибом, МЭМС-датчики, ПЛИС, *soft*-процессор, мультипроцессорные системы

1. Введение

Первоначально мультипроцессорные системы (МС) появились как системы отдельных ЭВМ, которые связывались друг с другом через разделяемые ресурсы (шина, память, коммуникационные связи) для решения общей задачи через ее декомпозицию на отдельные подзадачи. Следующим этапом в развитии МС стало появление микросхем с несколькими процессорами на кристалле таких как МС традиционного вычислительного назначения (например, многоядерные процессоры на материнской плате персональных компьютеров), так и МС на базе процессоров с определенной функциональной направленностью типа обработки изображения (видеопроцессоры) или реализации нейронной сети (нейрочипы).

Появление и быстрое развитие программируе-

мых интегральных логических схем (ПЛИС) в последние два десятилетия внесло в практику построения МС свои коррективы. Мультипроцессорные системы, реализуемые в ПЛИС, могут быть построены на основе встроенных *hard*-процессоров или с использованием *soft*-процессоров.

В первом случае это относится к семействам ПЛИС, в которых на кристалле интегрируются два и более процессорных ядра (*hard*-процессоры). Одним из ведущих производителей ПЛИС фирмой *Altera* в настоящее время выпускаются микросхемы семейства *CYCLONE V*, *STRATIX 10*, *ARRIA 5*, *ARRIA 10*, в которых используются *ARM* процессоры. В ПЛИС самого развитого семейства *STRATIX 10* интегрировано четыре 64-битных процессора *Cortex-A53 MPCore*, работающих на частоте 1,5 ГГц. Кроме этого, в составе ПЛИС есть узел для обработки данных в формате с плавающей точкой [1].

Во втором случае используются, так называемые *soft*-процессоры, которые могут быть реализованы как в ПЛИС, имеющих *hard*-процессоры, так и в ПЛИС без них. Фирма *Altera* предлагает софт-

¹Работа выполнена при финансовой поддержке РФФИ (проект № 15-08-02938-а) и Программы Президиума РАН № П-31.

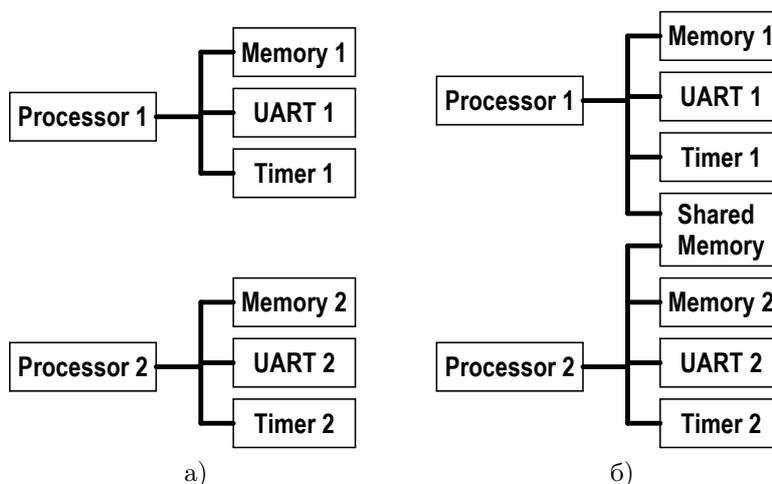


Рис. 1. Мультимикропроцессорная система (а) с автономными процессорами; (б) с разделяемой периферией

процессор под названием *NIOS II*, представляющий собой описание 32-разрядного процессора на языке описания аппаратуры *HDL (Hardware Description Language)*. В настоящее время есть две версии этого процессора, отличающиеся быстродействием и функциональными возможностями [2].

При этом конечная аппаратная реализация процессора, после выполнения этапа компиляции проекта средствами проектирования, зависит от выбранного семейства ПЛИС в соответствии с его внутренней архитектурой: структурой логических ячеек, наличием и структурой памяти, типом умножителей, матрицей соединений и т.д. Количество использованных ресурсов в ПЛИС и быстродействие синтезированного процессора могут сильно различаться от семейства к семейству. Возможное количество *soft*-процессоров, синтезируемых на кристалле, зависит от типа и логической емкости выбранной ПЛИС. Как следует из руководства «*Creating Multiprocessor NIOSII Systems*» фирмы *Altera* [3] возможны два варианта построения мультимикропроцессорных систем: на базе автономных процессоров (рис. 1(а)) и на базе процессоров с разделяемой периферией (рис. 1(б)). Применение автономных процессоров исключает их непосредственное взаимодействие, системы более просты в реализации на кристалле и при их использовании возникает меньше трудностей. В случае реализации систем с процессорами с разделяемой периферией возникает ряд трудностей, связанных с доступом отдельных процессоров к общим ресурсам. Разрешение проблемы связано с применением как дополнительных аппаратных ресурсов, исключающих одновременный доступ к общим ресурсам (арбитраж), так и написанием соответствующего программного кода для этих процессоров.

В общем случае в классических МС используются, как правило, одинаковые процессорные единицы (либо их функциональное разнообразие ограничено четко выделенным набором решаемых задач) и под решаемую задачу обычно подстраивается топология всей системы в целом. В ПЛИС возможно построение мультимикропроцессорных систем на базе процессоров со своим набором периферийных устройств, объемом памяти данных и команд, ориентированных на решение разных задач. При этом оптимизируются затрачиваемые ресурсы ПЛИС, быстродействие и потребляемая мощность.

2. Постановка задачи

Одной из задач в разработке манипулятора на базе звеньев с управляемым изгибом является задача управления положением его рабочей точки в пространстве. На данном этапе исследований проводится работа с одним звеном подобного манипулятора и решается задача определения его реального изгиба, так как наличие тросов и нежесткого контакта элементов звена приводит к отклонениям его положения от расчётного, полученного из кинематической модели [4] по измеренным значениям перемещений тросов [5].

Для решения этой проблемы, а также для исследования динамических свойств звена, в звено манипулятора были установлены шестиосевые датчики ускорения и угловой скорости, конструктивно выполненные в виде МЭМС-датчиков [5, 6]; датчики были установлены на нижние элементы звена. Информационной системе манипулятора требуется получить с них данные во время движения звена и произвести их обработку. При этом работа датчиков осуществляется асинхронно, поэтому, для большей достоверности получаемых с датчиков данных,

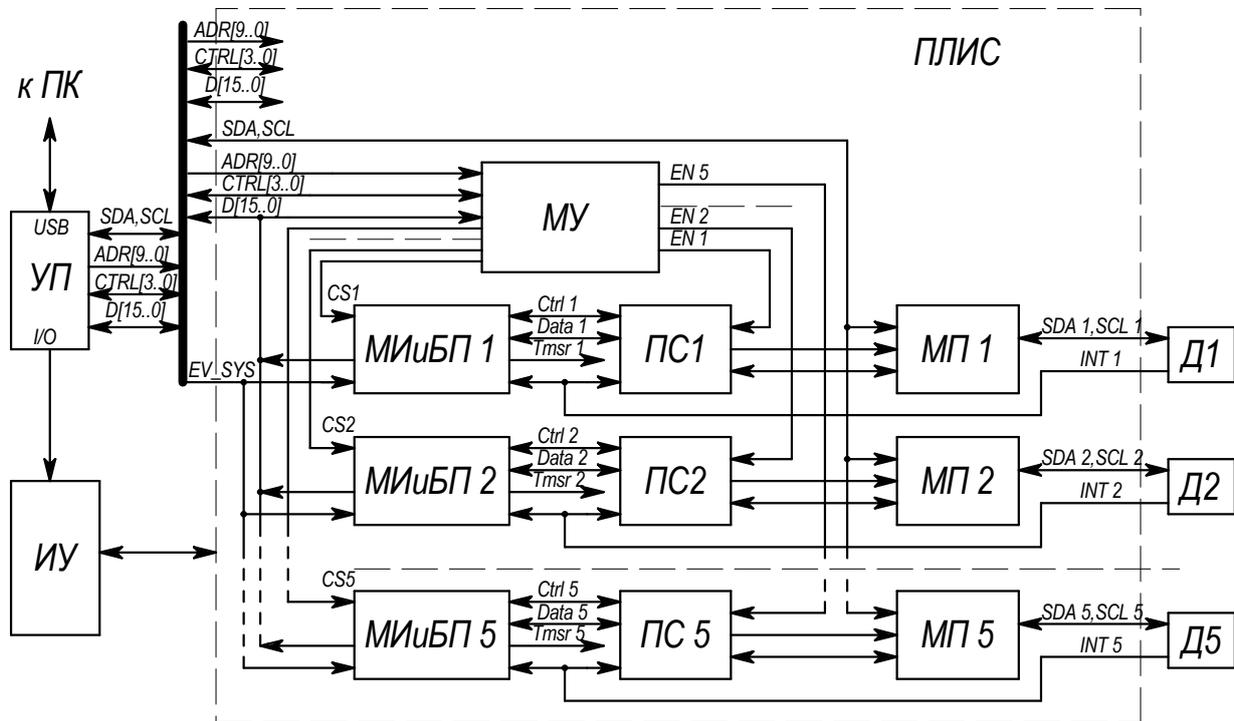


Рис. 2. Структурная схема МПС для приема и обработки данных с датчиков ориентации

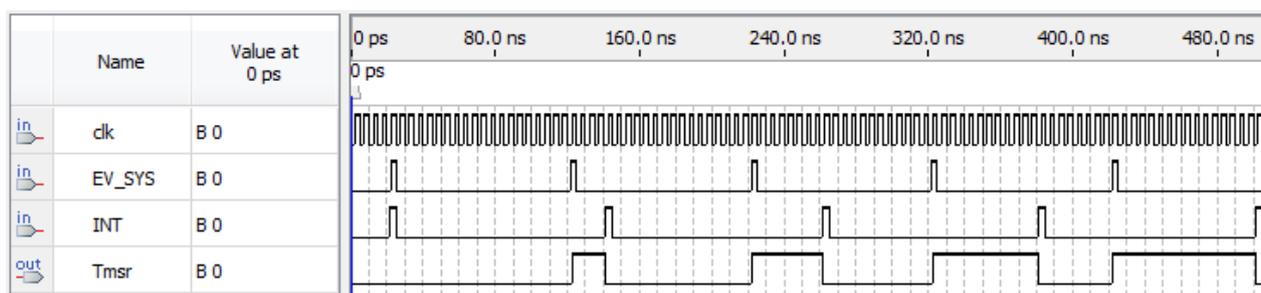
корректной работы системы управления, необходима их синхронизация по отношению к единому системному событию.

3. Методика реализации мультипроцессорной информационной системы

На начальных этапах задача ввода данных с датчиков решалась авторами на базе построения автоматов, реализующих интерфейс *I2C* и опрашивающих датчики ориентации по жесткому, ограниченному в функциональном плане алгоритму [5, 6]. При этом автоматы осуществляли только опрос датчиков, оставляя процесс обработки полученной информации управляющему процессору (УП), что сказывалось на быстродействии системы управления в целом. Анализ затрат ресурсов ПЛИС, привлекаемых для решения поставленной задачи с использованием *soft*-процессоров, показал, что при их незначительном росте возможно значительно разгрузить УП и реализовать, по необходимости, интерфейс *I2C* в полной мере путем модификации программы, что значимо с точки зрения синтеза внутренней структуры ПЛИС. Программная часть задачи синхронизации данных с датчиков обеспечивается той же синтезированной схемой системы, с сохранением в неизменном виде всех отлажен-

ных характеристик проекта (затраченные ресурсы, топология размещения базовых элементов кристалла, быстродействие). На рис. 2 показана структура мультипроцессорной системы на кристалле ПЛИС, синтезированной для решения поставленной задачи. Остальные элементы информационно-управляющей системы манипулятором для наглядности рассматриваемого вопроса на рисунке не показаны. Исполнительные устройства (ИУ) системы управления звеном манипулятора показаны в виде условного графического обозначения. УП подключен к персональному компьютеру верхнего уровня (ПК), который необходим для визуализации обработанных данных.

Как видно из рис. 2, вместо описанных в [5] автоматов опроса датчиков МЭМС Д1–Д5, в схеме установлены процессорные системы ПС1–ПС5. Посредством интерфейса *I2C*, который формируется программным путем, происходит обмен данными между ПС1–ПС5 и Д1–Д5 через мультиплексоры МП1–МП5, соответственно. В модулях измерения и буферизации данных (МИИБД1 МИИБД5) осуществляется формирование измерительных интервалов $Tmsr1$ – $Tmsr5$, преобразование их в двоичный код и передача в процессорные системы ПС1–ПС5 для дальнейшей обработки. Формируются сигналы $Tmsr1$ – $Tmsr5$ как интервалы времени меж-

Рис. 3. Результат моделирования схемы формирования сигнала *Tmsr*

ду событием *EV_SYS* (общий сигнал синхронизации системы управления) и событиями *INT1–INT5* (сигналы прерывания на выходах Д1–Д5, которые генерируются по мере готовности данных и являются асинхронными по отношению друг к другу и событию *EV_SYS*). Принцип работы схем формирования этих сигналов не отличается от описанных в [5, 6]. На рис. 3 показан результат работы схемы формирования сигнала *Tmsr* путем моделирования работы МИБД средствами пакета проектирования *QUARTUS*. Из-за неравенства периодов асинхронных сигналов *EV_SYS* и *INT* возникает их смещение друг относительно друга, что и фиксируется схемой.

Далее эта информация используется для приведения полученных данных к моменту времени появления сигнала *EV_SYS* путем программной обработки в ПС1–ПС5. Обработанные данные с Д1–Д5 записываются в память типа *FIFO* в МИБД–МИБД5, откуда по мере необходимости они считываются УП. Мультиплексоры МП1–МП5 позволяют УП последовательно подключаться к акселерометрам в режиме инициализации и отладки. Модуль управления (МУ) обеспечивает взаимодействие УП с процессорными системами ПС1–ПС5.

Архитектура процессорных систем ПС1–ПС5 формируется на базе библиотечного *soft*-процессора *NIOSII* с помощью инструмента *Qsys* пакета проектирования *QUARTUS* фирмы *Altera*. Результат синтеза в виде внутренней структуры ПС представлен на рис. 4.

Рассмотрим подробнее элементы процессорной системы и их назначение. Основой системы является компонент *nios2_qsys*: 32-разрядный *soft*-процессор *NIOSII*, который работает на базе собственной системы команд [2]. Программа для него может быть написана на языках *C* и *C++* с помощью среды программирования *NIOS_Eclipse*, которая также входит с систему проектирования *QUARTUS*. Полученный исполняемый программный код **.elf* (*executable linked file*) хранится в компоненте *onchip_mem*, который представляет со-

бой *RAM* память, где также хранятся данные. Параметры памяти (тип, объем, назначение) настраиваются при начальной компоновке процессорной системы. Компонент *clk* формирует системные сигналы синхронизации и сброса ПС и устанавливается автоматически при создании проекта системы в *Qsys*. Компонент *jtag_uart* служит для подключения ПС к системе проектирования для отладки разрабатываемой программы. Этот компонент при компиляции проекта в ПЛИС подключается к его встроенному порту *JTAG*, который, в свою очередь, подключается через кабель программирования к компьютеру, на котором в среде программирования *NIOS_Eclipse* отлаживается проект. Для ввода и вывода сигналов за пределы ПС используется библиотечный компонент *PIO* (*parallel input/output*). В данной системе три таких компонента, которые работают в следующих режимах: *prio_events* в режиме ввода сигналов (*input*), *prio_controls* в режиме вывода сигналов (*output*) и *prio_msr*, настроенный как порт с отдельным вводом и выводом сигналов (*bidir*). Через порт *prio_msr* осуществляется ввод измеренного сигнала *Tmsr* в двоичном эквиваленте и запись полученных и обработанных данных с акселерометра в память *FIFO*. Все управляющие сигналы и выходные сигналы интерфейса *I2C* формируются на выходе компонента *prio_controls*. Входные сигналы интерфейса *I2C* и сигналы управления от УП поступают через компонент *prio_events*, некоторые из входных линий которого являются источниками прерываний в ПС. Соответствующие настройки этих компонентов делаются во время конфигурирования ПС и во время работы изменены быть не могут.

4. Заключение

Построенную МС можно отнести к классу иерархических мультипроцессорных систем. В нее входит пять отдельных однотипных ПС нижнего уровня на базе *soft*-процессоров для ПЛИС и один управляющий процессор верхнего уровня, с которым ПС обмениваются данными независимо че-

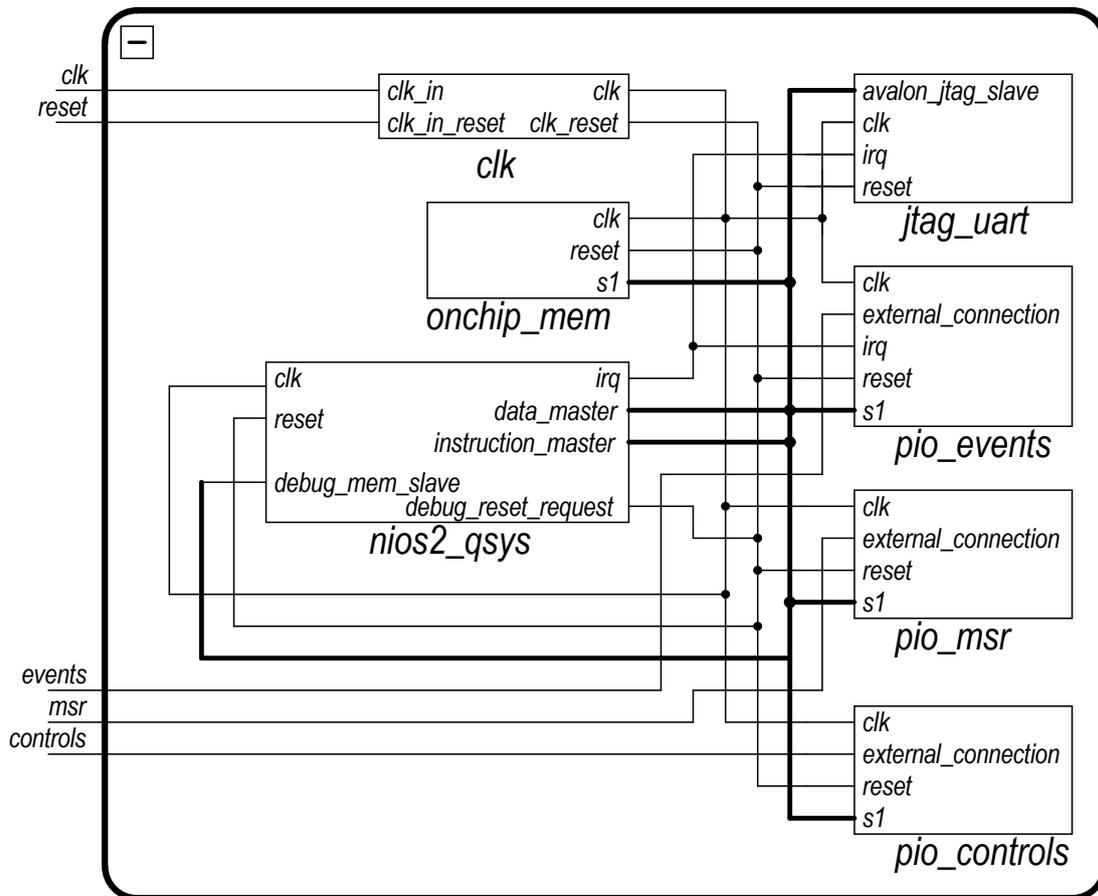


Рис. 4. Внутренняя структура синтезированной ПС на базе soft-процессора NIOSII

рез модули буферной памяти типа *FIFO* для исключения арбитража. Для физической реализации МС была использована плата разработки *DEONano* фирмы *Terasic* на базе ПЛИС типа *Cyclone IV EP4CE22* фирмы *Altera*. УП в рассмотренной МС представляет собой 32-разрядный микроконтроллер *STM32F407* фирмы *STMicroelectronics*. Теоретически, при наличии достаточных ресурсов, УП может быть реализован на базе soft-процессора *NIOS* с необходимыми характеристиками. В данном случае выбор в качестве УП отдельного микроконтроллера обусловлен наличием у него развитой периферии в виде различных стандартных интерфейсов и таймеров, а также готовой библиотеки подпрограмм для работы с ними. Также возможна реализация модуля МИИБД в виде компонента процессорной системы, что может снизить затрачиваемые ресурсы ПЛИС. Кроме того, возможно построение МС с процессорными системами и управляющим процессором, объединенными одной общей процессорной шиной и разделяющими одну память данных. Такое решение поддерживается набором библиотеки *IP*-компонентов пакета про-

ектирования *QUARTUS* и инструментом *Qsys*. Таким образом, выбор структуры МС является достаточно нетривиальной задачей и зависит от многих факторов: имеющихся ресурсов выбранной ПЛИС, сложности решаемой задачи и ее декомпозиции на подзадачи, опыта разработчика системы, необходимости расширения системы в будущем под новые задачи и т.д.

Современные робототехнические устройства являются многофункциональными, многоцелевыми и имеют в своем составе множество разнородных датчиков и исполнительных устройств. Описанный подход при проектировании подобных устройств обеспечивает их высокую адаптивность к решению новых задач. Надо отметить, что разработка подобных систем предъявляет особые требования к квалификации проектировщика. Эти требования обусловлены тем, что при проектировании подобных систем стоит задача оптимизации (с точки зрения использования ресурсов, быстродействия, удобства использования системы в дальнейшем) их декомпозиции на аппаратную и программную части.

Список литературы

- [1] Stratix 10 Device Overview // URL: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/stratix-10/s10-overview.pdf (дата обращения 08.09.2016).
- [2] Nios II Gen2 Processor Reference Guide // URL: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/nios2/n2cpu-nii5v1gen2.pdf (дата обращения 08.09.2016).
- [3] Creating Multiprocessor Nios II Systems Tutorial // URL: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/tt/tt_nios2_multiprocessor_tutorial.pdf (дата обращения 08.09.2016).
- [4] Богданов Д.Р., Даринцев О.В. Кинематика манипулятора с управляемым изгибом на базе твердых элементов со сферической поверхностью // Мехатроника, автоматизация, управление». М.: Изд-во «Новые технологии». 2015. Т. 16, № 10. С. 671–677.
- [5] Богданов Д.Р., Даринцев О.В. Информационная поддержка интеллектуальной системы управления звеном манипулятора // Proceedings of the 2nd International Conference «Information Technologies for Intelligent Decision Making Support» and the Intended International Workshop «Robots and Robotic Systems», May 18-21, Ufa, Russia. 2014. Vol. 3. P. 260–264.
- [6] Богданов Д.Р. Алгоритмические и аппаратные способы получения информации о динамических свойствах манипулятора с управляемым изгибом // Восьмая Всероссийская мультиконференция по проблемам управления (МКПУ–2015): материалы 8-ой Всероссийской мультиконференции. 2015. Т. 2. С. 18–21.

Multiprocessor systems based on FPGA for receiving and processing data from the position sensors of the manipulator elements with controlled bending

Bogdanov D.R.* , Darintsev O.V.**

* Mavlyutov Institute of Mechanics, Ufa

** Ufa State Aviation Technical University, Ufa

Key moments of technique of reception and processing of information from the MEMS position sensors in the information system of the manipulator built on the basis units with controlled bend is discussed in detail. The differences in the procedure for construction of multiprocessor information systems based on the new programmable logic integrated circuits large capacity which provide the use of soft-core processors is presented too. The results of qualitative comparison of the solutions obtained by use state machine circuits and schemes based on soft-processors is shown. As an example, consider the structure developed multiprocessor information system and variants of its hardware and structural implementation.

Keywords: manipulator with a controlled bend, MEMS, FPGA, software-processor, multiprocessor systems

