

# ВСТРОЕННЫЕ ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СОВРЕМЕННЫХ МИКРОКОНТРОЛЛЕРОВ

**Аркадий Ключев**, К.Т.Н., ведущий специалист,  
**Алексей Платунов**, К.Т.Н., директор (ООО «ЛМТ», г. Санкт-Петербург)

**Наиболее сложным этапом в создании микропроцессорных систем, безусловно, является отладка. Этот процесс требует от специалиста знания разнообразных инструментальных средств и технологий.**

**Статья посвящена важной составляющей проектирования микропроцессорных систем – аппаратно-программным средствам внутрисистемного программирования и отладки современных микроконтроллеров.**

В настоящее время все большее число микроконтроллеров (МК) снабжается встроенными инструментальными средствами. Эти средства обеспечивают загрузку программ и данных в память целевой системы, отладку и тестирование целевых программ и аппаратуры (под целевой системой здесь и далее будем понимать микропроцессорную систему, которая решает прикладную задачу пользователя).

Внутрисистемное (ISP – In System Programming) или внутрисхемное программирование (ICP – In Circuit Programming) – это способ инициализации памяти МК (RAM, E2PROM, Flash) в составе целевой системы. Для программирования доступна память данных, память программ, конфигурационная память ПЛИС. Суть технологии состоит в том, что зависящая от целевой аппаратуры часть программатора встраивается в МК. Технология ISP направлена на то, чтобы освободить разработчика от необходимости программировать память вне целевой системы (например, с использованием программаторов ППЗУ). Преимуществами применения ISP являются увеличение надежности целевой системы (нет необходимости ставить на плату панель для МК), сокращение цикла разработки (нет необходимости в разработке прототипных плат с панелями, сокращается время цикла программирования и отладки), улучшение качества конечного изделия (отладка системы становится более комфортной и, следовательно, более качественной).

Отладку можно определить как процесс проверки корректности функционирования вновь спроектированной системы с целью выявления и исправления ошибок и несоответствий исходному техническому заданию. В статье мы будем говорить о внутрисхемной отладке (OCD, On Chip Debugging), то есть отладке с

помощью специальных инструментальных средств, встроенных в МК.

Тестирование – это проверка корректности функционирования системы с целью выявления ошибок. Тестирование является частью отладки. Тестирование программного обеспечения и аппаратуры обычно стремятся автоматизировать, так как при тестировании фиксируется реакция системы на внешние сигналы и сверяется соответствие результатов работы со спецификацией.

До недавнего времени задачи загрузки в целевую систему, тестирования и отладки исполняемого кода и данных решались следующими способами:

1. многократное перепрограммирование ПЗУ;
2. использование программных мониторов-отладчиков, находящихся в ПЗУ целевой системы;
3. использование симуляторов;
4. использование внутрисхемных эмуляторов.

Первый способ используется очень редко. Он неэффективен, так как требует большого количества времени на перепрограммирование ПЗУ. Кроме того, требуется специальная прототипная плата с панелями для установки ПЗУ.

Второй способ достаточно популярен среди разработчиков. Его достоинствами являются простота и низкие накладные расходы. К недостаткам можно отнести невозможность полноценного тестирования аппаратуры целевой системы и сложность отладки системы в реальном времени.

Третий способ позволяет отлаживать систему в реальном времени. К сожалению, при отладке используется идеальная модель окружения, всегда немного отличающаяся от реальной целевой системы. Из-за этого несоответствия симуляторы используют только на начальных стадиях отладки.

Применение внутрисхемных эмуляторов позволяет решить почти все проблемы, связанные с отладкой и тестированием программного обеспечения и аппаратуры. К сожалению, крупным недостатком внутрисхемных эмуляторов является их высокая цена.

Чтобы сделать рентабельным производство малых партий изделий или создание изделий небольшими коллективами разработчиков, производители современных МК предлагают множество разных средств внутрисхемного тестирования, отладки и программирования. Улучшение ценовых показателей инструментальных систем происходит благодаря тому, что аппаратно-зависимые части внутрисхемных эмуляторов, программаторов и систем тестирования переносятся непосредственно в МК. Внешняя часть программаторов и эмуляторов существенно упрощается и становится относительно универсальной. За счет того, что МК выпускаются большими



**Промышленная  
и бортовая  
электроника**

## КОНТРОЛЛЕРЫ и СИСТЕМЫ

*индустриальные, приборные,  
учебные, инструментальные*

**Программные средства**

RS485, CAN, Ethernet, USB, PC104

- Готовые решения
- Производство
- Разработка по ТЗ заказчика

**ООО "ЛМТ", С.-Петербург**

Тел./факс (812) 233-3096

info@lmt.ifmo.ru <http://lmt.ifmo.ru>

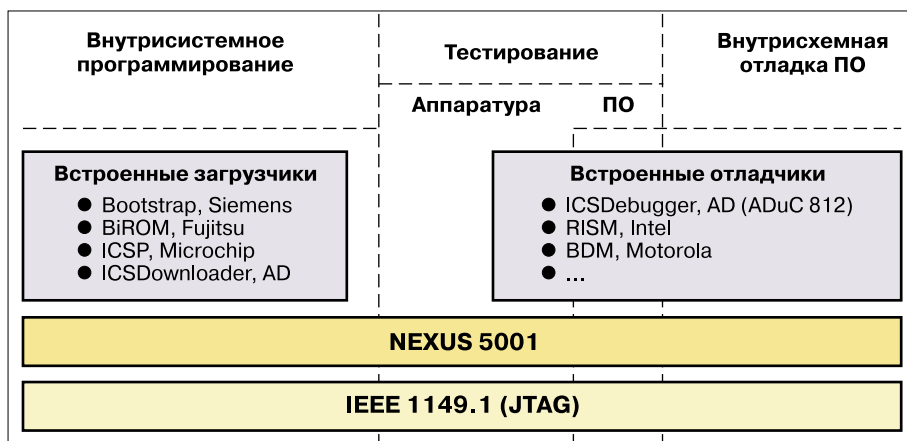


Рис. 1. Встроенные инструментальные средства

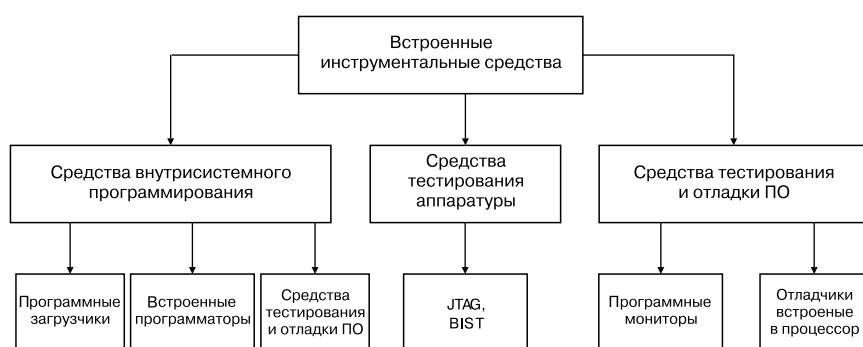


Рис. 2. Классификация встроенных инструментальных средств

тиражами, дополнительные блоки отладки и тестирования незначительно увеличивают цену микросхем.

На рисунке 1 показано, какие задачи позволяют решать различные встроенные инструментальные средства. Встроенные загрузчики решают задачи ISP, встроенные отладчики — задачи отладки и частично тестирования программного обеспечения (ПО). Механизм граничного сканирования JTAG [1,2] используется как в ISP, так и при отладке и тестировании в качестве универсального инструментального интерфейса. Средства, описанные в стандарте Nexus 5001 [4], также покрывают весь спектр инструментальных задач.

В таблице 1 приведены экспертные оценки качества тех или иных инструментальных средств для микроконтроллеров с различной разрядностью.

Встроенные загрузчики (ISP) встречаются в большинстве 8-разрядных МК с Flash-памятью. Средства тести-

рования и отладки в 8-разрядных МК встречаются пока достаточно редко.

Сравнительно простые отладчики встроены в 16-разрядные МК. В МК этой группы можно также встретить встроенные системы тестирования. Иногда системы ISP для этих МК существуют самостоятельно, но чаще всего они реализованы на базе встроенных систем тестирования и отладки.

Встроенные отладчики, соперничающие по набору своих возможностей с внутрисхемными эмуляторами, есть только в некоторых мощных 32-разрядных МК. Системы ISP в таких МК полностью базируются на средствах тестирования и отладки.

На рисунке 2 представлена классификация встроенных инструментальных средств (включая встроенные средства загрузки (программирования), тестирования аппаратуры, тестирования и отладки ПО).

Рассмотрим встроенные средства загрузки и отладки на примере прос-

тейших решений, которые характерны для большинства современных МК (специальные встроенные средства тестирования аппаратуры, такие как BIST (встроенная система само-тестирования) и другие, выходят за рамки данной статьи).

## СРЕДСТВА ВНУТРИСИСТЕМНОГО ПРОГРАММИРОВАНИЯ (ISP)

К основным функциям ISP можно отнести:

- загрузку данных в МК из инструментальной системы;
- программирование внутренней памяти программ и данных;
- стирание фрагментов внутренней памяти МК.

Встроенные загрузчики, как правило, весьма примитивны. В большинстве случаев их можно рассматривать как системы начального уровня. Для загрузки программ большого объема лучше использовать специальные средства доставки. Рассмотрим два варианта встроенных загрузчиков, отличающиеся по архитектуре и возможностям.

## Встроенные загрузчики на базе программаторов

Наиболее часто встречающийся вариант программатора — программная реализация загрузчика в ОЗУ или Flash. Загрузчик помещается в ПЗУ микроконтроллера и становится доступен после сброса, если подан сигнал на специальный вывод МК. Встречаются варианты, в которых запись программы пользователя производится только в ОЗУ (например, bootstrap loader фирмы Siemens или BiROM фирмы Fujitsu). Сразу после записи происходит передача управления. В некоторых системах (например, PIC фирмы Microchip, ADuC812 фирмы Analog Device) запись пользовательской программы производится сразу во Flash. Разными бывают и интерфейсы программирования; наиболее удобным является стандартный последовательный канал. В тех МК, где такой канал отсутствует, несколько выводов используется для подключения программатора (например, PIC16F84 или PIC16F87х фирмы Microchip) по уникальному интерфейсу. После программирования такие выводы можно использовать по их штатному назначению. На рисунке 3 представлена обобщенная модель МК со встроенным загрузчиком.

## Встроенные загрузчики на базе систем отладки и тестирования

Системы со встроенными отладчиками являются наиболее удобными и универсальными. Так как практически все отладчики позволяют обращаться к па-

Таблица 1. Зависимость набора встроенных инструментальных средств от разрядности микроконтроллера (оценка по 5-балльной шкале)

Инструментальные средства	Разрядность, бит		
	8	16	32
Программирование	5	5	5
Тестирование	3	4	5
Отладка	3	4	5

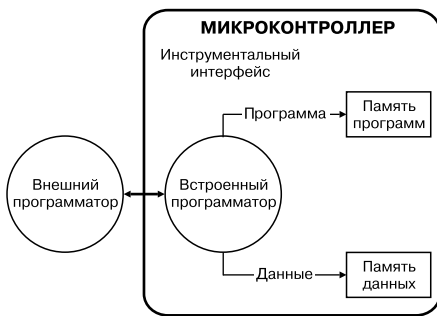


Рис. 3. Обобщенная модель микроконтроллера со встроенным загрузчиком

мемории целевой системы, есть возможность организации записи специальных кодовых последовательностей, необходимых для программирования Flash-памяти. Таким образом, инициализация системы происходит в два этапа: сначала при помощи отладчика во Flash записывается программа-загрузчик (программатор) Flash, затем при помощи загрузчика в память МК заносятся необходимые программы и данные.

Очень часто для начальной инициализации памяти МК используется интерфейс JTAG с соответствующим набором инструментальных программ (см. рис. 4).

Основные характеристики встроенных загрузчиков для распространенных МК приведены в таблице 2.

#### Встроенные отладчики

К основным функциям встроенного отладчика относятся:

- запуск и останов пользовательской программы;
- просмотр и редактирование содержимого основной и регистровой памяти.

В принципе возможно использование встроенного отладчика в качестве программатора Flash или EEPROM, если протокол обмена со встроенным отладчиком доступен для использования (например, RISM для МК MCSx96 фирмы Intel). Хотя такой способ программирования возможен, его нельзя назвать надежным и удобным. Как правило, протоколы взаимодействия со встроенным отладчиком не содержат средств контроля целостности передаваемых данных. Достаточно часто фирмы — производители МК скрывают протокол обмена со встроенным отладчиком. В результате конеч-

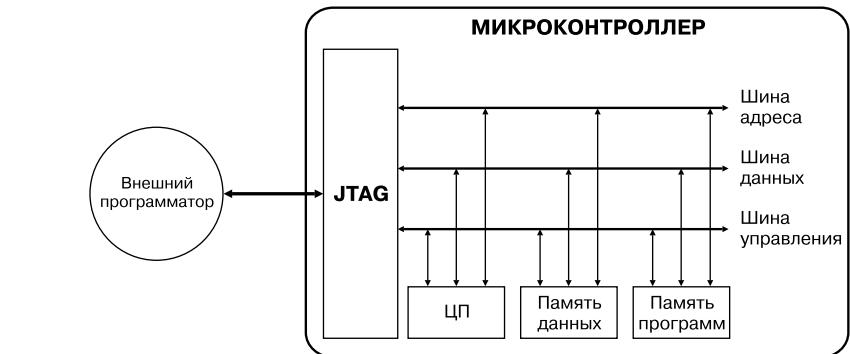


Рис. 4. Структура микроконтроллера с загрузчиком на базе интерфейса JTAG

ному пользователю доступны только готовые инструментальные средства третьих фирм-производителей ПО.

Механизм граничного сканирования (Boundary Scan) IEEE 1149.1-1990 (JTAG) [1, 2] обладает набором свойств, необходимых для проведения внутрисистемного программирования и тестирования аппаратуры. Достаточно трудно назвать JTAG надежным средством для программирования, так как протокол обмена не содержит средств контроля целостности данных. Механизм JTAG хорошо подходит для начальной инициализации платы, то есть доставки в память целевой системы надежного загрузчика. Производители МК обычно выполняют встроенный контроллер JTAG в виде одного из двух полярных вариантов. Первый можно назвать внешним, а второй — внутренним. Внешний JTAG позволяет получить доступ ко всем внешним выводам МК. Внутренний позволяет обеспечить доступ к внутрисистемной шине МК. Первый вариант предпочтительнее использовать в сравнительно сложных системах с внешней памятью, второй оптимален в простых системах со встроенной в МК памятью.

В некоторых процессорных ядрах МК JTAG позволяет осуществлять не только доступ к ячейкам памяти внутри процессора или к внешним выводам, но и позволяет проводить эффективную отладку программ с помощью дополнительного аппаратно-программного инструментария Embedded ICE (In Circuit Emulator). Примером такого решения является 32-разрядная архитектура ARM [3], которая приобретает все большую популярность у создателей МК, процессоров для встроенных применений и систем на кристалле (SOC).

Активные поиски решений в области технологий отладки систем реального времени и встроенных систем нашли свое отражение в перспективном стандарте инструментального канала Nexus 5001 [4]. В частности, в нем предложена классификация режимов работы отлаживаемой системы. Основу деления составляют статический и динамический режимы работы с целевой системой.

Статический режим подразумевает работу с целевой системой в тот момент, когда процессор остановлен. Во многих случаях встроенный отладчик интегрирован в процессор и его функции частично выполняются аппаратно, а частично программно. Таким образом, можно говорить о том, что в ряде систем на время работы в статическом режиме процессор реально не останавливается, а переключается с исполнения целевой программы на инструментальную.

Динамический режим подразумевает параллельную работу в реальном масштабе времени целевой программы и инструментальных средств. Параллельное исполнение команд целевой и инструментальной задач в рамках одного процессора — понятие достаточно условное. Такое исполнение связано с задержками из-за постоянного переключения контекстов, что отражается на времени исполнения. Поэтому реальный динамический отладочный режим возможен только при наличии встроенных в МК сложных дополнительных средств. Такими средствами являются инструментальный резидентный процессор с локальной памятью и последовательным или параллельным высокоскоростным инструментальным каналом. Сегодня такими инструментальными возможностями обладают МК фирм Motorola (смей-

Таблица 2. Встроенные средства доставки современных микроконтроллеров

Фирма-производитель	МК или семейство	Макс. объем доставляемых данных	Способ контроля целостности	Интерфейс	Тип памяти
Microchip	PIC 16F87x	8 Кслов	Чтение и сравнение	Синхронный последовательный	Flash E <sup>2</sup> PROМ
Infineon	C167	32 байт	Нет	UART	RAM
Fujitsu	F <sup>2</sup> MC-16LX	64 Кбайт	Контрольная сумма (1 байт)	UART	RAM
Motorola	M68HC12	Определяется объемом Flash	Чтение и сравнение	BDM	Flash

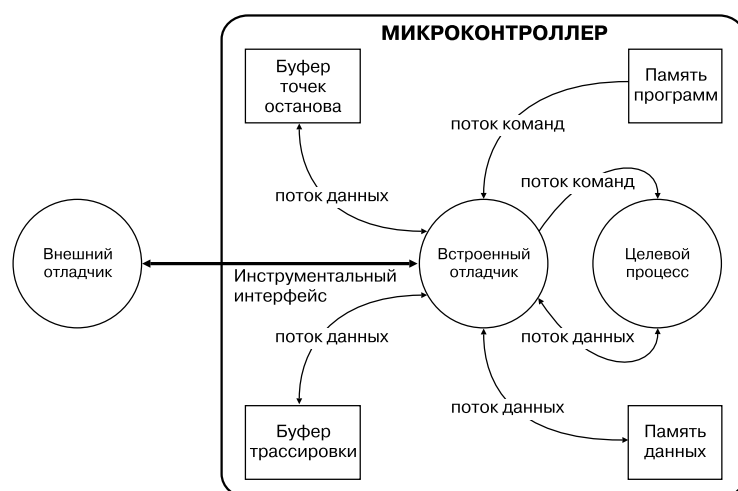


Рис. 5. Обобщенная модель микроконтроллера со встроенным отладчиком в режиме отладки

ства средней и высокой производительности MPC5xx, MPC82xx) и Infineon (семейство TriCore).

Отладчики, встроенные в различные МК, обладают достаточно широким диапазоном возможностей [5]. Попробуем выделить их основные характеристики. Для наглядности рассмотрим обобщенную модель МК со встроенным отладчиком.

Такая модель, представленная на рисунке 5, построена на базе потоковой модели данных (DFD). Кружками изображены процессы, стрелками — потоки данных, а прямоугольниками — хранилища данных.

Внутри МК выполняются два процесса: встроенный отладчик и целевой процесс. Снаружи МК есть еще один процесс — внешний отладчик, который решает высокоуровневые задачи управления встроенным отладчиком и организует интерфейс пользователя (ММИ). Мы не будем более подробно рассматривать устройство внешнего отладчика, так как это выходит за рамки данной публикации.

Внутренний и внешний отладчики соединены друг с другом инструментальным интерфейсом, который позволяет управлять внутренним отладчи-

ком, передавать и считывать данные в отладчик и из него. В рабочем режиме потоки данных и команд могут замыкаться непосредственно в целевом процессе. В режиме отладки потоки команд и данных идут транзитом через встроенный отладчик. Транзит потоков позволяет производить обработку данных в инструментальных целях в реальном масштабе времени. В частности, появляется возможность:

- анализировать данные с целью определения точки останова (watchpoint);

- анализировать адрес исполняемой инструкции с целью определения точки останова (breakpoint);

- накапливать данные и/или команды в буфере трассировки для последующего анализа;

- подменять потоки команд и данных, поступающих из памяти МК, на потоки команд и данных из памяти инструментальной машины (эмуляция памяти).

В таблице 3 представлены встроенные отладочные функции 16- и 32-разрядных МК ведущих фирм.

В заключение необходимо еще раз подчеркнуть важность и сложность современного инструментария, кото-

рый облегчает применение МК. Ситуация осложняется острой нехваткой литературы по данному вопросу, особенно на русском языке. Это заставляет разработчиков заново создавать инструментарий, зачастую «изобретая велосипед». Разработка собственных развитых инструментальных средств является сложной задачей, а эффективные покупные средства для большинства отечественных разработчиков по-прежнему дороги.

С другой стороны, разработка сложных и ответственных микроконтроллерных систем с помощью примитивного инструментария практически исключает использование современных технологий проектирования. Резко падает надежность проектирования, гарантированный результат удастся получить при создании только простых систем.

Состояние рынка инструментальных средств микропроцессорных систем в более широком плане обсуждалось авторами в [6].

## ЛИТЕРАТУРА

1. Платунов А., Постников Н., Чистяков А. Механизмы граничного сканирования в неоднородных микропроцессорных системах. М.: Chip News. 2000, №10, с. 8–13.
2. Ключев А.О., Коровьякова Т.А., Платунов А.Е. Использование интерфейса JTAG для отладки встраиваемых систем/ Изв. вузов. Приборостроение. 1998. Т41, №5, с. 45–50.
3. ARM Limited 1999 «Using Embedded ICE» URL: <http://www.arm.com/>.
4. The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface. Industry Standards and Technology Organization (IEEE-ISTO) <http://www.ieee-isto.org/>.
5. Craig A. Haller. The ZEN of BDM. Macraigor Systems Inc. URL: <http://www.macraigor.com/>.
6. Ключев А., Кустарев П., Платунов А. Инструментальные и учебные контроллеры семейства SDK/Компоненты и технологии. 2002, №5, с. 96–99.

Таблица 3. Встроенные отладочные функции некоторых современных МК

Встроенные отладочные функции	Микроконтроллер				
	Infineon TriCore	Infineon C166CBC	Motorola MPC 5xx	Motorola M68HC12	Intel MCSx96
Чтение и запись ячеек памяти	+	+	+	+	+
Запуск и останов программы	+	+	+	+	+
Исполнение инструкций	+	+	+	+	–
Точки останова по адресу	+	+	+	+	–
Точки останова по данным	+	+	+		
Мониторинг потока данных в реальном времени	–	–	+	–	–
Мониторинг потока команд в реальном времени	–	–	+	–	–
Эмуляция памяти	–	–	–	–	–
Мониторинг высокоуровневых процессов	–	–	–	–	–