

ИСПОЛЬЗОВАНИЕ ДВУЯДЕРНОГО ЦИФРОВОГО КОНТРОЛЛЕРА ВО ВСТРАИВАЕМЫХ ПРИЛОЖЕНИЯХ

ХАРША ДЖАГАДИШ (HARSHA JAGADISH), Microchip Technology

В статье рассматривается пример использования 16-бит цифрового сигнального контроллера dsPIC33CH. Приводятся достаточно убедительные аргументы в пользу его применения во встраиваемых приложениях.

Встраиваемые приложения все более усложняются. Нередко они используются для решения нескольких задач одновременно, а не ориентированы на одну из них, как это было еще относительно недавно. Широкое распространение интернета увеличивает коммуникативные возможности устройств, что также требует увеличения производительности и сложности микроконтроллеров, управляющих этими устройствами. Высокий уровень интеграции может потребовать участия в проекте специалистов разного профиля, решения которых будут интегрированы в один МК.

Разумеется, требуется уменьшить стоимость решения, для чего в микроконтроллер (МК) интегрируется несколько функций: интерфейс датчиков, управление электроприводом, преобразователями напряжения, а также защита данных и безопасность системы в целом. Если речь идет о силовом оборудовании, то предъявляются требования к повышению его КПД.

Именно более высокий КПД, достигающий 85–90%, стал одной из причин замены бесщеточными двигателями постоянного тока (BLDC) и синхронными двигателями с постоянными магнитами (PMSM) старых щеточных двигателей постоянного тока, КПД которых редко превышал 75–80%. Управление щеточным двигателем крайне простое, и с этой задачей может справиться любой простейший микроконтроллер (МК). Однако в настоящее время, когда на рынке предлагаются 16- и 32-бит недорогие специализированные МК, подобное обстоятельство уже не может считаться преимуществом.

Для управления бесщеточными электродвигателями BLDC и синхронными PMSM обычно используется векторное управление (FOC). Этот действительно более сложный метод, применение которого требует относительно серьезных вычислительных ресурсов. Для увеличения точности регулирования и уменьшения времени протекания переходных процессов, например при внезапном набросе нагрузки, необходимо увеличить частоту опроса параметров. Другими словами, для решения этой задачи необходимы немалые ресурсы процессора.

Рассмотрим круг задач, которые должны решать современные интеллектуальные встраиваемые устройства.

1. Основной функционал пользовательского приложения. В нашем случае это управление электроприводом и формирование интерфейса пользователя.
2. Практически любое современное интеллектуальное устройство немислимо без поддержки сетевых протоколов, поэтому в состав ПО должны входить стеки соответствующих протоколов.
3. Кроме того, необходимо обеспечить безопасность, конфиденциальность и целостность данных. Требуется поддержка аутентичности и контроля доступа для предотвращения взлома.

4. Возможны случаи, когда каждая из систем настраивается или масштабируется индивидуально в зависимости от объема управления или области применения.

Для решения указанных задач может потребоваться участие различных специалистов или даже разных команд. Причем, эти специалисты должны работать в тесной взаимосвязи. Сценарий еще больше усложнится, если эти специалисты найдутся в разных местах, а иногда и даже в разных странах.

Часто на разрабатываемую систему накладываются и другие ограничения, например по стоимости решения или по номенклатуре используемых компонентов, по массогабаритным показателям и т.д. Иногда у разработчиков возникает соблазн использовать несколько микроконтроллеров, каждый из которых станет решать только одну задачу. В этом случае упрощается разработка ПО разными специалистами, а их взаимодействие сводится к согласованию протокола обмена данными.

Однако, учитывая упомянутые ограничения, разработчикам далеко не всегда удастся облегчить себе жизнь. В таких случаях остается вроде бы очевидное решение – выбрать один-единственный высокопроизводительный МК с высокой степенью интеграции для решения всех задач. Но так ли это решение очевидно? Если учесть, что во встраиваемых системах МК приходится реализовывать сложные алгоритмы управления в реальном времени и одновременно выполнять несколько приложений, критичных ко времени исполнения, то выбор одноядерных, пусть и высокопроизводительных, микроконтроллеров кажется уже не столь очевидным.

В таком случае планировщик задач или операционная система реального времени (ОСРВ) могут и не помочь; к тому же, и планировщик, и ОСРВ тоже используют ресурсы МК. Задачу усложняет и интеграция ПО разных разработчиков – требуется модульная программная архитектура с динамическим распределением ресурсов. Даже если каждый из программных стеков корректно работает по отдельности, при интеграции могут возникнуть ошибки. Часть ресурсов отнимет время переключения контекста при переходе от одной задачи к другой. Отладка при этом превращается в кошмар для разработчиков.

В подобных обстоятельствах проще всего использовать многоядерный МК. Однако его стоимость может превысить заданные ограничения. Хорошо известно, что в некоторых приложениях микроконтроллер, обладающий высокой производительностью согласно всем известным тестам, может проигрывать своему менее производительному собрату. И это легко объяснить – никакие тесты не позволяют смоделировать все возможные ситуации, встречающиеся на практике.

В качестве примера рассмотрим двоядерный 16-бит цифровой сигнальный МК dsPIC33CH от Microchip Technology. Воз-

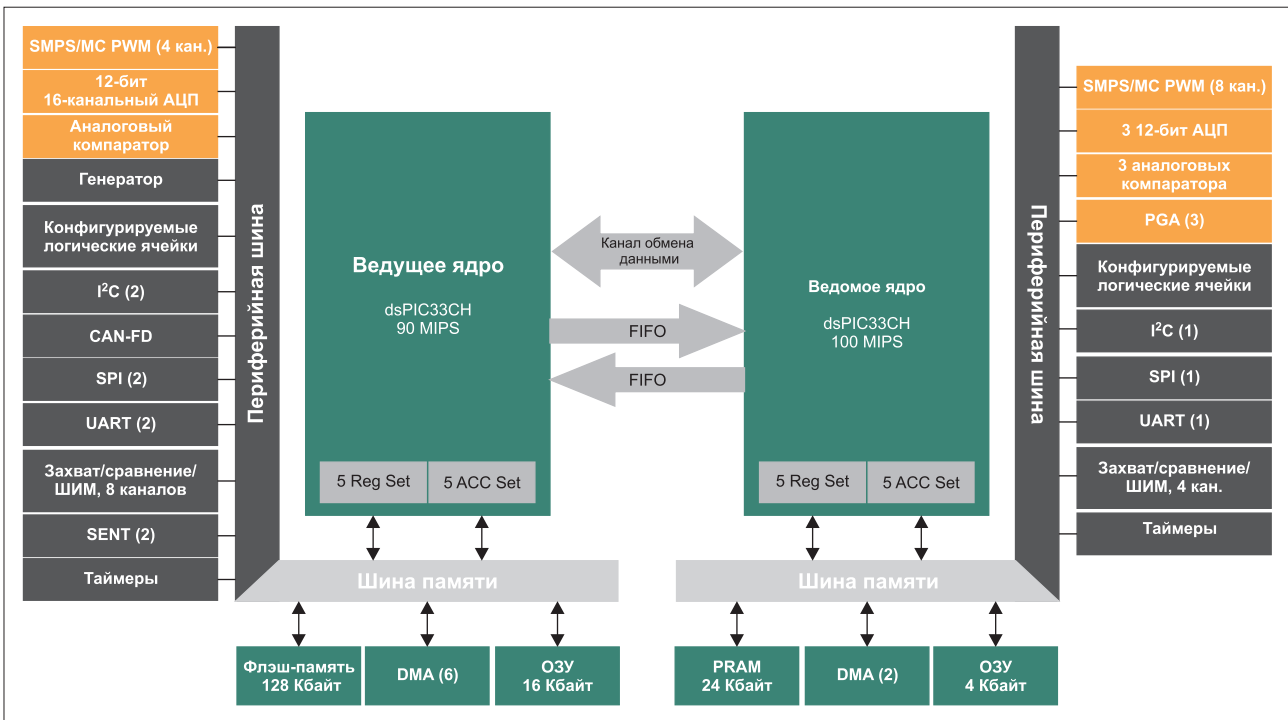


Рис. 1. Структурная схема МК dsPIC33CH от Microchip Technology

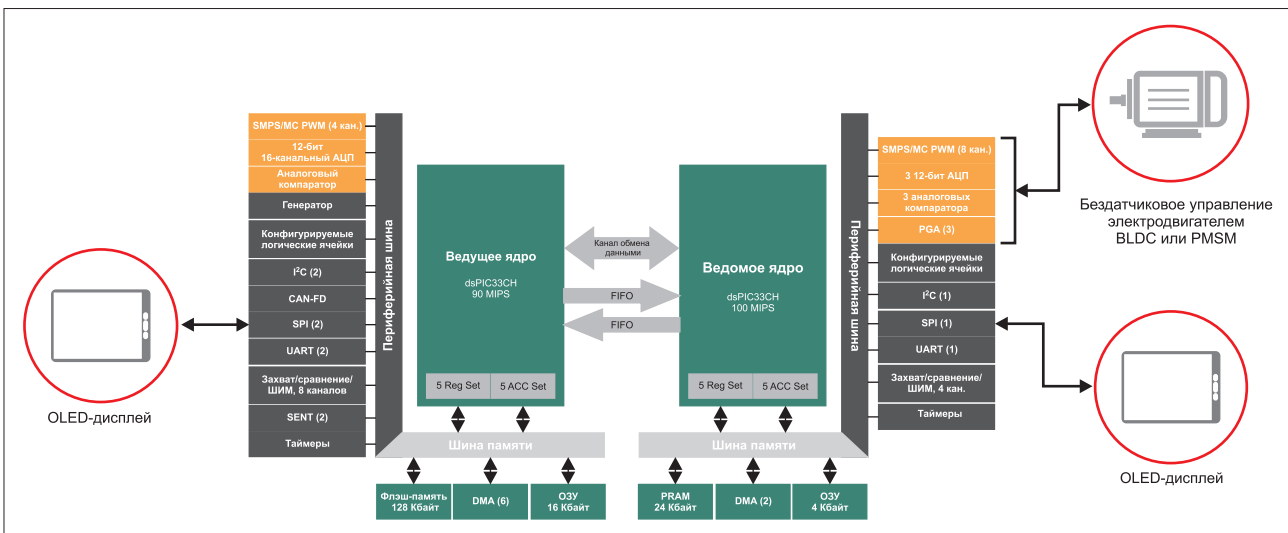


Рис. 2. Пример практического использования dsPIC33CH

можно, узнав, что речь идет о 16-бит МК, многие разработчики презрительно отложат статью – дескать, что за старье им предлагают! Современные инженеры давно перешли на 32-бит МК.

Возможно, будь предлагаемый МК одноядерным, единственным его преимуществом была бы цена, а область его применения была бы незначительна. Однако два ядра существенно меняют расстановку сил и отлично подходят для приложений, где имеются две или более задачи, критичные ко времени выполнения. К тому же, наличие двух ядер позволяет двум группам разработчиков ПО работать практически независимо. Структурная схема МК приведена на рисунке 1. Перечислим некоторые основные параметры dsPIC33CH:

- напряжение питания: 3–3,6 В;
- частота тактирования: 200 МГц;
- производительность ведущего ядра: 90 MIPS;
- производительность ведомого ядра: 100 MIPS;
- разрядность регистра аккумулятора: 40 бит;
- аппаратная поддержка умножения 32-бит чисел;
- количество коммуникационных интерфейсов: 9;
- ШИМ для управления электроприводом;

- независимая от ядра периферия;
- интерфейс квадратурного энкодера.

Как видно из приведенных данных, цифровой контроллер dsPIC33CH представляет собой довольно простое и экономичное устройство для использования во встраиваемых приложениях. Он довольно просто настраивается, а его отладка не вызывает затруднений. Кроме того, поддерживается отладка каждого ядра по отдельности. Заметим также, что использование двух ядер позволяет сэкономить время на переключении контекста между разными задачами – каждое ядро выполняет свою критичную ко времени исполнения задачу. Упрощается также интеграция стеков, базирующихся на разных архитектурах.

Пример практического использования dsPIC33CH показан на рисунке 2. В нем ведомое ядро реализует алгоритм векторного управления бесщеточным электродвигателем постоянно тока, а главное ядро поддерживает графический интерфейс пользователя. Причем, ПО для каждого из ядер разработчики разные команды. Обмен информацией между ядрами поддерживается простым протоколом. ☺