

# Отличия в программировании SDK-1.1 с ADuC842, ADuC831 и ADuC812

/ Версия 1 /

## Изменения в документе

Версия	Дата	Описание	Автор
1	28.04.04	Базовая версия	Лукичев А.Н.

## Оглавление

Предмет обсуждения .....	2
Хранение кода, карты памяти.....	2
Загрузка и старт программ пользователя .....	3
Система прерываний .....	3
Синхронизация и производительность .....	3
Работа с UART .....	3

## Предмет обсуждения

SDK-1.1, начиная с апреля 2004 года, поставляется не только с ADuC812 в качестве вычислителя, но и с другими микроконтроллерами семейства MicroConverter™ фирмы Analog Devices Inc. Микроконтроллеры этого семейства имеют одинаковую систему команд, базовый набор периферии, однако имеют и некоторые отличия. Эти отличия следует учитывать при подготовке программ для SDK-1.1: некоторыми из них можно просто воспользоваться как новыми возможностями, некоторые затрагивают вопросы совместимости стендов с различными вычислителями со старыми программами, написанными для стендов первых версий. Обсуждению этих вопросов и посвящен данный документ.

В следующую таблицу сведены некоторые заслуживающие внимания, с нашей точки зрения, отличия микроконтроллеров ADuC, устанавливаемых в SDK-1.1.

**Таблица 1. Ключевые отличия вычислителей.**

Параметр	ADuC812	ADuC831	ADuC842
FLASH-память команд	8 Кб	62 Кб	62 Кб
Перепрограммирование во время исполнения	Нет	Да	Да
Возможность исполнения команд из ОЗУ SDK-1.1	Да	Нет	Нет
FLASH-память данных	640 байт	4 Кб	4 Кб
ОЗУ	256 байт IDATA	256 байт IDATA + 2048 байт XDATA	256 байт IDATA + 2048 байт XDATA
Частота ядра, МГц	11.0592	11.0592	Настраиваемая (PLL)
Машинный цикл	12 тактов	12 тактов	1 такт
Скорость UART, бод	До 19200	До 115200	До 115200
Синхронизация UART	Timer 1, Timer 2	Timer 1, Timer 2, Timer 3	Timer 1, Timer 2, Timer 3
Доступная в SDK-1.1 дополнительная периферия (по сравнению с ADuC812)	Монитор питания	2 ШИМ, счетчик интервалов времени, монитор питания	2 ШИМ, счетчик интервалов времени

Ниже будут рассмотрены несколько аспектов программирования SDK-1.1, которые затрагивают различия между микроконтроллерами, устанавливаемыми в стендах.

## Хранение кода, карты памяти

В стендах с ADuC812 программы пользователя хранились в ОЗУ и с выключением питания терялись. После очередного включения стенда приходилось вновь доставлять в ОЗУ программу для ее исполнения. В ПЗУ стенда хранилось только системное ПО, обеспечивающее доставку пользовательских программ и передачу им управления (в том числе и при возникновении прерываний). Для хранения программы пользователю было доступно 56 Кб ОЗУ.

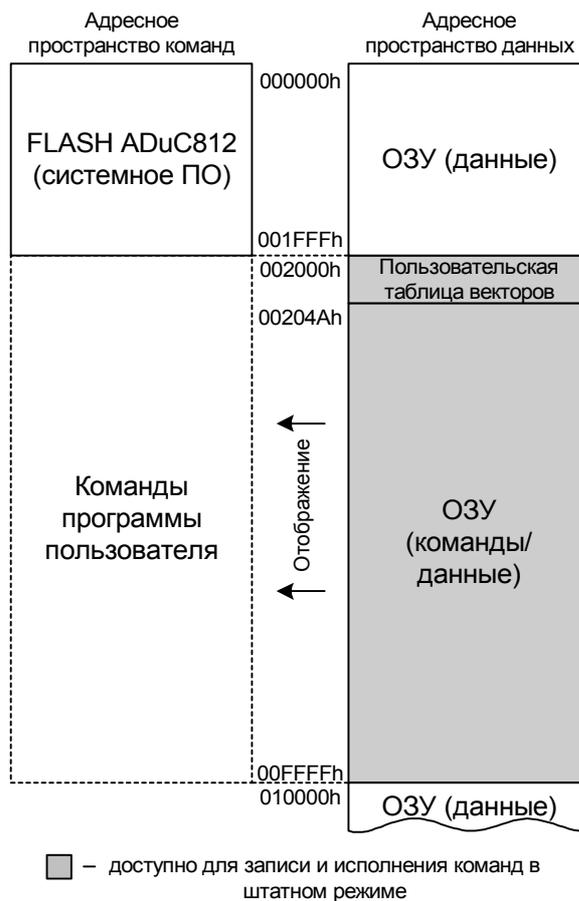


Рисунок 1. Размещение программ в SDK-1.1 с ADuC812.

В штатном режиме работы ADuC812 (т.е. не инструментальном, в котором микроконтроллер обеспечивает программирование своей FLASH-памяти) в SDK-1.1 имелась возможность модифицировать содержимое части (56 из 64 Кб) памяти программ. В стендах с ADuC812 команды по адресам 002000h – 00FFFFh извлекались микроконтроллером из единственной в стенде микросхемы статической памяти, в которой хранились также и данные. В системе команд семейства MicroConverter™ нет отдельных команд модификации памяти программ, но зато есть команды модификации ячеек внешней памяти данных. Таким образом, ячейки из указанного диапазона ОЗУ стенда могли быть модифицированы как данные, и, в то же время, их содержимое могло трактоваться микроконтроллером как программный код. Иными словами, ячейки ОЗУ SDK-1.1 в диапазоне адресов 002000h – 00FFFFh отображались как в пространство внешних данных микроконтроллера, так и в пространство его команд. Эта особенность легла в основу процесса размещения и запуска пользовательского кода в SDK-1.1: системное ПО (загрузчик), расположенное в ПЗУ, принимало пользовательский код в виде HEX-образа через последовательный канал ПК<-> SDK-1.1 (RS232C), записывало его в ОЗУ по нужным адресам, а затем передавало в эти адреса управление.

В стендах SDK-1.1 с микроконтроллерами старших модификаций (ADuC831, ADuC842) объем FLASH-памяти составляет 62 Кб. Возможность исполнения команд из внешних по отношению к микроконтроллеру запоминающих устройств у ADuC84x отсутствует вовсе, а у ADuC83x потенциально есть, но в SDK-1.1 не задействована<sup>1</sup>. В то

<sup>1</sup> В ADuC83x выборка команд из внутреннего или из внешнего устройства памяти задается логическим уровнем на входе #EA. В зависимости от значения напряжения на этом входе, микроконтроллер выбирает команды *либо* из внутреннего блока FLASH-памяти, *либо* из внешнего запоминающего устройства. В SDK-1.1 на вход #EA подается лог. «1», что соответствует выборке команд из внутреннего блока FLASH-памяти.

же время, у этих микроконтроллеров возможно перепрограммирование части внутреннего блока FLASH-памяти во время исполнения команд из другой его части.



**Рисунок 2. Размещение программ в SDK-1.1 с ADuC83x, ADuC84x.**

В ADuC83x, ADuC84x весь блок FLASH-памяти программ (диапазон адресов 0000h – F7FFh) доступен для стирания/перезаписи в инструментальном режиме работы микроконтроллера, однако в штатном режиме для стирания/перезаписи доступны только ячейки по адресам 0000h – DFFFh (56 Кб). Для этого вся память в указанном диапазоне разбита на сектора по 64 байта каждый, которые можно стирать независимо друг от друга. Программировать можно как отдельную ячейку (байт), так и страницу из 256 байт FLASH-памяти команд. При этом команды на стирание/программирование могут располагаться где угодно, т.к. на время выполнения операции с FLASH ядро микроконтроллера приостанавливается (не выбирает команды и не отвечает на запросы на прерывание). Таким образом, возможность модификации содержимого памяти программ в SDK-1.1 с ADuC831 и ADuC842 сохранена, но память программ стала энергонезависимой.

На рисунке 2 также присутствует заштрихованная область, которая соответствует диапазону адресов F800h – FFFFh (2 Кб). В штатном режиме работы микроконтроллера в этот диапазон адресов отображается последовательность из инструкций «пор» (код 00h).

## Загрузка и старт программ пользователя

В стендах с ADuC812 пользовательские программы загружались в ОЗУ. Поэтому при каждом старте системное ПО вначале ожидало доставки образа через UART с инструментального ПК, сигнализируя о готовности к приему посылкой символов ‘.’. После приема образа с программой вся периферия ADuC812 приводилась в исходное состояние (как по сигналу RESET), в том числе и UART, и после этого управление передавалось принятой программе.

Кроме того, в стендах (исключая самые первые версии) с ADuC812 имелась возможность запуска демонстрационной программы, которая вместе с системным ПО располагалась во FLASH-памяти ADuC812. При старте стенда анализировалось состояние переключателя EP7 на плате и, если он находился в состоянии «ON», вместо приема образа с ПК запускалась демонстрационная программа, которая предлагала пользователю ознакомиться с возможностями стенда через диалог с помощью ЖКИ и клавиатуры. Если переключатель EP7 находился в состоянии «OFF», то ожидался прием программы с ПК.

Состояние процесса приема образа отображалось на ЖКИ стенда. На момент передачи управления пользовательской программе ЖКИ был инициализирован, и повторной инициализации его не требовалось.

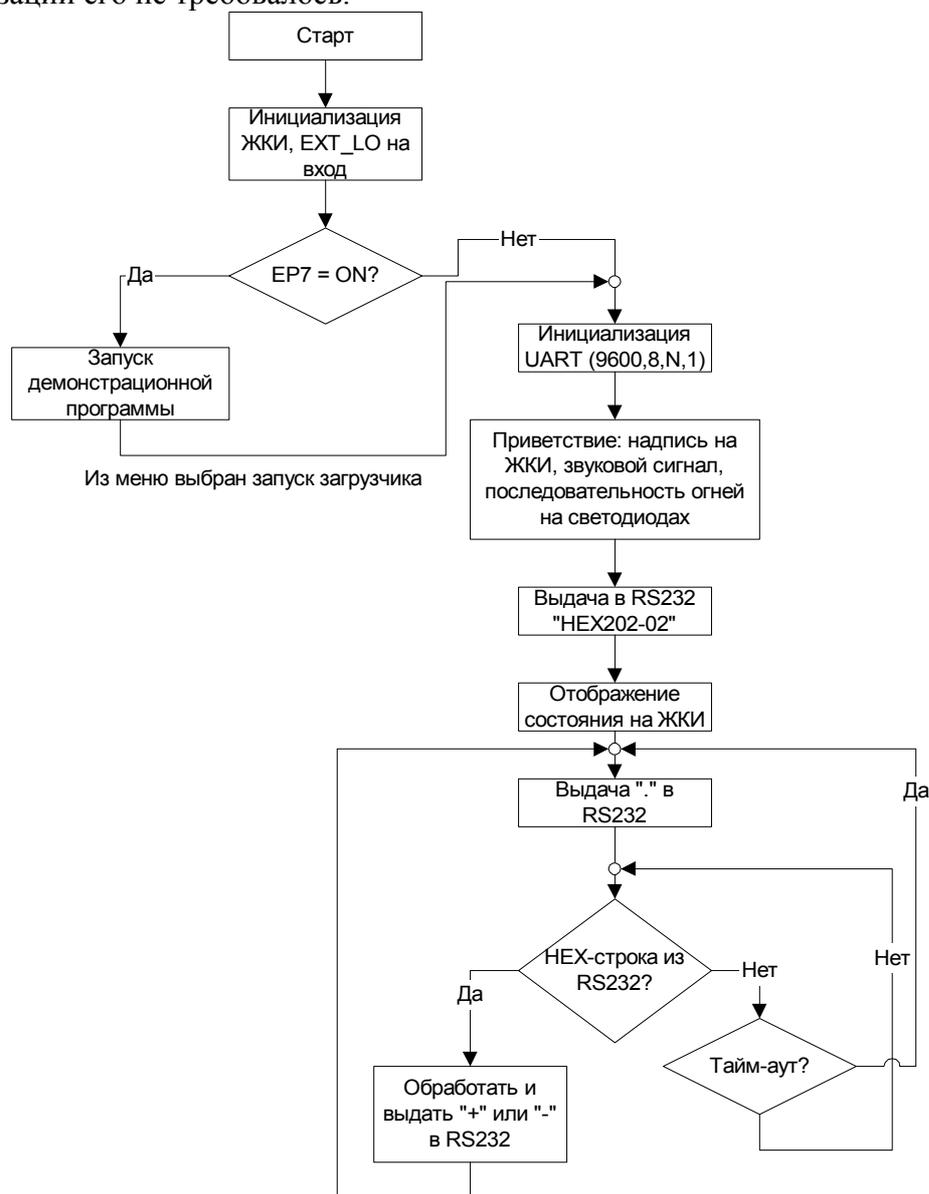


Рисунок 3. Действия при старте и загрузке программ в SDK-1.1 с ADuC812.

В SDK-1.1 с ADuC831 или ADuC842 программы хранятся во FLASH-памяти микроконтроллера, поэтому нет нужды при каждом старте стенда ожидать от ПК появления образа новой пользовательской программы. Может быть запущена программа, помещенная во FLASH некоторое время ранее. Однако, может потребоваться доставить с ПК новый образ с программой пользователя. Проверка на наличие такой ситуации происходит сразу после старта стенда, т.к. у системного ПО нет гарантированной возможности проверить состояние канала связи с ПК после передачи управления программе пользователя, т.к. UART может использоваться ею. В стендах с ADuC83x и

ADuC84x протокол взаимодействия с ПК по доставке пользовательских программ остался таким же, как и у стандов с ADuC812, поэтому для связи с ПК резидентный загрузчик SDK-1.1 сначала должен продемонстрировать готовность к приему HEX-команд выдачей символа '.' в канал связи. При старте SDK-1.1 в канал выдается последовательность из 10 символов '.', после каждого из которых производится ожидание реакции со стороны ПК (приход HEX-команды), тайм-аут на которое составляет примерно 20 мс. Таким образом, суммарное время установления связи с ПК составляет примерно 200 мс, после чего управление передается пользовательской программе, которая была записана в память ранее. Кроме последовательности из 10 символов '.', в канал связи с ПК также выдается надпись «HEX202-03», а также, возможно, другие (в отладочных целях), однако все они составляют не более 20 символов и выдаются до 10 символов '.'.

В SDK-1.1 с ADuC812 вместе с системным ПО во FLASH-памяти хранилась также демонстрационная программа, которая системных функций не выполняла, однако должна была поставляться вместе со стандом. Оформление этой программы отдельно от системного ПО привело бы к тому, что ее нужно было бы загружать в ОЗУ каждый раз для демонстрации возможностей станда. Поэтому она хранилась в энергонезависимой памяти вместе с системным ПО. Поскольку в SDK-1.1 с ADuC83x и ADuC84x программы пользователя хранятся в энергонезависимой памяти, было решено оформить демонстрационную программу как «пользовательскую» и разместить ее отдельно от системного ПО. При поставке стандов демонстрационная программа содержится в памяти, но, как и любая другая программа пользователя, может быть стерта при первой же загрузке с ПК другой программы.

Запуск программ пользователя в стандах с ADuC83x и ADuC84x осуществляется с адреса, запомненного в момент загрузки. Для хранения этого адреса (точки входа), а также другой информации **в стандах с ADuC83x и ADuC84x зарезервированы две страницы FLASH-памяти данных** по адресам 00h и 01h.

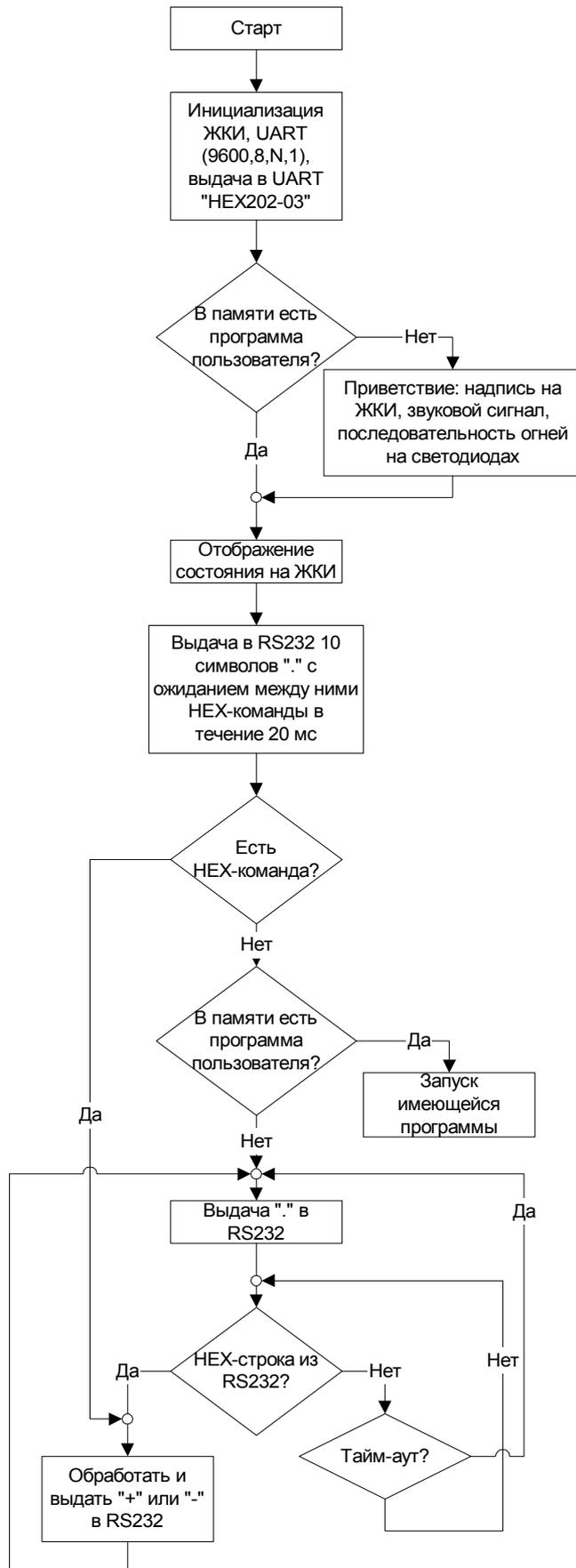


Рисунок 4. Действия при старте SDK-1.1 с ADuC83x и ADuC84x по загрузке и запуску программ пользователя.

## Система прерываний

Микроконтроллер семейства MicroConverter™ при возникновении прерывания передает управление на один из адресов в начале памяти программ, а именно в диапазоне 0003h – 004Ah (для ADuC812, ADuC83x, ADuC84x). В ADuC812 эти адреса соответствуют ячейкам FLASH-памяти, перепрограммирование которых во время исполнения программы пользователя невозможно. Для предоставления пользовательским программам возможности установки собственных векторов прерываний в SDK-1.1 была принята следующая процедура. По возникновении прерывания процессор передавал управление на один из адресов, соответствующий источнику запроса на прерывание, например x. По этому адресу находилась команда передачи управления на адрес (2000h + x), соответствующий одной из ячеек ОЗУ стенда. Пользователь, записав по этому адресу свой код (скажем, команду передачи управления собственному обработчику прерываний), имел возможность обрабатывать прерывания от данного источника. Таким образом, по адресам 2003h – 204Ah в SDK-1.1 могла располагаться «пользовательская» таблица векторов. Таким образом, для того, чтобы установить собственный обработчик прерываний, пользователю достаточно было записать по нужному адресу в ОЗУ команду передачи управления на его начало. Более подробно, см. «Руководство пользователя SDK-1.1» версий 1.0.8 и старше.

Для сохранения совместимости программ для стендов с ADuC812 с более поздними стендами (с ADuC83x и ADuC84x) в последних предусмотрена поддержка старой процедуры установки пользовательских векторов с помощью записи по адресу (2000h + x) команды передачи управления на код обработки прерываний. При этом предполагалось, что по адресу (2000h + x) пользователь записывает команду безусловной дальнейшей передачи управления «ljmp address», где address – адрес обработчика прерываний пользователя. Именно такой способ обработки прерываний (дальняя передача управления на свой обработчик) предполагался наиболее вероятным в SDK-1.1 с ADuC812. Для сохранения совместимости в SDK-1.1 с ADuC83x и ADuC84x при возникновении прерываний управление передается на код, который извлекает 16-ти разрядное число по адресу (2000h + x + 1), интерпретирует его как адрес обработчика и передает по нему управление. Загрузив в стенд с ADuC83x (или ADuC84x) программу, скомпилированную для стендов с ADuC812 и использующую прерывания по старой схеме (через ОЗУ), можно убедиться, что в части обработки прерываний эта программа будет работать. Код, который обеспечивает передачу управления в новых стендах через ОЗУ, вносит некоторую задержку в процесс обработки прерываний. Величина этой задержки зависит от тактовой частоты ядра конкретного микроконтроллера и представлена для различных вариантов в таблице.

**Таблица 2. Задержка, вносимая системным кодом в обработку прерываний.**

Микроконтроллер	Число машинных циклов	Время, мкс
ADuC812 @ 11.0592 МГц	4	4,34
ADuC831 @ 11.0592 МГц	43	46,658
ADuC841 @ 11.0592 МГц	46	4,159
ADuC832 @ 0.131072 МГц	43	3 936
ADuC832 @ 0.262144 МГц	43	1 968
ADuC832 @ 0.524288 МГц	43	984
ADuC832 @ 1.048576 МГц	43	492
ADuC832 @ 2.097152 МГц	43	246
ADuC832 @ 4.194304 МГц	43	123
ADuC832 @ 8.388608 МГц	43	61,512
ADuC832 @ 16.777216 МГц	43	30,756
ADuC842 @ 0.131072 МГц	46	350
ADuC842 @ 0.262144 МГц	46	175

ADuC842 @ 0.524288 МГц	46	87,738
ADuC842 @ 1.048576 МГц	46	43,869
ADuC842 @ 2.097152 МГц	46	21,935
ADuC842 @ 4.194304 МГц	46	10,967
ADuC842 @ 8.388608 МГц	46	5,484
ADuC842 @ 16.777216 МГц	46	2,742

В SDK-1.1 с ADuC83x (или ADuC84x) код, обеспечивающий совместимость со старым механизмом установки векторов прерываний, располагается в диапазоне адресов 0000h – 007Fh. Эти адреса соответствуют области FLASH-памяти, отведенной для хранения пользовательского кода. При загрузке программ, располагающихся в адресах, которые не пересекаются с диапазоном 0000h – 007Fh, ячейки в этом диапазоне адресов не стираются. В случае, если не предполагается использовать описанную выше схему установки векторов, в этом диапазоне можно разместить собственный код/данные. При этом, очевидно, имевшийся там на момент поставки стенда код будет утерян. Если в дальнейшем потребуется его восстановить, то для этого достаточно загрузить в стенд образ демонстрационной программы, входящей в комплект поставки.

## Синхронизация и производительность

В SDK-1.1 в качестве вычислителя могут использоваться разные микроконтроллеры семейства MicroConverter™, отличающиеся, в том числе, и блоком синхронизации, а также длительностью машинного цикла (см. таблицу 1).

Микроконтроллеры ADuC831/832, как и ADuC812, имеют длительность машинного цикла 12 тактов синхронизации, однако различаются между собой тактовой частотой: у ADuC831 она равна 11.0592 МГц (в SDK-1.1), а у ADuC832 настраивается с помощью встроенного блока PLL как частное от деления 16.777216 МГц на число из набора [1, 2, 4, 8, 16, 32, 64, 128].

Микроконтроллера ADuC841/842/843 имеют длительность машинного цикла в 1 такт синхронизации. Это, безусловно, повысило производительность микроконтроллеров, по сравнению с ADuC812/83x. Однако, из-за кардинального снижения длительности машинного цикла в них пришлось несколько увеличить число циклов, необходимых для исполнения части инструкций (это, в основном, инструкции передачи и возврата управления). Данное обстоятельство может повлиять на совместимость программ, учитывающих длительность исполнения инструкций и скомпилированных для ADuC812/83x. Частота синхронизации ADuC841 (в SDK-1.1) равна 11.0592 МГц. У ADuC842/843 тактовая частота настраивается с помощью блока PLL, аналогично ADuC832.

Частота синхронизации микроконтроллера влияет на весь набор его периферии, поэтому данное обстоятельство следует учитывать при работе с разными стендами и написании совместимых программ. В случае с ADuC84x на работу части периферии также влияет длительность машинных циклов: она начинает тактироваться в 12 раз чаще, по сравнению с ADuC812. Это касается, в первую очередь, таймеров 0 и 1, UART (если он тактируется от таймера 1).

Различия в производительности микроконтроллеров становятся особенно заметными при использовании задержек, основанных на времени исполнения инструкций или блоках таймера 0 или 1. Например, задержка, основанная на таймере 0, в стенде с ADuC842 на 16.777216 МГц будет в 18,2 раза короче, чем в стенде с ADuC812.

## Работа с UART

В ADuC812 UART тактировался с помощью таймера 1 или таймера 2. При частоте синхронизации ядра 11.0592 МГц максимальная скорость передачи через UART равнялась 19200 бод. В ADuC83x/84x имеется возможность синхронизации UART как от таймеров 1 и 2, так и от специального таймера 3, предназначенного только для тактирования UART. Максимальная стандартная скорость, развиваемая UART при использовании таймера 3 равняется 230400 бод. Однако, в SDK-1.1 с ADuC83x/84x порт RS232 может быть рассчитан только на максимальную скорость 115200 бод, что вполне достаточно для быстрого обмена данными через RS232.

При тактировании UART от таймера 1 следует учитывать различия в скорости его работы в разных микроконтроллерах, устанавливаемых в SDK-1.1. См. раздел «Синхронизация и производительность».