

Динамическое программирование аналоговых схем Anadigm управляющим методом

Александр ЩЕРБА
shcherba@prosoft.ru

В статье описаны этапы создания конфигурационных данных для динамического программирования аналоговых интегральных схем (ПАИС) Anadigm управляющим методом. Метод основан на использовании предварительно откомпилированных конфигураций с заранее известными параметрами.

Программируемые аналоговые микросхемы Anadigm

Компания Anadigm — безусловный лидер в области разработки и производства программируемых аналоговых микросхем. Линейка продукции, которую выпускает компания Anadigm, состоит из динамически и статически программируемых аналоговых микросхем. В отличие от статически программируемых схем, где конфигурационные данные загружаются при включении ПАИС и остаются неизменными во время ее работы, динамически конфигурируемая схема позволяет изменять полностью или частично функциональную структуру в реальном

времени в работающем устройстве. Для этого в ПАИС предусмотрена так называемая тень памяти, в которую во время работы аналоговой структуры можно с помощью микроконтроллера загрузить обновленную конфигурацию, а затем по команде мгновенно активизировать новую аналоговую конфигурацию.

При использовании бюджетных недорогих микроконтроллеров для динамического конфигурирования программируемых аналоговых микросхем Anadigm оправдано использование управляющего метода (State-Driven), который в отличие от алгоритмического метода использует предварительно сконфигурированные данные.

Создание основного конфигурационного кода

Рассмотрим пример динамического программирования частоты среза фильтра низкой частоты, выполненного на базе ПАИС Anadigm 2-го поколения AN221E04. Для динамического программирования будем использовать управляющий метод. Хранение/передача конфигураций и управление ПАИС будут осуществляться с помощью микроконтроллера по последовательной шине SPI (рис. 1).

Работа цифровой части ПАИС

В первый момент времени после подачи питания выводы ACLK, CFGFLGb и OUTCLK/SPIMEM находятся в неопределенном состоянии, пока не сгенерируется внутренний импульс сброса и не переведет их в исходное состояние. Типовое время — около 30 мс.

Затем, одновременно с передачей конфигурационных данных по последовательной шине DIN, на выводе ACTIVE установится низкий логический уровень, который сохраняется до тех пор, пока не завершится процесс передачи данных (конфигурирования). После завершения этого процесса вывод ACTIVE переходит в третье состояние, и на нем устанавливается сигнал высокого логического уровня благодаря наличию подтягивающего резистора R1. Внешний сигнал низкого логического уровня на выводе ACTIVE также может быть использован для блокировки активации переданных конфигурационных данных.

Вход CS2b используется как сигнал выбора кристалла с активным низким логическим уровнем, при этом он должен быть синхронизирован с синхросигналом DCLK. Выводы CS2b и CS1b соединены по схеме ИЛИ-НЕ, выход которой используется для запрета или разрешения выдачи сигнала синхронизации на конфигурационную логику. Когда оба сигнала CS имеют низкий уровень, выдача

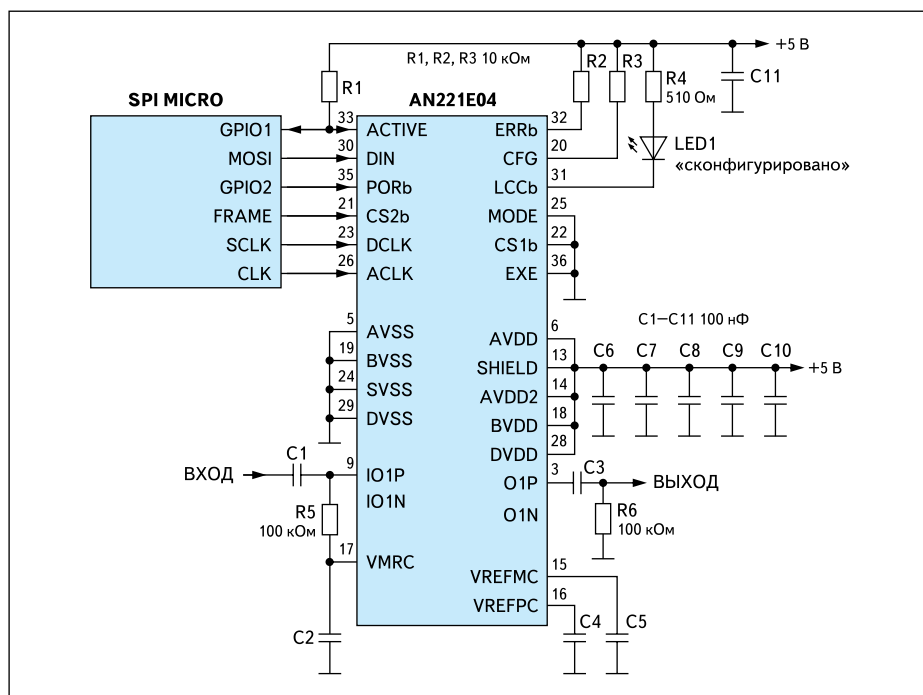


Рис. 1. Электрическая принципиальная схема подключения ПАИС для динамического программирования с помощью микроконтроллера

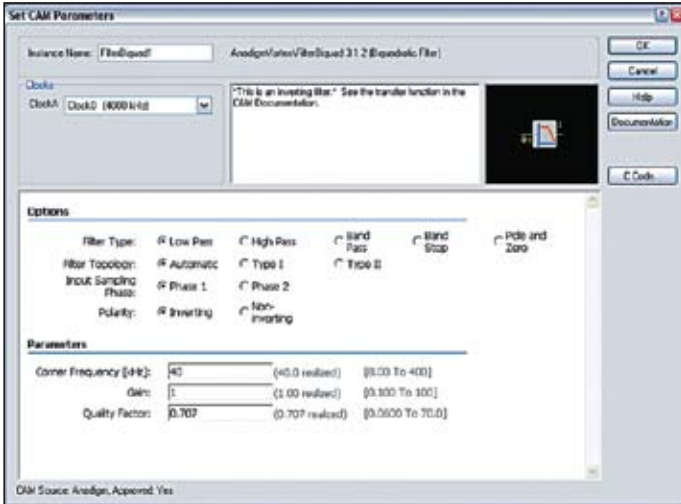


Рис. 2. Окно настройки фильтра

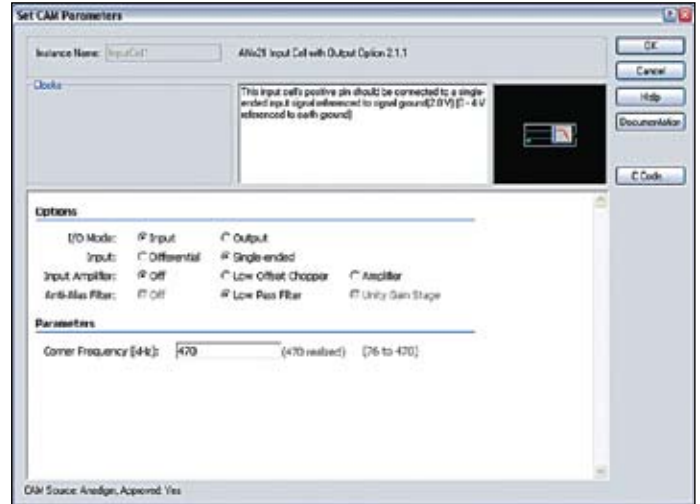


Рис. 4. Окно настройки ячейки ввода/вывода

синхросигнала разрешена. Поэтому необходимым условием для передачи конфигурационных данных в ПАИС является наличие сигналов низкого уровня на входах CS1b и CS2b. При подаче на вход CS2b сигнала высокого уровня ПАИС переходит в неактивное состояние, с пониженным энергопотреблением.

Как правило, вывод LCCb используется для индикации завершения процесса конфигурирования, а подсоединенный по схеме светодиод LED1 будет свидетельствовать о завершении процесса конфигурирования без ошибок. При конфигурировании ПАИС на чтение [1] с выхода LCCb можно считывать содержимое конфигурационного ОЗУ. Считываемые данные появляются на LCCb через 16 тактов синхросигнала.

Вывод ERRb можно использовать в режиме входа или выхода с открытым стоком. В обоих случаях к нему необходимо подключить подтягивающий резистор на +5 В (типичное значение — 10 кОм). Во время инициализации ПАИС он используется в режиме выхода, и до завершения этого процесса на нем присутствует низкий логический уровень. После завершения конфигурирования вывод переходит в третье состояние, и на нем устанавливается сигнал высокого логического уровня благодаря наличию подтягивающего резистора R2.

Вывод CFGFLGb используется при программировании системы (цепочки), состоящей из нескольких параллельно включенных микросхем. Наличие на нем логического нуля свидетельствует о течении процесса конфигурирования. В этом случае принимающая конфигурационные данные ПАИС установит на общей для всех микросхем шине CFGFLGb низкий логический уровень, при котором все остальные микросхемы в цепочке, для которых не предназначены данные, будут их игнорировать.

Перезагрузка ПАИС может быть осуществлена подачей импульса низкого логического уровня на входе PORb, при загрузке первичной конфигурации программно или при

возникновении ошибки — логическим нулем на входе ERRb. Если в процессе работы на вход PORb подать импульс низкого уровня, ПАИС будет принудительно перезагружена в режиме включения питания. Если на входе PORb присутствует ноль длительное время, ПАИС будет удержана в состоянии сброса.

Работа аналоговой части ПАИС

Обработка сигнала внутри ПАИС осуществляется с использованием однополярных дифференциальных сигналов, с постоянной составляющей +2 В. Для работы с недифференциальным униполярным аналоговым сигналом необходимо использовать входную ячейку, работающую в режиме “Single-ended” [2]. Входной сигнал должен поступать на позитивный дифференциальный вход ячейки ввода/вывода IO1P и иметь постоянную составляющую +2 В. Необходимое смещение можно подавать от внутреннего источника опорного напряжения VMRC (вывод 17) через постоянный резистор R5, предотвратив попадание постоянной составляющей на вход источника сигнала конденсатором C1.

Постоянную составляющую в выходном сигнале удаляют с помощью конденсатора C3. При программировании ПАИС необходимо учитывать тот факт, что выходной сигнал будет равен половине амплитуды, из-за того, что используется один (O1P) из двух (O1P и O1N) дифференциальных выходов.

Проектирование электрической схемы

Для проектирования аналоговой структуры ПАИС понадобится САПР AnadigmDesigner2, который распространяется бесплатно и является полнофункциональным симулятором, позволяющим отладить электрическую схему без программирования микросхемы ПАИС [3]. В состав программы включены разнообразные библиотеки [4]. На первоначальном этапе проектирования необходимо

выбрать тип используемой микросхемы — ПАИС 2-го поколения AN221E04, установить в окне свойств **Chip Setting** частоту синхронизирующего сигнала **Master Clock**, равную частоте тактового сигнала на шине ACLK (рекомендуемая частота 16 МГц), а затем приступить к построению аналоговой структуры.

Встроенная в САПР AnadigmDesigner2 библиотека конфигурируемых аналоговых модулей (KAM) может быть вызвана с помощью пункта меню **Edit** → **Insert New CAM** или при нажатии горячей клавиши **m**. Справа в прокручиваемом списке открывшегося окна выбора KAM необходимо добавить модуль биквадратичного фильтра (FilterBiquad) и настроить его параметры (рис. 2):

- тип фильтра — фильтр нижней частоты (Low Pass);
- частоту среза **Corner Frequency** = 40 кГц;
- коэффициент передачи фильтра **Gain** = 1;
- добротность **Quality Factor** = 0,707 (фильтр Баттерворта).

Соединяем фильтр с ячейками ввода/вывода так, как показано на рис. 3, а затем на-

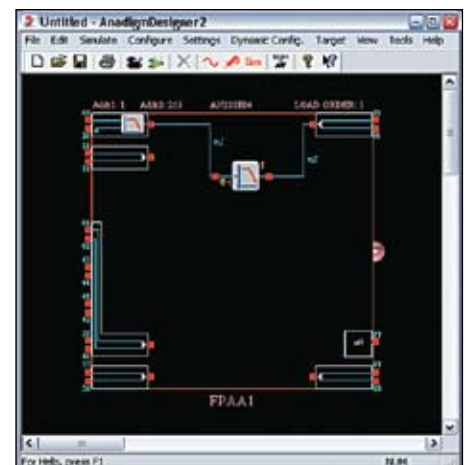


Рис. 3. Электрическая принципиальная схема в окне программы AnadigmDesigner2

страиваем параметры входной ячейки для работы с недифференциальным входным сигналом. Для этого в свойствах ячейки устанавливаем следующие параметры (рис. 4):

- Режим ячейки ввода/вывода (I/O Mode) — вход (Input).
- Тип входа (Input) — недифференциальный (Single-ended).
- Входной усилитель (Input Amplifier) — выключен (Off).
- Фильтр (Anti-Alias Filter) — фильтр низкой частоты (Low Pass Filter).
- Частоту среза фильтра (Corner Frequency), равную 470 кГц (по умолчанию), оставляем без изменения.

Получение основных конфигурационных данных

После завершения разработки схемы необходимо определить метод динамического конфигурирования, а также какие из ее частей могут нуждаться в динамическом пере-конфигурировании. Для выбора динамического программирования управляющим

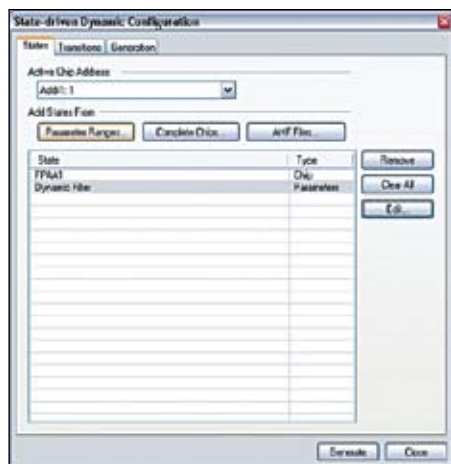


Рис. 5. Окно State-Driven Dynamic Configuration

методом необходимо в меню программы AnadigmDesigner2 выбрать **Dynamic Config** (динамическое конфигурирование), а затем в открывшемся подменю выбрать **State-Driven Method...** (управляющий метод). Открывшееся окно **State-Driven Dynamic Configuration** (рис. 5) предназначено для добавления различных схем и параметров, конфигурационные данные которых необходимо откомпилировать [5].

С помощью элемента управления **Complete Chips...** в закладке **States** выбираем основную конфигурацию FPA1, созданную на предыдущем этапе, а затем с помощью элемента управления **Parameter Ranges...** параметры, которые необходимо изменять при динамическом конфигурировании, их диапазон изменения и количество ключевых состояний (рис. 6). В средней закладке **Transitions** отмечаем основное (начальное) состояние ПАИС галочкой (рис. 7) и переходим к закладке **Generation** для генерации конфигурационного кода (рис. 8). В диалоговом окне **Generation** возможен выбор формата выходных данных **Output Type** и места их сохранения **Destination Directory**.

Сохраненный файл в формате **Generate C Formatted Configuration Text Files** имеет текстовый формат и содержит основные и реконфигурационные данные со следующей структурой (рис. 9):

- Основные конфигурационные данные (Primary Configurations) — 134 байта, будут содержать режим работы ПАИС, логические адреса.
- Реконфигурационные данные (Transition Configurations):
 - Dynamic Filter: Fo [40] (37 байтов) — реконфигурационные данные динамического изменения конфигурационной логики для частоты среза фильтра низкой частоты 40 кГц;
 - Dynamic Filter: Fo [50] (37 байтов) — реконфигурационные данные динамиче-

ского изменения конфигурационной логики для частоты среза фильтра низкой частоты 50 кГц;

- FPA1 (37 байтов) — возврат ПАИС в первоначальное состояние на момент загрузки конфигурации.

Полученные конфигурационные данные можно включать с С-код микроконтроллера, используя типовые алгоритмы [5].

Алгоритмы загрузки основной конфигурации и обновления

После подачи напряжения питания ПАИС запустит процесс инициализации и сформирует сигнал внутреннего сброса, после чего основная и «теневая» память будут очищены. Процесс инициализации ПАИС завершается появлением на выводе ACTIVE «лог. 1», который будет означать готовность ПАИС к приему основных конфигурационных данных **Primary Configurations** (рис. 9). Затем микроконтроллер может подать сигнал выбора кристалла CS2b на первую микросхему в цепочке FPA1 для инициализации приема данных на вывод DIN с помощью алгоритма, представленного в листингах 1–3.

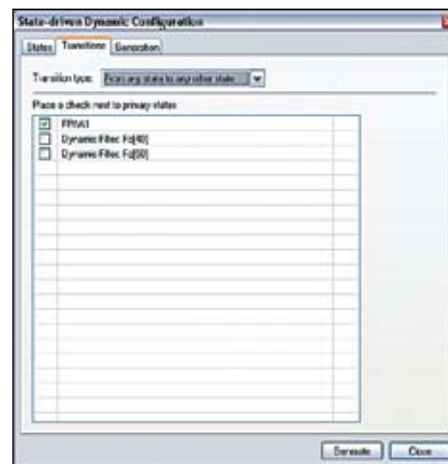


Рис. 7. Переход к закладке Generation

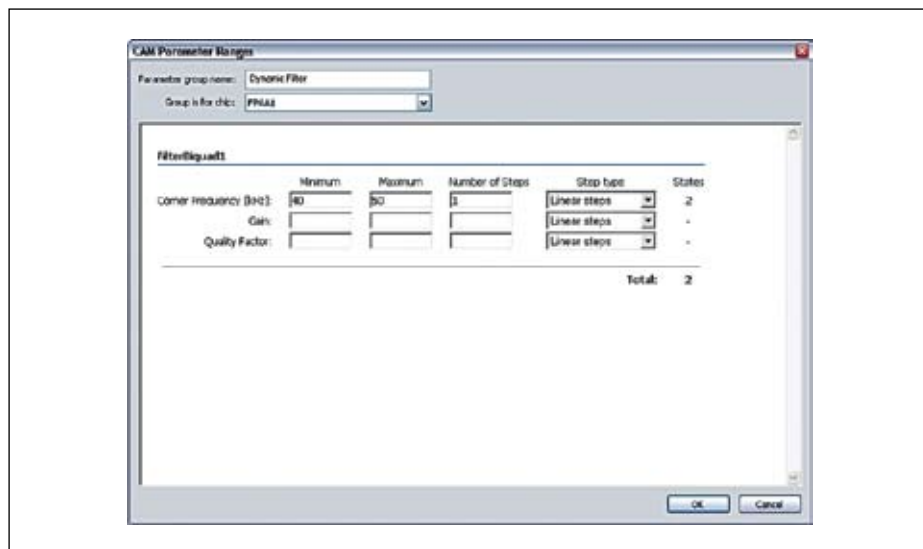


Рис. 6. Окно настройки диапазона изменения параметров конфигурационной логики

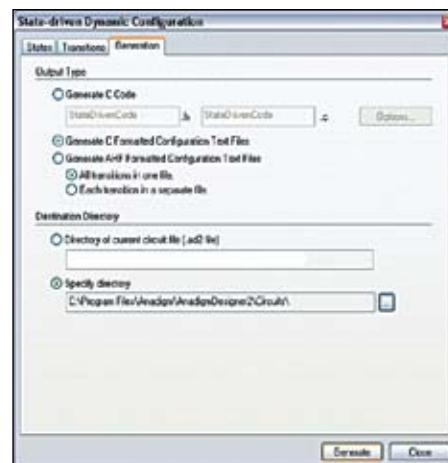


Рис. 8. Окно выбора формата конфигурационных данных и места их сохранения

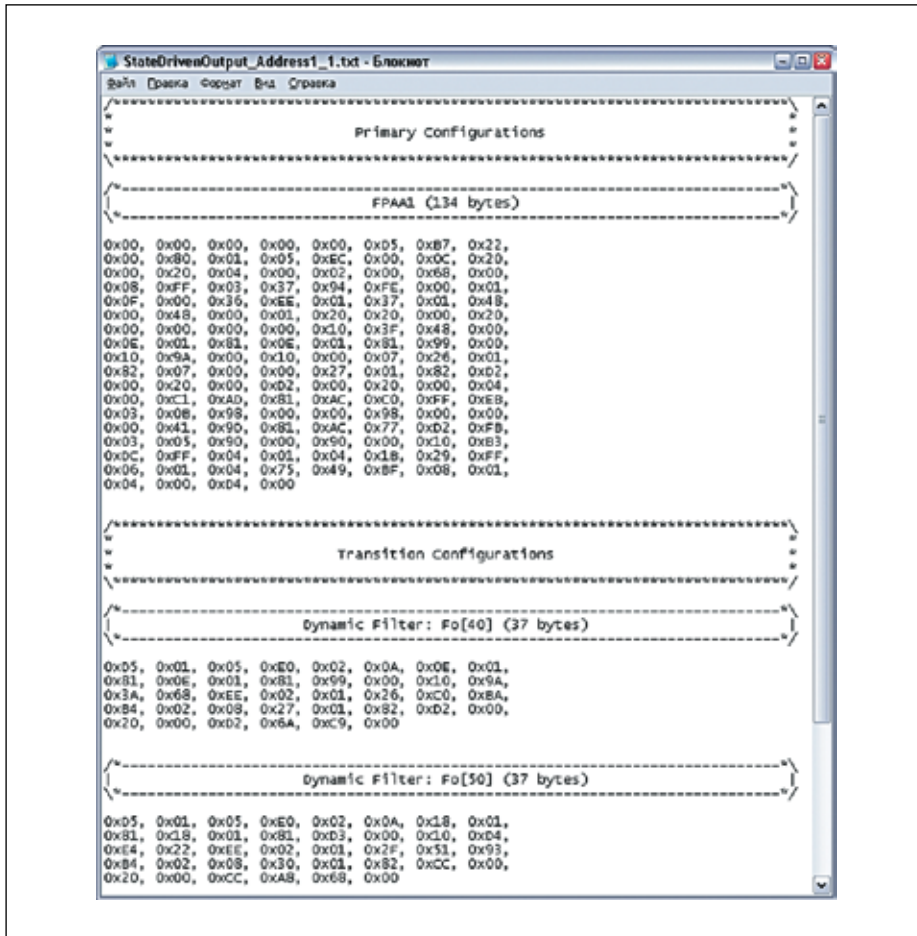


Рис. 9. Файл в формате Generate C Formatted Configuration Text Files

```
#define Primary_Config_Size 134
const an_Byte Primary_Data[] =
{
    0x00, 0x00, 0x00, 0x00, 0x00, 0xD5, 0xB7, 0x22,
    ...
    0x04, 0x00, 0xD4, 0x00
};
```

Листинг 1. Основные конфигурационные данные

```
# Биты передаются в ПАИС по шине DIN, запись логического
состояния "лог.1" в "лог.0".
void Data_Write(int8 data)
{
    an_Byte bit = 8;
    while (bit--)
    {
        if (data & 0x80)
            DIN_High;
        DCLK_High;
        DCLK_Low;
        DIN_Low;
        data <<= 1;
    }
}
```

Листинг 2. Запись одного байта данных в ПАИС с помощью вызова функции

```
void Load_Primary_Circuit(void)
{
    an_Byte i;
    for (i = 0; i < Primary_Config_Size; i++)
        Data_Write(Primary_Data[i]);
}
```

Листинг 3. Запись конфигурационных данных с помощью вызова функции

После загрузки конфигурационных данных светодиод LED1, подключенный к выводу LCCb, загорится, а на шине ACTIVE будет установлен уровень «лог. 1», означающий, что микросхема сконфигурирована и готова для приема обновленных конфигурационных данных.

Во время работы и выполнения основного конфигурационного кода ПАИС будет искать на шине DIN байт синхронизации (0xD5) и следующий за ним байт логического адреса ADDR1, полученного на этапе загрузки основного конфигурационного кода [1]. При совпадении адреса передаваемых реконфигурационных данных с адресом, полученным при загрузке основной конфигурации, ПАИС установит на шине ACTIVE уровень «лог. 0». После чего микроконтроллер может отправить ПАИС реконфигурационные данные (рис. 9) Dynamic Filter: Fo [40] (частота среза фильтра нижней частоты 40 кГц) или Dynamic Filter: Fo [50] (частота среза фильтра нижней частоты 50 кГц) по шине DIN, используя алгоритм, представленный в листингах 4, 5.

После того как передача реконфигурационных данных будет завершена, на шине ACTIVE будет установлен уровень «лог. 1». Значит, загруженная аналоговая структура активирована, и ПАИС готова к следующему приему реконфигурационных данных.

```
#define Reconfig_Size 37
const an_Reconfig Data [] =
{
    0xD5, 0x01, 0x05, 0xE0, 0x02, 0x0A, 0x0E, 0x01,
    ...
    0x20, 0x00, 0xD2, 0x6A, 0xC9, 0x00
};
```

Листинг 4. Реконфигурационные данные

```
void Load_Reconfig_Circuit (void)
{
    an_Byte i;
    for (i = 0; i < Reconfig_Size; i++)
        Data_Write(Reconfig_Data[i]);
}
```

Листинг 5. Запись реконфигурационных данных с помощью вызова функции

Заключение

Программируемые аналоговые микросхемы могут стать незаменимым инструментом в приложениях, для которых необходимо низкое время отклика при обработке аналоговых сигналов с высокой точностью: это PID-регуляторы, аналоговые фильтры высокого порядка, нейронные сети и т. п. Такие микросхемы используются и при обработке сигналов от датчиков. Использование ПАИС Anadigm позволяет отказаться от значительной части внешних навесных элементов и тем самым уменьшить габариты устройства и повысить его надежность.

Представленный в статье способ динамического конфигурирования программируемых аналоговых микросхем управляющим методом прост в реализации, не требует расширенных знаний в программировании, обладает возможностью гибкого программирования аналоговой схемы и переконфигурирования отдельных параметров или функций в работающем устройстве в режиме реального времени.

Литература

1. Щерба А. Конфигурационный протокол динамически программируемых аналоговых схем Anadigm // Компоненты и технологии. 2009. № 12.
2. Щерба А. Построение входных и выходных цепей программируемых аналоговых схем Anadigm // Компоненты и технологии. 2008. № 12.
3. Полищук А. Система автоматизированного проектирования программируемых аналоговых интегральных схем AnadigmDesigner2. Часть 1. Первый шаг: знакомство с интерфейсом // Компоненты и технологии. 2005. № 6–7.
4. Щерба А. Программируемые аналоговые ИС Anadigm: применение конфигурируемых аналоговых модулей в составе программы AnadigmDesigner2 // Компоненты и технологии. 2007. № 12.
5. Application Note: State-Driven Control of a dpASP using a Microchip PIC. App Note 206. Anadigm, 2008.