

Конфигурационный протокол динамически программируемых аналоговых схем Anadigm

Александр ЩЕРБА
shcherba@prosoft.ru

В статье описывается конфигурационный протокол для программируемых аналоговых схем Anadigm 2-го поколения. Приводится схема подключения для динамического конфигурирования ПАИС и пример загрузки реконфигурационных данных.

Программируемые аналоговые микросхемы Anadigm

Компания Anadigm, выпускающая программируемые аналоговые микросхемы, — безусловный лидер в этой области. Эта компания занимается производством как динамически программируемых аналоговых микросхем, так и статически программируемых. Статически программируемые микросхемы работают с неизменной конфигурацией, полученной при включении ПАИС. В отличие от них, динамически программируемые схемы позволяют изменять полностью или частично функциональную структуру в реальном времени в работающем устройстве. Для предотвращения потери данных в процессе изменения аналоговой структуры при динамическом конфигурировании необходимо «мгновенно» менять конфигурационные данные. Для этого в ПАИС предусмотрена так называемая «теневая» память, в которую во время работы аналоговой структуры можно загрузить обновленную конфигурацию (реконфигурировать), а затем по команде активизировать новую аналоговую структуру за один такт синхросигнала.

Первоначальные (основные) конфигурационные данные, содержащие аналоговую структуру, создаются в программе AnadigmDesigner2 и загружаются в ПАИС с помощью внешней микросхемы памяти или микроконтроллера [1]. Обновленные конфигурационные данные при динамическом программировании создаются как набор конфигураций с заранее известными параметрами и загружаются в ПАИС с помощью управляющего микроконтроллера (State-Driven Method, управляющий метод). Данные также могут быть рассчитаны в самом микроконтроллере по специальному алгоритму, сгенерированному программой AnadigmDesigner2 (Algorithmic Method, алгоритмический метод) [2]. Алгоритмический метод формирования конфигурационных данных обладает высокой гибкостью изменения параметров схем. Применяя же управля-

ющий метод, разработчик строго ограничен предварительно скомпилированными конфигурациями.

При использовании бюджетных недорогих микроконтроллеров для динамического конфигурирования программируемых аналоговых микросхем Anadigm невозможно использование алгоритмического метода получения конфигурации, для которого необходимо выполнять достаточно сложный расчет. В этом случае применяются управляющий метод [3]. Конфигурационный протокол, рассматриваемый в статье, позволяет понять формат обмена данными между управляющим микроконтроллером и ПАИС при управляющем динамическом конфигурировании.

Конфигурационный протокол

САПР AnadigmDesigner2 собирает файл конфигурационных данных, который состоит из блока заголовка (Header Block) и одного или нескольких блоков данных (Data Block). В простейшем случае автоматической загрузки конфигурационных данных из внешней микросхемы памяти SPI EPROM вся необходимая для работы ПАИС информация будет содержаться в блоке заголовка и блоках данных, последовательно загружаемых в ПАИС. При динамическом конфигурировании управляющий микроконтроллер должен определять правильность конфигурационных данных и передавать данные в ПАИС, используя конфигурационный протокол. Существует два типа данных, составляющих конфигурационный протокол: основной (первичный) конфигурационный формат (Primary Configuration Format) и формат обновления (Update Format) (табл. 1) [4].

Первичный конфигурационный формат

Первичный конфигурационный формат используется при первой загрузке конфигу-

Таблица 1. Структура файла основной конфигурации и файла обновления конфигурации для ПАИС 2-го поколения AN221E04

	Данные	Описание
Основная конфигурация		
	00	
	00	
	00	
	00	
	00	
Блок заголовка	D5	Байт синхронизации
	B7	Индификатор JTAG ID1
	22	Индификатор JTAG ID2
	00	Индификатор JTAG ID3
	80	Индификатор JTAG ID4
	01	ADDR1
Блок данных	05	Контрольный байт
	CC	Байт адреса
	00	Адрес банка
	xx	Счетчик
	xx	Данные
	xx	Данные
	2A	Константа контроля ошибки
	00	
Обновленная конфигурация		
Блок заголовка	00	
	D5	Байт синхронизации
	01	ADDR1
Блок данных	05	Контрольный байт
	CC	Байт адреса
	00	Адрес банка
	xx	Счетчик
	xx	Данные
	xx	Данные
00	xx	Данные
	2A	Константа контроля ошибки

рационных данных после подачи питания или сброса ПАИС. Блок заголовка содержит байт синхронизации, индификатор JTAG, логический адрес и контрольный байт. В табл. 1 представлена структура файла основной конфигурации для динамически программируемой микросхемы AN221E04. Необходимо отметить, что САПР AnadigmDesigner2 при формировании файла конфигурации добавляет 5 незначащих байтов (обычно 0x00) перед блоком заголовка. Эти байты необходимы для формирования задержки,

равной по времени 40 тактам синхросигнала, для завершения цикла переустановки после подачи напряжения питания или после сброса ПАИС Anadigm.

Байт синхронизации

Конфигурационная логика ПАИС всегда предполагает наличие байта синхронизации в блоке заголовка. Передача основных и реконфигурационных данных происходит согласно конфигурационному формату (табл. 1) и всегда начинается с байта синхронизации D5 (11010101).

Индификатор JTAG ID

После передачи байта синхронизации следует индификатор JTAG ID, состоящий из 4 байтов, индивидуальных для каждого типа микросхем ПАИС (например, 0x800022B7 для AN221E04 и 0x800012B7 для AN121E04). Требование совпадения JTAG ID во время первичной конфигурации — один из способов предотвращения загрузки конфигурационных данных, не предназначенных для данного типа устройства. Индификатор ID передается один раз в составе основных конфигурационных данных, далее он не используется. В случае несовпадения JTAG ID с ожидаемым значением при загрузке конфигурационных данных они будут отклонены, а сама микросхема ПАИС сформирует на выводе ERRb сигнал ошибки.

Логический адрес ADDR1

Затем следует логический адрес ADDR1. Он используется для адресной загрузки реконфигурационных данных с помощью одного последовательного интерфейса SPI при объединении нескольких ПАИС в цепочку. Для корректной работы ПАИС адрес ADDR1 должен быть уникальным для каждой микросхемы в цепочке и не должен принимать значения 213 (D5) и 255 (FF). При необходимости одновременной загрузки одинаковых конфигурационных данных в несколько ПАИС применяется альтернативный адрес ADDR2. Для нескольких ПАИС адрес ADDR2 может иметь одинаковое значение. В отличие от основного логического адреса ADDR1, который передается в блоке заголовка, второй логический адрес ADDR2 передается в составе конфигурационных данных. Логические адреса ADDR1 и ADDR2 можно установить с помощью программы AnadigmDesigner2 в соответствующих редактируемых полях (рис. 1) [2].



Рис. 1. Конфигурирование логических адресов ADDR1 и ADDR2 в программе AnadigmDesigner2

Таблица 2. Контрольный байт

Контрольный байт							
7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1
PULLUPS 1: Установка данного конфигурационного бита активизирует внутренние подтягивающие резисторы к выводам CFG_LCb и ACTIVATE. Онджды установленные, они остаются подключенными до полной перезагрузки ПАИС. Подключение внутреннего подтягивающего резистора к выводу DIN производится с помощью установки соответствующего бита в первом блоке конфигурационных данных. Замечание. Вывод ERRb требует наличия внешнего подтягивающего резистора. 0: Отключает внутренние подтягивающие резисторы							
Зарезервирован 1: Зарезервирован 0: Значение по умолчанию							
ENDEXECUTE 1: Конфигурационные данные «теневой» SRAM будут скопированы в основную конфигурационную память, сразу после загрузки последнего конфигурационного блока. 0: Нет действия							
SRESET 1: Инициализация «мягкого» сброса ПАИС, при получении данного бита произойдет «мягкая» перезагрузка ПАИС, при этом основная и «теневая» память будут очищены. 0: Нет действия							
READ 1: Устанавливает ПАИС в режим чтения 0: Устанавливает ПАИС в режим записи							
STOP_READBACK 1: Бит может быть установлен при загрузке первичной конфигурации или обновления конфигурационных данных. При этом устанавливается внутренний флаг, запрещающий чтение конфигурационных данных, записанных в память ПАИС. При попытке чтения данных на шине ERRb будет установлено значение «лог. 0» в течение 14 тактов DCLK, которое приведет к перезагрузке ПАИС. Этот флаг может быть сброшен перезагрузкой ПАИС. 0: Разрешает считывание любых данных из ПАИС							
RESET_ALL 1: В случае возникновения ошибки в работе ПАИС на выводе ERRb будет установлено значение «лог. 0» в течение 15 тактов DCLK, ПАИС будет перезагружена, а конфигурационная память очищена. По окончании перезагрузки ПАИС будет готова к принятию основных конфигурационных данных. 0: В случае возникновения ошибки, вывод ERRb будет находиться в состоянии «лог. 0» в течение 1 такта DCLK, и ПАИС будет перезагружена в режиме обновления конфигурации. Замечание. Короткий импульс на шине ERRb будет проигнорирован всеми ПАИС в цепочке, кроме микросхемы, которая сгенерировала этот импульс.							
Зарезервирован 1: Зарезервирован 0: Значение по умолчанию							

Контрольный байт

За байтом адреса следует контрольный байт (Control Byte), который выставляет внутренние флаги ПАИС и устанавливает внутренние подтягивающие резисторы, режим работы ПАИС и действия при ошибках в работе (табл. 2).

Загружаемые в ПАИС данные размещаются в «теневом» ОЗУ, скопировать эти данные в основную память (активировать аналоговую структуру) можно, подав соответствующий сигнал «лог. 1» на вывод EXECUTE или установить в контрольном байте (CONTROL BYTE) бит ENDEXECUTE = 1. В последнем варианте по окончании загрузки конфигурационных данных в «теневую» память ПАИС автоматически копирует эти данные в основную память. Копирование данных из «теневой» в конфигурационную память осуществляется за один такт. Это позволяет менять аналоговую структуру ПАИС целиком, без потери данных.

Адрес банка и байт адреса

Конфигурационная и «теневая» память внутри ПАИС организована в виде 18 строк (банков) и 32 столбцов (байтов). Адреса BYTE и BANK определяют начальный загрузочный адрес «теневом» ОЗУ для следующего за ними блока конфигурационных данных. Бит DATA_FOLLOWS в байте адреса (BYTE ADDRESS) указывает ПАИС, является передаваемый блок данных последним или после приема данных следует ожидать передачу очередного блока. При передаче потока данных в память ПАИС переход между банками или байтами происходит автоматически. Размещение адресов «теневом» ОЗУ приведено в таблицах 3, 4.

Таблица 3. Назначение битов в байте адреса

Адресные байты								
7	6	5	4	3	2	1	0	
0	0	0	BYTE ADDRESS					Начальный байт адреса «теневом» ОЗУ для загрузки
							1: В конце передаваемых данных будет передана контрольная сумма CRC16 0: В конце блока данных будет передан байт 0x2A	
							1: За текущим передаваемым блоком данных последует еще один 0: Передаваемый блок данных является последним	
							1: Бит должен всегда установлен в 1 0: Действие не определено	
							CONSTANT 1	

Таблица 4. Структура внутренней памяти ПАИС

Банк адресов	Адресные байты					
	0x1F	...	0x10	0x0F	...	0x00
00	Нижний вспомогательный банк «теневом» ОЗУ					
01	Верхний вспомогательный банк «теневом» ОЗУ					
02	Банк А «теневом» ОЗУ САВ 1					
03	Банк В «теневом» ОЗУ САВ 1					
04	Банк А «теневом» ОЗУ САВ 2					
05	Банк В «теневом» ОЗУ САВ 2					
06	Банк А «теневом» ОЗУ САВ 3					
07	Банк В «теневом» ОЗУ САВ 3					
08	Банк А «теневом» ОЗУ САВ 4					
09	Банк В «теневом» ОЗУ САВ 4					
0A-0F	Зарезервировано					
10	Таблица преобразования, банк 0					
11	Таблица преобразования, банк 1					
12	Таблица преобразования, банк 2					
13	Таблица преобразования, банк 3					
14	Таблица преобразования, банк 4					
15	Таблица преобразования, банк 5					
16	Таблица преобразования, банк 6					
17	Таблица преобразования, банк 7					
18-1F	Зарезервировано					

Байт счетчика данных

Байт счетчика данных (DATA COUNT BYTE) указывает ПАИС, какое количество байтов будет передано в следующих за этим

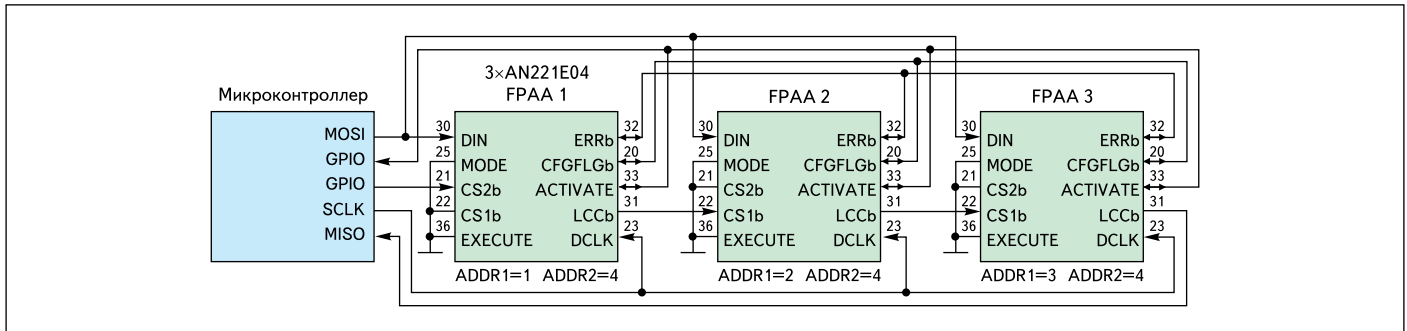


рис. 2. Упрощенная типовая схема подключения нескольких ПАИС AN221E04 к микроконтроллеру для загрузки конфигурационных данных по SPI-интерфейсу при динамическом программировании

байтом блоке конфигурационных данных. Присваивание этому полю значения 0x00 будет означать длину передаваемого блока данных 256 байт.

Конфигурационные данные и контроль ошибок при передаче данных

Передаваемые конфигурационные данные загружаются в «теневое» ОЗУ или в область таблиц преобразования согласно ранее полученным адресом (BYTE и BANK). Блок передаваемых данных может иметь длину от 1 до 256 байт.

После передачи блока данных, длину которого определяет байт DATA COUNT BYTE, ПАИС будет ожидать байт 0x2A (константа контроля ошибки). В случае получения любого другого значения вместо константы контроля ошибки ПАИС сформирует сигнал ошибки ERRb, и процесс загрузки будет прерван.

Формат обновления конфигурационных данных

Для изменения аналоговой структуры работающей ПАИС служит протокол обновления конфигурационных данных. Структура формата обновления конфигурационных данных будет полностью совпадать с основным конфигурационным форматом, за исключением индикатора JTAG ID, который не используется (табл. 1). При этом адрес микросхемы, для которой предназначены данные обновления конфигурации, будет указан в поле ADDR1.

Пример загрузки основной конфигурации и обновления

Рассмотрим загрузку основной конфигурации и обновления на примере трех ПАИС, объединенных в цепочку. На рис. 2 представлена типовая схема подключения трех ПАИС Anadigm AN221E04 в цепочку для загрузки конфигурационных данных при использовании одного интерфейса SPI. Процесс конфигурирования будет состоять из двух этапов: загрузка в ПАИС основного конфигурационного кода и загрузка обновленных данных.

Загрузка основных конфигурационных данных

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 CFGFLGb «лог. 1», который будет означать готовность ПАИС к приему конфигурационных данных. Затем микроконтроллер может подать сигнал выбора кристалла CS2b на первую микросхему в цепочке FPAА1 для инициализации приема данных на вывод DIN. Последовательность загрузки данных в цепочке FPAА1 → FPAА2 → FPAА3 будет достигаться тем, что каждая загрузившая полностью конфигурационные данные микросхема будет формировать сигнал выбора кристалла на выводе CS1b следующей за ней ПАИС. На этом этапе всем микросхемам в цепочке будет присвоены два логических адреса — ADDR1 или ADDR2, причем адрес ADDR1 будет уникальным для каждой микросхемы в цепочке. После загрузки конфигурационных данных без ошибок микроконтроллер получит сигнал завершения конфигурирования с вывода LCCb последней микросхемы в цепочке, а на общей для всех микросхем шине ACTIVATE будет установлен уровень «лог. 1», означающий, что все микросхемы сконфигурированы и готовы для приема обновленных конфигурационных данных.

На схеме, приведенной на рис. 2, выходы EXECUTE всех микросхем соединены с «землей». Для того чтобы загруженная в память микросхем конфигурация автоматически выполнялась (скопировалась в конфигурационную память из «теневой»), необходимо установить в контрольном байте (CONTROL BYTE) бит ENDEXECUTE = 1 (табл. 2).

Загрузка обновленных конфигурационных данных

Во время работы и выполнения основного конфигурационного кода все ПАИС в цепочке будут искать на шине SPI байт синхронизации (0xD5) и следующий за ним байт логического адреса ADDR1 (или ADDR2)

(табл. 1), полученного на этапе загрузки основного конфигурационного кода. При совпадении адреса передаваемых реконфигурационных данных с адресом, полученным при загрузке основной конфигурации, принимающая данные ПАИС установит на шине CFGFLGb уровень «лог. 0», в этом случае остальные микросхемы в цепочке не будут реагировать на передаваемые по интерфейсу SPI данные. После того как передача реконфигурационных данных будет завершена, на шине CFGFLGb будет установлен уровень «лог. 1». ПАИС в цепочке будут готовы к следующему приему реконфигурационных данных.

Заключение

На приведенном примере загрузки конфигурационных данных видно, насколько просто обновляются конфигурационные данные программируемых аналоговых схем Anadigm при динамическом программировании. Для изменения аналоговой структуры или ее характеристик управляющему микроконтроллеру необходимо загружать в ПАИС, используя конфигурационный протокол, предварительно откомпилированные с помощью программы AnadigmDesigner2 конфигурационные данные.

Литература

1. Щерба А. Программируемые аналоговые ИС Anadigm: подключение и загрузка конфигурационных данных с внешней микросхемы памяти при статическом конфигурировании // Современная электроника. 2008. № 8–9.
2. Полищук А. Система автоматизированного проектирования программируемых аналоговых интегральных схем AnadigmDesigner2. Часть 2. Особенности разработки проектов в среде программирования ПАИС Anadigm // Компоненты и технологии. 2005. № 8.
3. Application Note: State-Driven Control of a dpASP using a Microchip PIC. App Note 206. Anadigm, 2008.
4. AN121E04 AN221E04 Field Programmable Analog Arrays. User Manual, Anadigm Publications, UM021200-U007g. Anadigm, 2003.