Scan Engineering Telecom SPb

SETFabric-Stream PE
Аппаратно-программное решение

Описание

Версия 1.2

Код документа: SP-SFBS-00
Дата сборки: 7 июля 2016 г.
Листов в документе: 29

© 2016, ООО «Скан Инжиниринг Телеком - СПб»
http://www.setdsp.ru

История ревизий

Ревизия	Дата	Изменения
1.2	04.07.2016	Добавлена информация по поддержке ОСРВ QNX Neutrino и ЗОСРВ «Нейтрино» КПДА.10964-01.
1.1	09.11.2015	Внесены исправления в таблицу «Вариантов комплектации поставки».
1.0	—	Начальная версия

Содержание

Перечень рисунков 4

Перечень таблиц 4

Перечень сокращений и условных обозначений 5

1 Введение 6

2 Основы 7
2.1 Фундаментальные понятия 7
2.2 Возможные топологии 8
2.3 Проектирование 10

3 Состав 11
3.1 Аппаратная часть 11
3.2 Программная часть 12

4 Ресурсы 15
4.1 Интерфейс прикладного программирования 15
4.2 События 15
4.3 Статистика 15

5 SFBS задачи 17
5.1 Свойства SFBS задачи 17
5.2 Действия над SFBS задачей 17

6 SFBS каналы 18
6.1 Свойства SFBS канала 18
6.2 Действия над SFBS каналом 19

7 Примеры 20
7.1 Построение IPC пользователя 20
7.2 Перемещение данных в пределах одной процессорной системы 20
7.3 Перемещение данных между несколькими процессорными системами 21

8 Особенности 22
8.1 Использование уникального идентификатора (ID) SFBS задачи 22
8.2 Работа с SFBS задачами типами «MASTER» или «USER» 22
8.3 Использование уникального идентификатора (ID) SFBS канала 23
8.4 Работа с SFBS каналами типа «STREAM» или «BUFFER» 23

9 Варианты поставки 25

Список литературы 29

Перечень рисунков

2-1 УГО понятия «SFBS задача» 7
2-2 УГО понятия «SFBS канал» 7
2-3 УГО отношения понятий «SFBS задача владелец» и «SFBS канал» 7
2-4 УГО отношения понятий «SFBS задача назначения» и «SFBS канал» 7
2-5 УГО отношения «SFBS» задач друг к другу (тип №1) 8
2-6 УГО отношения «SFBS» задач друг к другу (тип №2) 8
2-7 УГО отношения «SFBS» задач друг к другу (тип №3) 8
2-8 Пример произвольной топологии №1 8
2-9 Пример произвольной топологии №2 9
2-10 Пример произвольной топологии №3 9
2-11 Пример произвольной топологии №4 9
3-1 Древовидная топология вычислительной системы на базе шины PCI Express 11
3-2 Основные компоненты SETFabric-Stream PE, их местоположение и взаимодействие 13
3-3 Экранная форма утилиты «sfbs_monitor» 14
7-1 Топология связей SFBS задач при построении IPC пользователя 20
7-2 IPC пользователя между двумя приложениями на процессорной системе с архитектурой «x86», «x86_64» или «e2k» 20
7-3 Отношения трех «SFBS задач» друг к другу при построении IPC пользователя 21
7-4 IPC пользователя между тремя приложениями, работающими на разных процессорных системах 21

Перечень таблиц

9-1 Варианты комплектации поставки программной части решения SETFabric-Stream PE для CPU модулей 26
9-2 Варианты комплектации поставки программной части решения SETFabric-Stream PE для DSP модулей 26
9-3 Варианты комплектации поставки программной части решения SETFabric-Stream PE для FPGA модулей 26

Перечень сокращений и условных обозначений

ADC	Analog-to-Digital Converter	15
ANSI	American National Standards Institute	15
API	Application Programming Interface	15, 23
DAC	Digital-to-Analog Converter	15
DMA	Direct Memory Access	15
DSP	Digital Signal Processor	22
FIFO	First In First Out	23, 24
FPGA	Field-Programmable Gate Array	11, 15, 25
IPC	Inter Process Communication	3, 4, 6, 20, 21
IP	Intellectual Property	15
PCIe	PCI Express	4, 6, 11, 12, 15, 20, 22–24
PCI	Peripheral Component Interconnect	4–6, 11, 12, 15, 20, 22–24
SFBS	SETFabric-Stream	3, 4, 7, 8, 10, 12, 15, 17–24
ЗАО	Закрытое Акционерное Общество	6, 15, 25
ЗОСРВ	Защищенная Операционная Система Реального Времени	2, 12, 25, 26
ОС	Операционная Система	6, 7, 12, 23, 25, 26
ОСРВ	Операционная Система Реального Времени	2, 12, 25
УГО	Условное Графическое Обозначение	4, 7, 8

1 Введение

SETFabric-Stream PE — это аппаратно-программное решение, предоставляющее разработчикам приложений простой в использовании механизм решения сложных проблем перемещения данных в многопроцессорных системах, построенных на основе шины PCI Express.

Основным местом применения решения SETFabric-Stream PE являются разрабатываемые на базе продукции производства ЗАО «Скан Инжиниринг Телеком» целевые системы с распределенной обработкой данных между несколькими процессорными системами.

В основу решения SETFabric-Stream PE ложится идея перемещения данных через пространство виртуальных каналов связи, выстроенных между процессами обработки данных.

Физическая реализация решения SETFabric-Stream PE охватывает использование шины PCI Express в качестве основной транспортной среды для перемещения данных между процессорными системами.

Использование SETFabric-Stream PE позволяет строить гибкие, масштабируемые аппаратно-программные решения распределенной обработки данных, концентрируясь в период разработки непосредственно над решением конкретных прикладных задач и организацией собственного IPC (Inter Process Communication) между ними, не неся временных затрат на реализацию и отладку поддержки межпроцессорного обмена потоками данных по шине PCIe на уровне драйверов аппаратных устройств в ОС.

SETFabric-Stream PE позволяет решить проблему реализации драйверов аппаратных устройств и библиотек их поддержки для тех разработчиков приложений, кто выберет это решение за основу для перемещения данных между исполняемыми процессами их обработки в своих целевых системах.

Решение SETFabric-Stream PE предназначено существенно сокращать временные затраты при разработке целевых систем на базе продукции производства ЗАО «Скан Инжиниринг Телеком».

2 Основы

2.1 Фундаментальные понятия

Аппаратно-программное решение SETFabric-Stream PE оперирует следующими фундаментальными понятиями:

- задача (Task);
- канал (Channel).

SETFabric-Stream PE задача (SFBS задача) представляет собой исполняемый процесс обработки данных. Исполняется такой процесс на вычислителях общего или специализированного назначения в рамках ОС, функционирующих на этих вычислителях. УГО (Условное Графическое Обозначение) SFBS задачи приведено на рисунке 2-1.

Рисунок 2-1: УГО понятия «SFBS задача»

SETFabric-Stream PE канал (SFBS канал) представляет собой виртуальное понятие, которое служит для связывания SFBS задач между собой, с целью осуществления переноса данных между ними. УГО SFBS канала приведено на рисунке 2-2. Направления переноса данных в SFBS канале всегда однонаправленное и определяется направленность стрелки УГО.

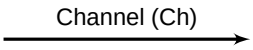

Рисунок 2-2: УГО понятия «SFBS канал»

Между понятием SFBS задача и SFBS канал существуют два типа связей:

- первый тип связи, когда SFBS канал всегда связан с одной SFBS задачей, которая называется задачей «владельцем» канала. УГО данного типа связи представленно на рисунке 2-3;

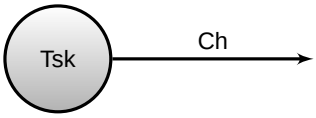

Рисунок 2-3: УГО отношения понятий «SFBS задача владелец» и «SFBS канал»

- второй тип связи, когда SFBS канал может быть присоединен к другой SFBS задаче, такая задача называется задачей «назначения» канала. УГО данного типа связи представленно на рисунке 2-4.

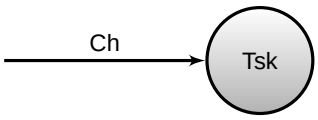

Рисунок 2-4: УГО отношения понятий «SFBS задача назначения» и «SFBS канал»

В решении SETFabric-Stream PE могут одновременно существовать множество SFBS задач и SFBS каналов. Условия существования определяют исходя из следующего:

- SFBS задача может существовать сама по себе, будучи не связанной ни с одним из SFBS каналов;
- SFBS канал не может существовать без SFBS задачи владельца;
- SFBS канал может существовать без связи с SFBS задачей назначения.

В решении SETFabric-Stream PE существует три типа отношения SFBS задач между собой. Эти типы отношений являются основными строительными блоками для построения топологий связей SFBS задач. УГО этих типов представлены на рисунках 2-5, 2-6 и 2-7.

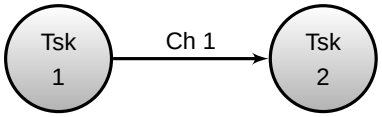

Рисунок 2-5: УГО отношения «SFBS» задач друг к другу (тип №1)

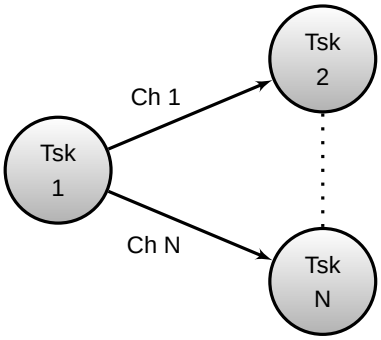

Рисунок 2-6: УГО отношения «SFBS» задач друг к другу (тип №2)

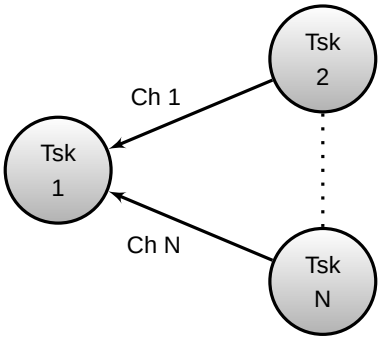

Рисунок 2-7: УГО отношения «SFBS» задач друг к другу (тип №3)

Отношения SFBS задач между собой определяется исходя из направленности SFBS канала(ов), связывающих эти SFBS задачи и самого множества SFBS задач.

2.2 Возможные топологии

Комбинация трех типов отношений SFBS задач между собой позволяет построить произвольные топологии связей между SFBS задачами. На рисунках 2-8, 2-9, 2-10 и 2-11 приведены примеры связей между SFBS задачами.

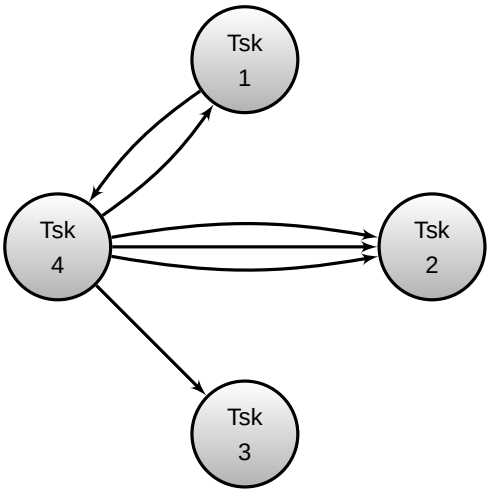

Рисунок 2-8: Пример произвольной топологии №1

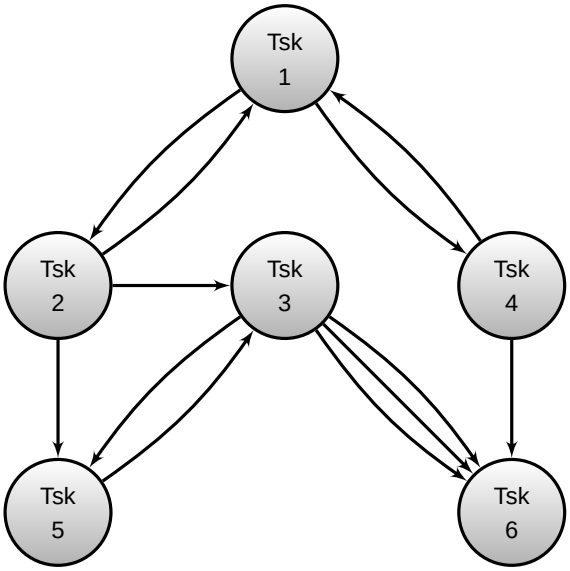

Рисунок 2-9: Пример произвольной топологии №2

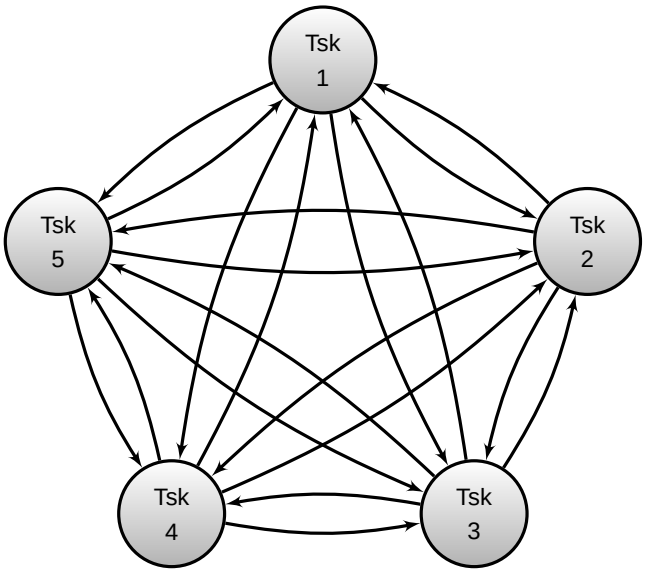

Рисунок 2-10: Пример произвольной топологии №3

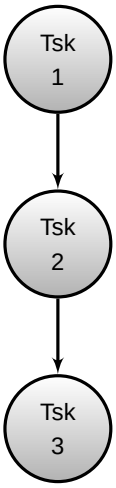

Рисунок 2-11: Пример произвольной топологии №4

2.3 Проектирование

При решении прикладной задачи разработчики приложений проходят этап ее функциональной декомпозиции. В результате декомпозиции прикладная задача разбивается на множество функциональных задач.

На следующем этапе, архитектурного проектирования, разработчики приложений определяют алгоритмы решения и технологии реализации функциональных задач. В соответствии с выбранными технологиями реализации определяется состав аппаратных средств, в которые помещается реализация для исполнения.

Если на стадии архитектурного проектирования условно заключить реализации алгоритмов решения функциональных задач в SFBS задачи, а потоки обрабатываемых данных связать с SFBS каналами, то в результате такого условного заключения и связывания можно построить топологию связей SFBS задач друг с другом, которая будет отображать реальную картину процессов перемещения данных между алгоритмами их обработки при решении прикладной задачи в целом.

3 Состав

Аппаратно-программное решение SETFabric-Stream PE состоит из двух частей:

- аппаратной части;
- программной части.

3.1 Аппаратная часть

Аппаратная часть является основой для функционирования программной и отражает области применимости решения SETFabric-Stream PE в целом. В качестве основных компонентов аппаратной части выступают процессорные системы, поддерживающие подключения к шине PCI Express на аппаратном уровне.

В соответствии с древовидной топологией шины PCI Express на ней существуют устройства типа «Root Complex», «PCIe Switch» и «Endpoint», причем во главе шины всегда стоит один «Root Complex». На рисунке 3-1 представлено классическая топология организации вычислительной системы на базе шины PCI Express.

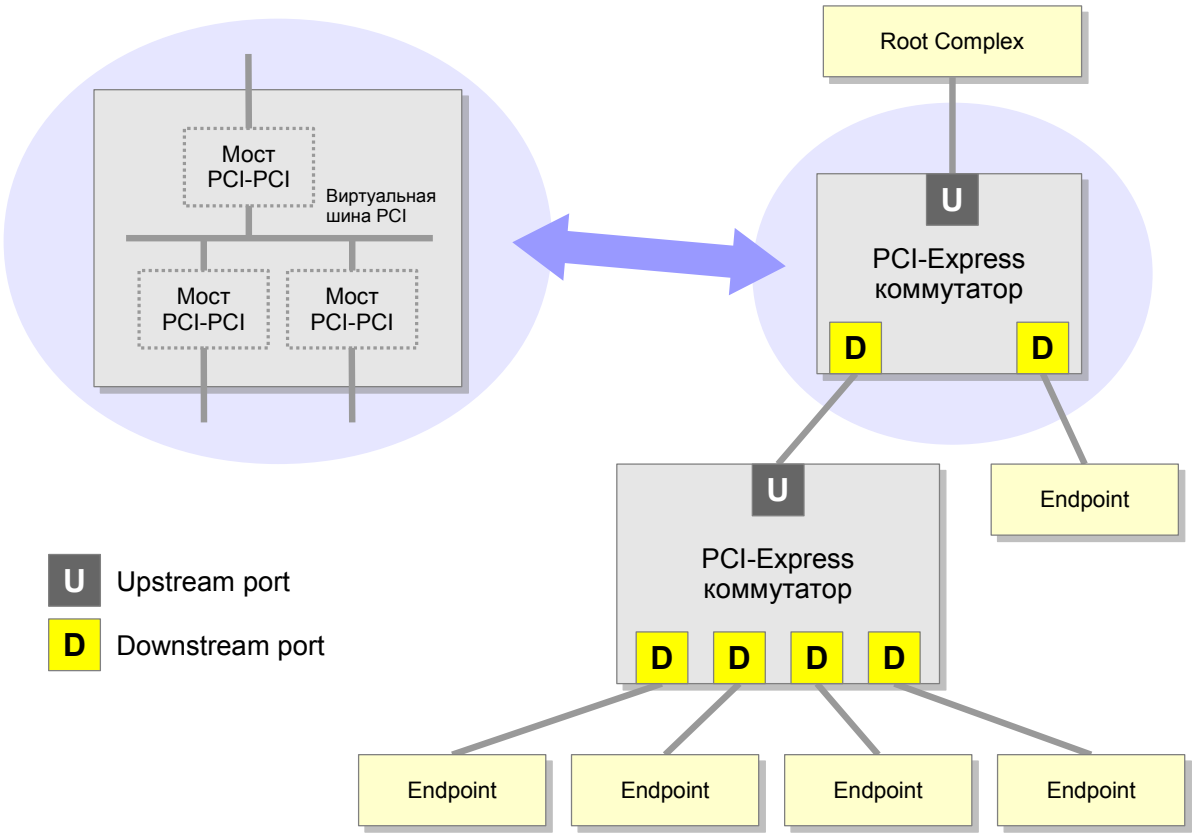

Рисунок 3-1: Древовидная топология вычислительной системы на базе шины PCI Express

В терминологии SETFabric-Stream PE под аппаратной частью целевой системы подразумевается вычислительная система, построенная на основе древовидной топологии шины PCI Express, как изображено на рисунке 3-1, а под процессорной системой подразумевается законченное схемотехническое решение, обеспечивающее работоспособность отдельно взятого процессора той или иной архитектуры, подключенного к шине PCI Express. Таким образом, в соответствии с рисунком 3-1, в рамках аппаратной части конкретной целевой системы в качестве процессорных систем в решении SETFabric-Stream PE выступают устройства «Root Complex» и «Endpoint» шины PCI Express.

В аппаратно-программном решении SETFabric-Stream PE устройством типа «Root Complex» всегда является процессорная система на базе процессора архитектур «x86» или «x86_64» производства компании «Intel» или процессор «Эльбрус-4С» архитектуры «e2k» производства компании «МЦСТ». Устройствами типа «Endpoint» могут выступать процессорные системы на базе процессоров архитектуры «c66x» производства «Texas Instruments» или на базе процессоров архитектуры «MicroBlaze», реализованных внутри FPGA производства «Xilinx».

3.2 Программная часть

Программная часть представляет собой набор драйверов, библиотек и утилит, распределенных по аппаратным частям на которых может функционировать решение SETFabric-Stream PE. Схема состава основных компонентов, их местоположение и взаимодействия между ними в программной части решения SETFabric-Stream PE представлена на рисунке 3-2.

Драйверы, библиотеки и утилиты программной части решение SETFabric-Stream PE функционируют на разных процессорных системах и поэтому их реализация зависит от той ОС, которая работает на этих процессорных системах. Программная часть решения SETFabric-Stream PE выполнена для следующих ОС:

- Linux — процессорные архитектуры «x86» и «x86_64»;
- Эльбрус ОС — процессорная архитектура «e2k»;
- QNX Neutrino — процессорная архитектура «x86»;
- ЗОСРВ «Нейтрино» КПДА.10964-01 — процессорная архитектура «x86»;
- ОСРВ SYS/BIOS — процессорная архитектура «c66x»;
- ОСРВ FreeRTOS — процессорная архитектура «MicroBlaze».

Ядром программной части решения SETFabric-Stream PE является драйвер[1] «SFBSDrvCore», работающий на процессорной системе с архитектурами «x86», «x86_64» или «e2k». Этот драйвер обеспечивает высокоскоростное перенос данных между SFBS задачами посредством SFBS каналов. Реализация этого драйвера существует для ОС Linux, Эльбрус ОС, QNX Neutrino и ЗОСРВ «Нейтрино» КПДА.10964-01.

Драйвер «SFBSDrvCore» выполнен в виде модуля ядра[2] ОС и функционально является аппаратно-независимым. Под аппаратной независимостью подразумевается то, что драйвер не обслуживает никакие устройства на шине PCI Express.

Для работы с драйвером «SFBSDrvCore» из приложений пользователя, решающих прикладные задачи целевых систем, используется библиотека «SFBSLibStream». Эта библиотека компилируется для всех, поддерживаемых решением SETFabric-Stream PE, ОС: Linux, Эльбрус ОС, QNX Neutrino, ЗОСРВ «Нейтрино» КПДА.10964-01, FreeRTOS, SYS/BIOS.

Для того, чтобы существовала возможность работы с драйвером «SFBSDrvCore» приложениям пользователя, исполняемым на процессорных системах с архитектурой «c66x» и «MicroBlaze» в окружении ОС «SYS/BIOS» или FreeRTOS, необходима драйверная поддержка процессорных систем с этими архитектурами на процессорной системе с архитектурами «x86», «x86_64» и «e2k». Для этой цели используются аппаратно-зависимые драйверы «SFBSDrvHwC66x» и «SFBSDrvHwMb». Эти драйверы осуществляют первоначальную инициализацию процессорных систем с архитектурами «c66x» и «MicroBlaze» на шине PCI Express и подключают эти процессорные системы к драйверу «SFBSDrvCore» путем регистрации в нем конкретного экземпляра процессорной системы на шине PCI Express.

Совместное функционирование экземпляров библиотеки «SFBSLibStream» в рамках приложений пользователя на процессорных системах с архитектурами «x86», «x86_64», «e2k», «c66x», «MicroBlaze», аппаратно-независимого драйвера «SFBSDrvCore», аппаратно-зависимых драйверов «SFBSDrvHwC66x» и «SFBSDrvHwMb», формирует информационное пространство решения SETFabric-Stream PE.

В зависимости от требуемой функциональной декомпозиции прикладных задач компоненты программной части решения SETFabric-Stream PE могут быть распределены на множестве процессорных систем с архитектурой «c66x» и «MicroBlaze», но в обязательном порядке в составе аппаратной части целевой системы должна присутствовать одна процессорная система с архитектурой «x86», «x86_64» или «e2k». Наличие этой процессорной системы необходимо для функционирования ядра решения SETFabric-Stream PE в виде аппаратно-независимого драйвера «SFBSDrvCore» для поддержваемых ОС.

Для оперативного контроля текущего состояния всего информационного пространства решения SETFabric-Stream PE используется диагностическая консольная утилита «sfbs_monitor», запускаемая на процессорной системе с архитектурой «x86», «x86_64» или «e2k» в окружении ОС Linux, Эльбрус ОС, QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01. Данная утилита работает и отображает данные в режиме реального времени. Пример результатов работы утилиты приведен на рисунке 3-3.

[1] Менеджер ресурсов в случае ОСРВ QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01.
[2] Для ОСРВ QNX Neutrino и ЗОСРВ «Нейтрино» КПДА.10964-01 драйвер «SFBSDrvCore» выполнен в виде менеджера ресурсов.

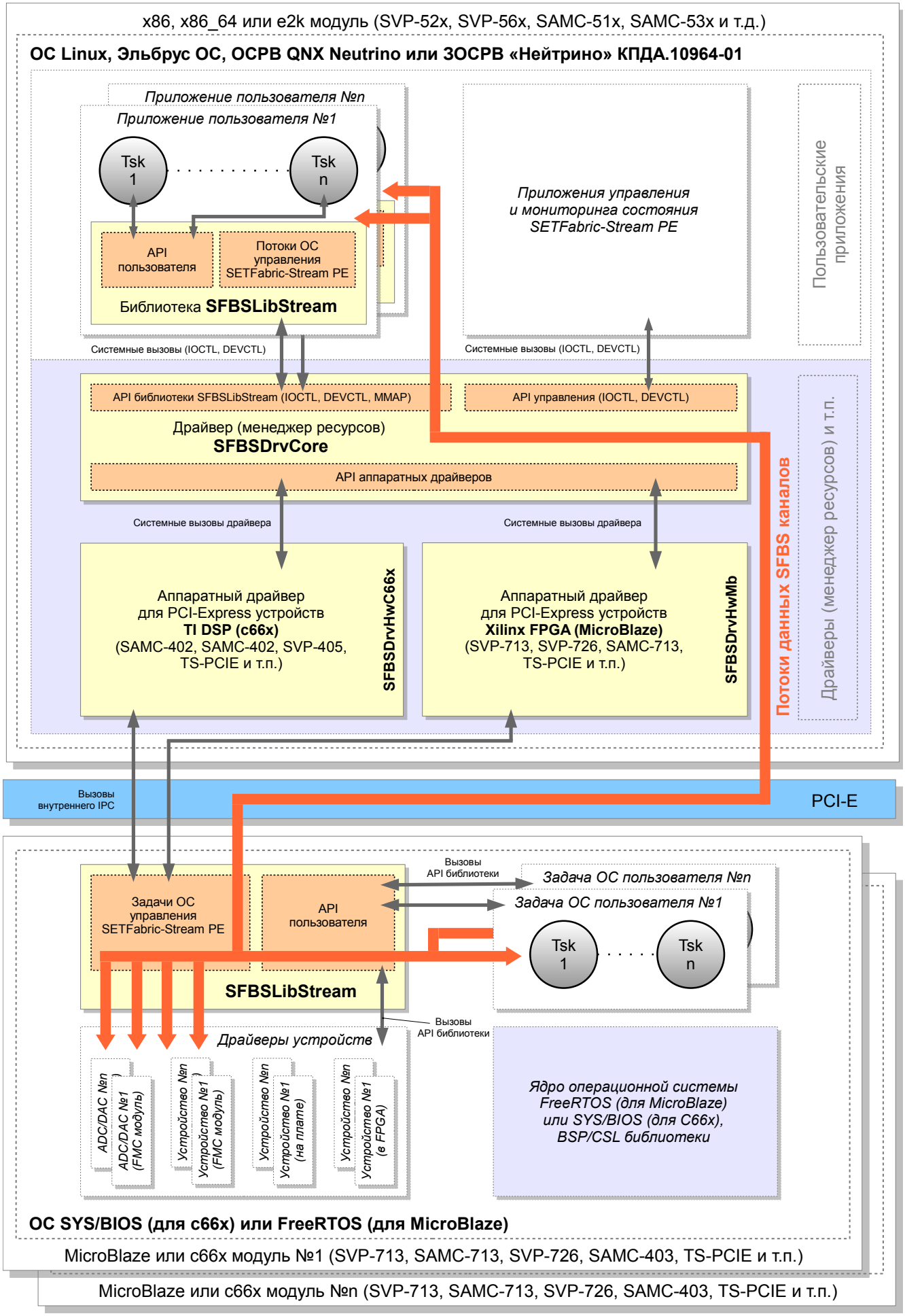

Рисунок 3-2: Основные компоненты SETFabric-Stream PE, их местоположение и взаимодействие

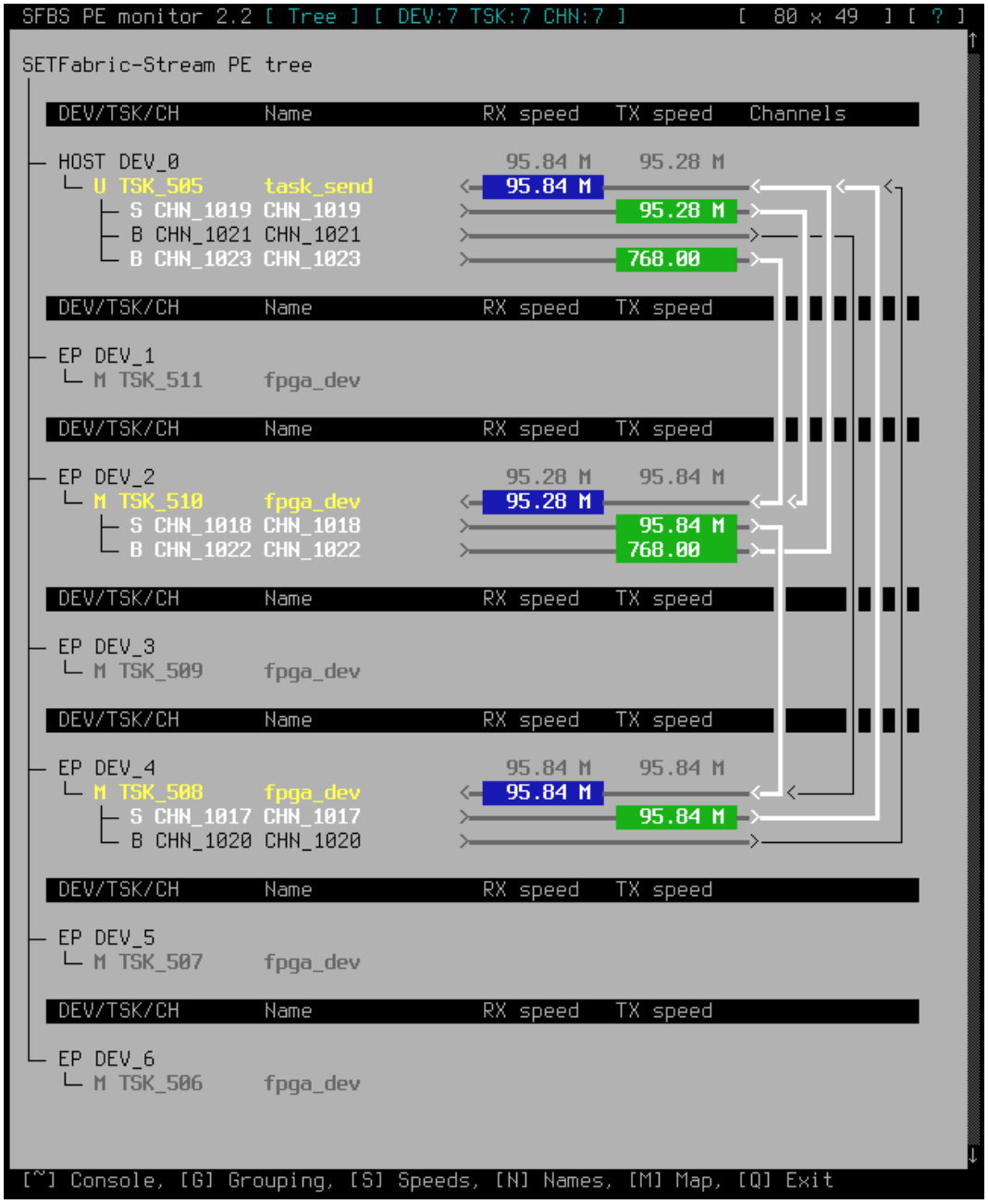

Рисунок 3-3: Экранная форма утилиты «sfbs_monitor»

Более подробную информацию о работе диагностических утилит можно получить из документа [1].

4 Ресурсы

Аппаратно-программное решение SETFabric-Stream PE предоставляет в распоряжение разработчиков приложений ресурсы, обеспечивающие эффективное решение проблем перемещения данных в разрабатываемых прикладных задачах.

Для обеспечения перемещения данных с одной процессорной системы на другую по шине PCI Express разработчики приложений могут использовать:

- DMA контроллеры;
- специальные устройства в виде IP-ядер для FPGA, генерирующие потоки данных, например IP-ядра DAC или ADC DataMover разработки ЗАО «Скан Инжиниринг Телеком» [2];
- простое копирование данных командами микропроцессоров, например, функцию «memcpy» стандартной библиотеки языка программирования ANSI «Си».

Программная часть решения SETFabric-Stream PE предоставляет для работы с информационным пространством SETFabric-Stream PE следующие ресурсы:

- интерфейс прикладного программирования;
- события;
- статистику.

4.1 Интерфейс прикладного программирования

Интерфейс прикладного программирования (API) служит для разработки приложений пользователя на языках программирования «C/C++», использующих решение SETFabric-Stream PE. Он спроектирован как единый интерфейс для всех процессорных систем на которых может функционировать решение SETFabric-Stream PE. В состав API входят функции разбитые по категориям в соответствии со своим функциональным назначением:

- управление;
- работа с SFBS задачами;
- работа с SFBS каналами.

Более подробно с интерфейсом прикладного программирования можно ознакомится в документе [3].

4.2 События

Реализация программной части решения SETFabric-Stream PE поддерживает модель событий, предназначенную для уведомления приложений пользователя о событиях происходящих в информационном пространстве SETFabric-Stream PE. К этим событиям относятся:

- создание SFBS задачи;
- удаление SFBS задачи;
- создание SFBS канала;
- удаление SFBS канала;
- присоединение SFBS канала к SFBS задаче назначения;
- отсоединение SFBS канала от SFBS задачи назначения.

Использование в приложениях пользователя механизмов доступа к событиям позволяет осуществлять динамическую реализацию функциональных задач. Динамическая реализация призванна повысить масштабируемость и гибкость архитектурных решений при решении прикладных задач.

4.3 Статистика

Реализация программной части решения SETFabric-Stream PE обеспечивает сбор, обработку и доступ к информации оперативного состояния, статистики, всего информационного пространства SETFabric-Stream PE. Доступ к статистике осуществляется из диагностических утилит, входящих в состав программной части решения SETFabric-Stream PE.

Использование доступа к статистике на стадии реализации функциональных задач целевых систем облегчает процесс отладки приложений пользователя.

Использование доступа к статистике в периоды тестовой, опытной и промышленной эксплуатации всей реализации прикладных задач на целевых системах позволяет определять реальную составляющую информационных потоков в конкретных условиях эксплуатации.

5 SFBS задачи

В рамках информационного пространства SETFabric-Stream PE SFBS задачи обладают следующими свойствами:

- уникальный идентификатор (ID);
- символьное имя;
- тип (MASTER или USER);
- количество присоединенных каналов.

Над SFBS задачами можно осуществлять следующие действия:

- создать;
- удалить.

5.1 Свойства SFBS задачи

Свойство «Уникальный идентификатор (ID)»

Данное свойство служит для однозначной идентификации конкретной SFBS задачи в пределах информационного пространства SETFabric-Stream PE. Этот идентификатор может назначаться разработчиком статически на стадии проектирования, либо запрашиваться динамически, посредством сервисной функции информационного пространства SETFabric-Stream PE. В пределах одного информационного пространства SETFabric-Stream PE может существовать одновременно не более 511 уникальных идентификаторов SFBS задач.

Свойство «Символьное имя»

Данное свойство носит чисто информативный характер, не является уникальным, в основном используется для визуальной идентификации SFBS задач в диагностических утилитах информационного пространства SETFabric-Stream PE и может произвольно назначаться разработчиком.

Свойство «Тип»

Данное свойство служит признаком системного назначение SFBS задачи. Беря за основу клиент-серверную технологию разработки программного обеспечения, чтобы облегчить разработчикам способ идентификации SFBS задачи в информационном пространстве SETFabric-Stream PE введены два условных типа «MASTER» или «USER», которые функционально предназначены для разделения SFBS задач на два типа: серверные и клиентские. Тип «MASTER» относится к серверным задачам, тип «USER» к клиентским задачам. Какой тип назначать SFBS задачам разработчик определяет самостоятельно на стадии проектирования.

Свойство «Количество присоединенных каналов»

Данное свойство носит информативный характер и определяет сколько SFBS каналов присоединено к SFBS задаче. Новое присоединение происходит каждый раз, когда устанавливается связь между двумя SFBS задачами посредством одного SFBS канала.

5.2 Действия над SFBS задачей

Действие «Создать»

Данное действие позволяет создать SFBS задачу в информационном пространстве SETFabric-Stream PE. В качестве входных параметров при создании служат следующие свойства: «Уникальный идентификатор (ID)», «Символьное имя» и «Тип (MASTER или USER)». Если не указывать при создании SFBS задачи свойство «Символьное имя», то оно будет создано автоматически по шаблону «TSK_ID», где вместо ID будет номер уникального идентификатора SFBS задачи.

Действие «Удалить»

Данное действие удаляет SFBS задачу из информационного пространства SETFabric-Stream PE. Удаление SFBS задачи приводит к удалению всех SFBS каналов, для которых SFBS задача является владельцем. При наличии не нулевого значения в свойстве «Количество присоединенных каналов» все присоединенные к удаляемой SFBS задаче SFBS каналы будут автоматически отсоединены от SFBS задачи.

6 SFBS каналы

В рамках информационного пространства SETFabric-Stream PE SFBS каналы имеют следующие свойства:

- уникальный идентификатор (ID);
- символьное имя;
- тип (STREAM или BUFFER);
- ID задачи владельца;
- ID задачи назначения;
- размер выделяемого буфера (если тип BUFFER).

Над SFBS каналами можно осуществлять следующие действия:

- создать;
- удалить;
- присоединить к задаче назначения;
- отсоединить от задачи назначения;
- отправить данные;
- принять данные;
- назначить размер буфера (если тип BUFFER).

6.1 Свойства SFBS канала

Свойство «Уникальный идентификатор (ID)»

Данное свойство служит для однозначной идентификации конкретного SFBS канала в пределах информационного пространства SETFabric-Stream PE. Этот идентификатор может назначаться разработчиком статически на стадии проектирования, либо запрашиваться динамически, посредством сервисной функции информационного пространства SETFabric-Stream PE. В пределах одного информационного пространства SETFabric-Stream PE может существовать одновременно не более 1023 уникальных идентификаторов (ID) SFBS каналов.

Свойство «Символьное имя»

Данное свойство носит чисто информативный характер, не является уникальным, в основном используется для визуальной идентификации SFBS канала в диагностических утилитах информационного пространства SETFabric-Stream PE и может произвольно назначаться разработчиком.

Свойство «Тип»

Данное свойство определяет характер и способ перемещения данных ядром информационного пространства SETFabric-Stream PE через SFBS канал. Тип «BUFFER» определяет обычные данные, которые имеют конечную размерность. Тип «STREAM» определяет потоковые данные, которые бесконечны по своей природе.

Свойство «ID задачи владельца»

Данное свойство определяет уникальный идентификатор (ID) SFBS задачи которой принадлежит SFBS канал. В информационном пространстве SETFabric-Stream PE SFBS канал не может существовать без SFBS задачи владельца. У каждого SFBS канала существует одна SFBS задача владелец. SFBS задача владелец определяется при создании SFBS канала.

Свойство «ID задачи назначения»

Данное свойство определяет уникальный идентификатор (ID) SFBS задачи к которой присоединен SFBS канал. В информационном пространстве SETFabric-Stream PE связи между SFBS задачами устанавливаются путем присоединения SFBS канала к любой из SFBS задач информационного пространства SETFabric-Stream PE. При этом устанавливается однонаправленная связь между SFBS задачей владельцем SFBS канала и SFBS задачей назначения, к которой присоединен SFBS канал. SFBS задача назначения определяется при вызове сервисной функции присоединения информационного пространства SETFabric-Stream PE.

Свойство «Размер выделяемого буфера»

Данное свойство относится к каналам SFBS с типом «BUFFER» и определяет размер выделяемого буфера памяти, используемого для хранения перемещенных данных от SFBS задачи владельца SFBS канала к SFBS задаче назначения.

6.2 Действия над SFBS каналом

Действие «Создать»

Данное действие позволяет создать SFBS канал в информационном пространстве SETFabric-Stream PE. В качестве входных параметров при создании служат свойства «Уникальный идентификатор (ID)» SFBS канала: «Символьное имя», «Тип (STREAM или BUFFER)», «Размер выделяемого буфера» и «Уникальный идентификатор (ID)» SFBS задачи владельца. Если не указывать при создании SFBS канала свойство «Символьное имя», то оно будет создано автоматически по шаблону «CHN_ID», где вместо ID будет номер уникального идентификатора (ID) SFBS канала.

Действие «Удалить»

Данное действие удаляет SFBS канал из информационного пространства SETFabric-Stream PE. При удалении SFBS канала происходит автоматическое отключение от SFBS задачи владельца и SFBS задачи назначения. Если SFBS канал типа «BUFFER», то происходит автоматическое освобождение выделенного буфера для хранения перемещаемых данных через SFBS канал на стороне SFBS задачи назначения.

Действие «Присоединить к задаче назначения»

Данное действие присоединяет SFBS канал к конкретной SFBS задаче назначения. В результате присоединения устанавливается однонаправленная связь между SFBS задачей владельцем SFBS канала и SFBS задачей назначения.

Действие «Отсоединить от задачи назначения»

Данное действие отсоединяет SFBS канал от конкретной SFBS задачи назначения. В результате отсоединения разрывается однонаправленная связь между SFBS задачей владельцем SFBS канала и SFBS задачей назначения.

Действие «Отправить данные»

Данное действие осуществляет перемещение данных от SFBS задачи владельца SFBS канала к SFBS задаче назначения.

Действие «Принять данные»

Данное действие ожидает перемещения данных от SFBS задачи владельца SFBS канала к SFBS задаче назначения.

7 Примеры

7.1 Построение IPC пользователя

Используя отношения SFBS задач между собой можно построить IPC пользователя, обеспечивающий информационный обмен между функциональными задачами, заключенными в SFBS задачи. Для этого необходимо использовать два SFBS канала типа «BUFFER», и построить топологию связей между SFBS задачами, как показано на рисунке 7-1.

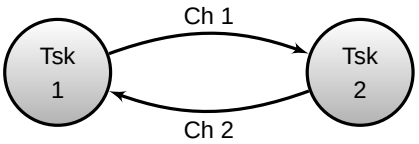

Рисунок 7-1: Топология связей SFBS задач при построении IPC пользователя

Используя предпочтительный формат представления передаваемых данных между функциональными задачами, можно реализовать обработчик передаваемых данных от одной SFBS задачи к другой и поместить его реализацию в обе задачи. В результате такой обработчик может стать основой для построения простого обмена сообщениями, являющегося одной из форм IPC.

7.2 Перемещение данных в пределах одной процессорной системы

Используя решение SETFabric-Stream PE можно реализовать процесс перемещения данных между несколькими SFBS задачами в пределах одной процессорной системы с архитектурами «x86», «x86_64» или «e2k» без участия других систем с архитектурами «c66x» или «MicroBlaze». В этом случае, разместив SFBS задачи в разных приложениях пользователя, весь процесс перемещения данных из одного приложения в другое будет происходить в памяти процессорной системы с архитектурами «x86», «x86_64» или «e2k» без участия шины PCI Express.

Простым примером топологии связей между SFBS задачами в такой конфигурации может выступать топология, предназначенная для построения IPC пользователя, представленная на рисунке 7-1. Если поместить SFBS задачи этой топологии в разные приложения пользователя, то процесс перемещения потоков данных IPC пользователя будет таким, как представлен на рисунке 7-2.

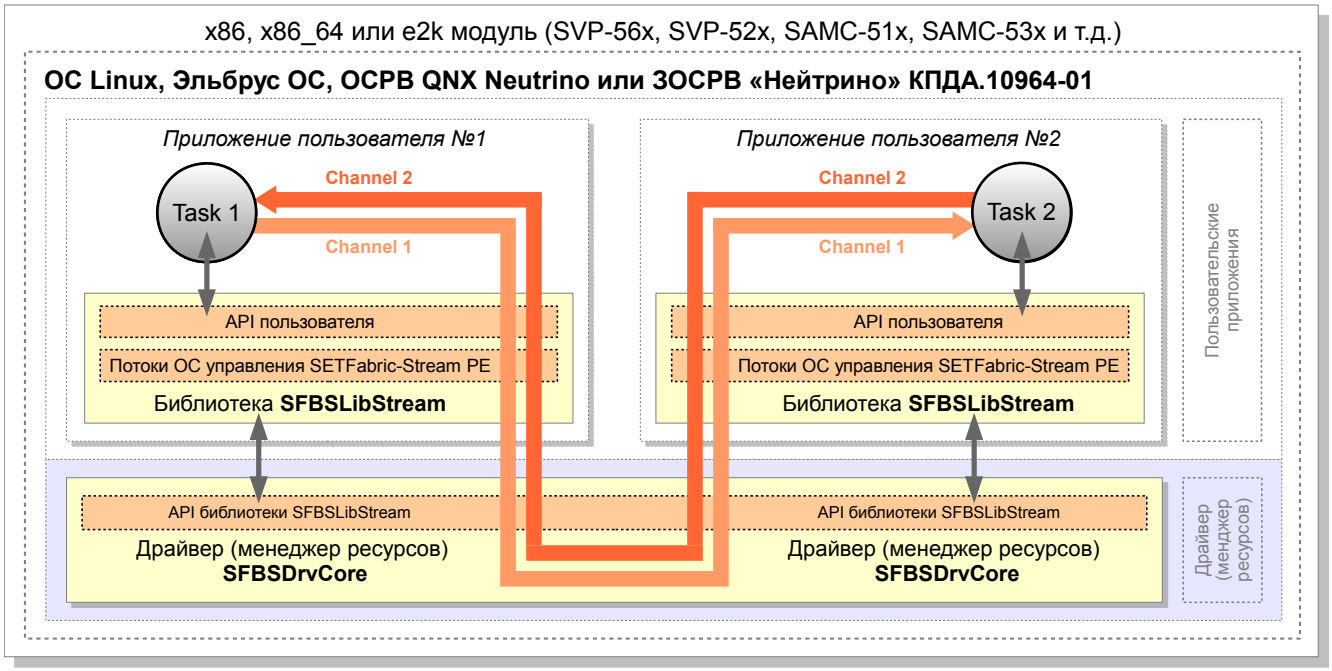

Рисунок 7-2: IPC пользователя между двумя приложениями на процессорной системе с архитектурой «x86», «x86_64» или «e2k»

7.3 Перемещение данных между несколькими процессорными системами

Используя решение SETFabric-Stream PE можно реализовать процесс перемещения данных между несколькими SFBS задачами, реализующими функциональные задачи на разных процессорных системах с архитектурами «x86», «x86_64», «e2k», «c66x» и «MicroBlaze».

В качестве простого примера возьмем топологию связей между SFBS задачами, реализующими IPC пользователя, как показанно на рисунке 7-3, и поместим ее в процессорные системы с архитектурами «x86_64»[1], «c66x» и «MicroBlaze», как показанно на рисунке 7-4.

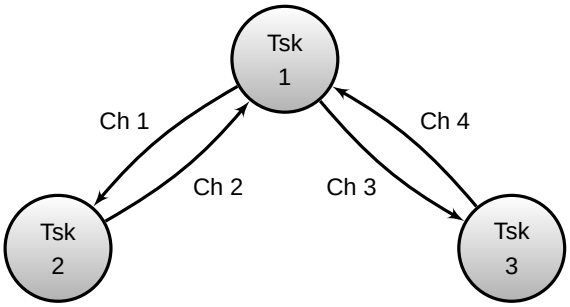

Рисунок 7-3: Отношения трех «SFBS задач» друг к другу при построении IPC пользователя

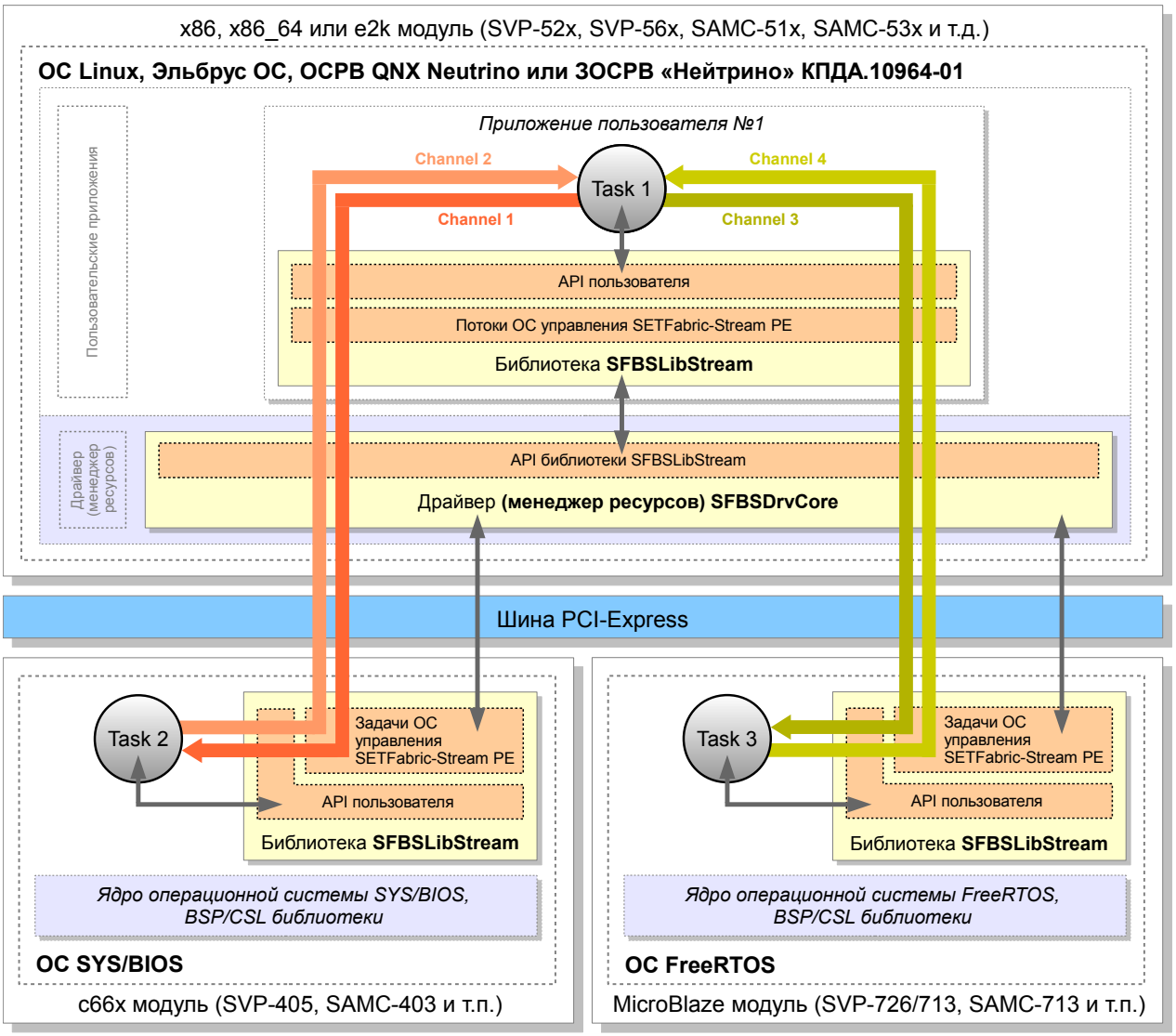

Рисунок 7-4: IPC пользователя между тремя приложениями, работающими на разных процессорных системах

[1] Вместо процессорной системы с архитектурой «x86_64» может быть система с архитектурой «x86» или «e2k».

8 Особенности

8.1 Использование уникального идентификатора (ID) SFBS задачи

Выбор статического или динамического способа назначения уникального идентификатора (ID) SFBS задачам зависит от личных предпочтений разработчиков. Однако следует понимать тот факт, что при статическом назначении уникального идентификатора (ID) масштабирование таких решений в пределах одной целевой системы объединённой информационным пространством SETFabric-Stream PE возможно будет осложненно.

Примером такой сложности может служить организация исполняемого кода прошивки, допустим для DSP процессорной системы архитектуры «c66x», которая могла бы быть откомпилирована один раз и одновременно запущена на нескольких DSP процессорных системах в пределах одной целевой системы. В этом случае, статическое назначение уникальных идентификаторов (ID) потребует, либо размещения самих идентификаторов в какой то энергонезависимой памяти каждой процессорной системы на стадии ее производства, либо, можно его жестко фиксировать в самом коде прошивки на стадии компиляции самой прошивки.

В случае с жёсткой фиксацией уникального идентификатора (ID) на стадии компиляции в самом коде прошивки не возможно будет корректно запустить работу этой прошивки более чем на одной DSP процессорной системе. Дело в том, что исполняемая в коде прошивки функция создания SFBS задачи не даст создать более одной SFBS задачи с одинаковым номером уникального идентификатора (ID) в пределах одного информационного пространства SETFabric-Stream PE. Фактически такой подход к назначению уникального идентификатора (ID) ведет к порождению уникальных прошивок для каждого из DSP процессоров, входящих в состав целевой системы. В каких то целевых системах такой подход к назначению уникального идентификатора (ID) приемлем, в каких то нет.

В случае с энергонезависимой памятью в коде прошивки необходимо организовать доступ к этой памяти для извлечения уникального идентификатора (ID) перед передачей его в качестве параметра в функцию создания SFBS задачи. В этом случае, код прошивки может запуститься на нескольких DSP процессорных системах, но нужно четко гарантировать несовпадение значения уникального идентификатора (ID) в энергонезависимой памяти каждой из DSP процессорных систем. В противном случае, если у каких-либо DSP процессорных систем уникальные идентификаторы (ID) совпадут по значению, то исполняемая в коде прошивки функция создания SFBS задачи не даст создать более одной SFBS задачи с таким значением уникального идентификатора (ID), и как следствие, одна из прошивок не сможет нормально функционировать.

Если взять за основу динамическое назначение уникального идентификатора (ID) в пределах одной целевой системы, объединенной информационным пространством SETFabric-Stream PE, то возможно спроектировать универсальный код прошивки для DSP процессорных систем и запускать его одновременно на нескольких DSP процессорных системах. В этом случае несовпадение значения уникального идентификатора (ID) будет гарантирована самим информационным пространством SETFabric-Stream PE.

Помимо статического и динамического назначения уникального идентификатора (ID) в пределах одной целевой системы, объединенной информационным пространством SETFabric-Stream PE, возможно смешанное назначение. Т. е. одной части SFBS задач можно задать значения статически, другой части динамически. При статическом назначении необходимо соблюдать уникальность значения уникального идентификатора (ID) в конкретном информационном пространстве SETFabric-Stream PE.

В не зависимости от выбранной стратегии назначения уникального идентификатора (ID) SFBS задачам необходимо помнить, что его значение уникально в пределах одного информационного пространства SETFabric-Stream PE.

8.2 Работа с SFBS задачами типами «MASTER» или «USER»

Выбор назначаемого типа «MASTER» или «USER» SFBS задаче при ее создании обуславливается представлениями разработчика о функциональном назначении реализуемой прикладной задачи. Так как изначально градация типов «MASTER» и «USER» отражает клиент-серверную модель разработки программного обеспечения, то целесообразно проводить функциональную декомпозицию прикладных задач целевых систем так, чтобы реализация их алгоритмов обработки данных потенциально укладывалась в эту модель.

По мнению разработчиков решения SETFabric-Stream PE, целесообразней всего производить функциональную декомпозицию прикладных задач так, чтобы их реализации содержали хотя бы по одной SFBS задаче типа «MASTER» на каждой из процессорных систем с архитектурами «c66x» и «MicroBlaze». На процессорных системах «x86», «x86_64» или «e2k», по мнению разработчиков решения SETFabric-Stream PE, лучше всего использовать SFBS задачи типа «USER».

Так как в рамках целевых систем процессорные системы с архитектурами «c66x» и «MicroBlaze», как правило, являются устройствами типа «Endpoint» на шине PCI Express и создают хотя бы одну SFBS задачу типа «MASTER» на каждом из «Endpoint», назначив свойство SFBS задачи «Символьное имя» используют алгоритм его формирования на основе уникальных значений номеров шин, устройств и функций шины PCI Express, то в пределах информационного пространства SETFabric-Stream PE появляется возможность однозначной идентификации каждого устройства «Endpoint». Такая

однозначная идентификация важна при создании целевых систем, на которых реализации прикладных задач разнесены по множеству устройств типа «Endpoint» и большому количеству приложений пользователя на устройстве типа «Root Complex».

В составе функций информационного пространства SETFabric-Stream PE есть функция, которая возвращает список SFBS задач зарегистрированных на момент времени вызова данной функции, производя их выборку по типу: «MASTER» или «USER» SFBS задачи.

В любом случае, право выбора стратегии размещения функциональных задач по «Endpoint» устройствам шины PCI Express и «Root Complex», стратегии именования задач, типу и количеству, остается за разработчиками конкретных целевых систем.

8.3 Использование уникального идентификатора (ID) SFBS канала

Выбор статического или динамического способа назначения уникального идентификатора (ID) SFBS каналам зависит от личных предпочтений разработчиков. В целом, проблематика предпочтения этого выбора такая же как и при выборе уникального идентификатора (ID) SFBS задач. Вариант использования смешанного назначения уникального идентификатора (ID) SFBS каналу тоже имеет право на жизнь.

В любом случае, всегда нужно помнить, что этот идентификатор уникален и при попытке создать в пределах одного информационного пространства SETFabric-Stream PE нескольких SFBS каналов с одним и тем же значением этого идентификатора приведет к отказу в создании более чем одного SFBS канала.

8.4 Работа с SFBS каналами типа «STREAM» или «BUFFER»

Выбор типа «STREAM» или «BUFFER» при создании SFBS канала определяется природой перемещаемых через канал данных. Обслуживание SFBS каналов этих типов со стороны ядра информационного пространства SETFabric-Stream PE носит одинаковый характер, но имеет различие в работе с функциями API библиотеки «SFBSLibStream» для приложений пользователя.

В реализации ядра решения SETFabric-Stream PE для перемещения данных SFBS каналов всех типов используется концепция кольцевых буферов FIFO. Размер и количество этих буферов конечны, носят настраиваемый характер при загрузке драйвера «SFBSDrvCore», определяются опциями драйвера «SFBSDrvCore», задаваемыми при его загрузке.

При использовании SFBS каналов типа «BUFFER», для хранения перемещаемых пользователем данных, ядром решения SETFabric-Stream PE выделяются специальные буфера, размер которых определяется опцией «Размер выделяемого буфера» SFBS каналов. Выделение буферов происходит на стороне SFBS задач назначения, т. е. в памяти той процессорной системы, на которой исполняется SFBS задача назначения. Максимальный размер выделяемой памяти для данного буфера зависит от свободного объёма памяти, доступной для работы функции выделения памяти в конкретной ОС, работающей на процессорной системе.

При использовании SFBS каналов типа «BUFFER» пользователь должен помнить, что функции отправки и приёма данных в этот тип SFBS канала носят блокируемый характер с точки зрения освобождения буфера. Это означает, что:

- поток исполнения отправки данных в канал типа «BUFFER» SFBS задачи владельца SFBS канала приостановит свое выполнение до тех пор, пока данные не будут доставленны в буфер SFBS задачи назначения и поток SFBS задачи назначения не освободит этот буфер для дальнейшего использования потоком SFBS задачи владельца;
- поток исполнения приёма данных из канала типа «BUFFER» SFBS задачи назначения приостанавливает свое выполнение до тех пор, пока данные не будут доставлены в буфер SFBS задачи назначения и не станут доступны для чтения.

Размер перемещаемых данных через SFBS канал типа «BUFFER» может быть меньше или больше размера одного кольцевого буфера ядра решения SETFabric-Stream PE. Для перемещения данных ядром будет использовано от одного до нескольких кольцевых буферов FIFO, необходимых для перемещения всех данных. В случае отсутствия свободных кольцевых буферов FIFO при перемещении данных пользователя функция отправки данных завершится с ошибкой. Фактически, это означает потерю буфера перемещаемых данных внутри ядра решения SETFabric-Stream PE.

При использовании SFBS каналов типа STREAM ядром решения SETFabric-Stream PE для хранения перемещаемых данных пользователя используются только кольцевые буфера FIFO. Функции отправки данных через канал SFBS типа STREAM не являются блокируемыми, но в случае отсутствия свободных кольцевых буферов FIFO функция оправки данных завершится с ошибкой. Фактически, это означает потерю буфера перемещаемых данных внутри ядра решения SETFabric-Stream PE.

Приём перемещаемых данных пользователя посредством SFBS каналов типа «STREAM» от SFBS задачи владельца осуществляется SFBS задачей назначения через функции обратного вызова со стороны ядра, в задачу которых входит

перемещение данных из кольцевых буферов FIFO в требуемые участки памяти приложений пользователя и в обязательном порядке необходимо освобождение кольцевых буферов FIFO путем выхода из функций обратного вызова.

Важным моментом при проектировании целевых систем на базе решения SETFabric-Stream PE является понимание того, что:

- перемещение данных пользователя через SFBS каналы любых типов между SFBS задачами осуществляется ядром решения через одни и те же кольцевые буфера FIFO;
- своевременное выполнение процедуры освобождения кольцевых буферов FIFO при использовании SFBS каналов типа «STREAM» путем выхода из функций обратного вызова со стороны ядра влияет на наличие свободных кольцевых буферов FIFO и, как следствие, на работоспособность информационного пространства SETFabric-Stream PE с конкретным экземпляром процессорной системы (с конкретным устройством «Endpoint» шины PCI Express).

9 Варианты поставки

Аппаратно-программное решение SETFabric-Stream PE может поставляться в нескольких видах:

- в составе системных платформ производства ЗАО «Скан Инжиниринг Телеком», в предустановленном виде на аппаратных средствах платформы;
- на отдельных носителях типа CD-ROM, для самостоятельной установки на аппаратные средства производства ЗАО «Скан Инжиниринг Телеком».

Комплектность поставки программной части включает базовую и опциональную части.

Базовая часть предназначенна для процессорной системы с архитектурами «x86», «x86_64» или «e2k» и включает в себя:

- библиотеку «SFBSLibStream» для ОС Linux Эльбрус ОС, QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01 с примерами использования;
- аппаратно-независимый драйвер[1] «SFBSDrvCore»;
- утилиты диагностики.

Опциональная часть зависит от архитектуры процессорной системы («c66x» или «MicroBlaze») и включает в себя:

- библиотеку «SFBSLibStream» для ОС SYS/BIOS или FreeRTOS с примерами использования;
- аппаратно-зависимый драйвер «SFBSDrvHwC66x» или «SFBSDrvHwMb» для ОС Linux, Эльбрус ОС, QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01.

Базовая и опциональная части поставляются в:

- двоичном формате библиотеки «SFBSLibStream» для процессорной системы с архитектурами «x86», «x86_64» или «e2k» под управлением ОС Linux, Эльбрус ОС, QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01 и для процессорной системы «c66x» или «MicroBlaze» под управлением ОС FreeRTOS и SYS/BIOS, для двух режимов работы: отладки (debug) и выпуска (release). Заголовочные файлы библиотек поставляются в исходном виде текстов программ;
- двоичном формате драйвера и утилиты мониторинга для ОС Linux, Эльбрус ОС, QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01;
- исходном виде текстов программ с примерами использования библиотек для ОС Linux, Эльбрус ОС, QNX Neutrino, ЗОСРВ «Нейтрино» КПДА.10964-01, FreeRTOS и SYS/BIOS.

Поддерживаемые ОС их версии:

- Linux (ядро 2.6.32 или выше);
- QNX Neutrino (версии 6.5.0 или 6.6.0);
- ЗОСРВ «Нейтрино» КПДА.10964-01;
- SYS/BIOS (версия 6.35.03.47 или выше);
- FreeRTOS (версия 7.5.2 или выше).

Комплект поставки опциональной программной части для процессорной системы с архитектурой «MicroBlaze» дополнительно включает в себя проект аппаратной платформы «Core-0» (см. документ [4]), разворачиваемой в FPGA микросхеме производства фирмы «Xilinx». Функционирование программной части решения SETFabric-Stream PE осуществляется на базе данной аппаратной платформы.

В случае приобретения аппаратно-программного решения SETFabric-Stream PE в составе системных платформ производства ЗАО «Скан Инжиниринг Телеком», определение полной комплектации поставки решения производится на стадии формирования заказа по закупке аппаратных средств производства ЗАО «Скан Инжиниринг Телеком», исходя из их функционального назначения и количественного состава платформы.

В случае приобретения аппаратно-программного решения SETFabric-Stream PE на отдельных носителях типа CD-ROM необходимо предварительно согласовать список и количественный состав аппаратных средств производства ЗАО «Скан Инжиниринг Телеком», на которых предполагается развернуть решение.

Варианты комплектации поставки программной части решения SETFabric-Stream PE представлены в таблицах 9-1, 9-2 и 9-3.

[1] Менеджер ресурсов в случае ОСРВ QNX Neutrino или ЗОСРВ «Нейтрино» КПДА.10964-01.

Таблица 9-1: Варианты комплектации поставки программной части решения SETFabric-Stream PE для CPU модулей

Модули CPU	Операционная система	Поддержка		Код заказа
		FPGA	DSP	
SVP-522 SVP-524 SVP-562 SVP-564 SAMC-512 SAMC-514 SAMC-532 SAMC-534	Astra Linux Special Edition 1.2	+	–	RU.СЖАН.34001-01
		–	+	RU.СЖАН.34001-02
		+	+	RU.СЖАН.34001-03
	Astra Linux Special Edition 1.3	+	–	RU.СЖАН.34001-04
		–	+	RU.СЖАН.34001-05
		+	+	RU.СЖАН.34001-06
	МСВС 5.0	+	–	RU.СЖАН.34001-07
		–	+	RU.СЖАН.34001-08
		+	+	RU.СЖАН.34001-09
	Ubuntu 12.04 LTS	+	–	RU.СЖАН.34001-10
		–	+	RU.СЖАН.34001-11
		+	+	RU.СЖАН.34001-12
	ЗОСРВ «Нейтрино» КПДА.10964-01	+	–	RU.СЖАН.34001-13
		–	+	RU.СЖАН.34001-14
		+	+	RU.СЖАН.34001-15
	ОС «Эльбрус»	+	–	RU.СЖАН.34001-16
		–	+	RU.СЖАН.34001-17
		+	+	RU.СЖАН.34001-18
	QNX Neutrino 6.5.0	+	–	RU.СЖАН.34001-20
		–	+	RU.СЖАН.34001-21
		+	+	RU.СЖАН.34001-22
	QNX Neutrino 6.6.0	+	–	RU.СЖАН.34001-23
		–	+	RU.СЖАН.34001-24
		+	+	RU.СЖАН.34001-25

Таблица 9-2: Варианты комплектации поставки программной части решения SETFabric-Stream PE для DSP модулей

Модули DSP	Код заказа
SAMC-402	RU.СЖАН.34005-01
SAMC-403	RU.СЖАН.34005-02
SVP-405	RU.СЖАН.34005-03
SVP-407	RU.СЖАН.34005-04
SVP-465	RU.СЖАН.34005-05
TS-PCIE	RU.СЖАН.34005-06

Таблица 9-3: Варианты комплектации поставки программной части решения SETFabric-Stream PE для FPGA модулей

Модули FPGA	FPGA микросхема		Код заказа
	Тип	Порядковый номер	
SAMC-713	LX130T	1	RU.СЖАН.30288-01
	LX195T	1	RU.СЖАН.30288-02
	LX240T	1	RU.СЖАН.30288-03
	LX365T	1	RU.СЖАН.30288-04
	SX315T	1	RU.СЖАН.30288-05
SAMC-715	LX130T	1	RU.СЖАН.30304-01
	LX195T	1	RU.СЖАН.30304-02
	LX240T	1	RU.СЖАН.30304-03
	LX365T	1	RU.СЖАН.30304-04
	SX315T	1	RU.СЖАН.30304-05

Продолжение таблицы на следующей странице

Продолжение таблицы 9-3

Модули FPGA	FPGA микросхема		Код заказа
	Тип	Порядковый номер	
SAMC-717	LX130T	1	RU.СЖАН.30320-01
	LX195T	1	RU.СЖАН.30320-02
	LX240T	1	RU.СЖАН.30320-03
	LX365T	1	RU.СЖАН.30320-04
	SX315T	1	RU.СЖАН.30320-05
SVP-713	LX130T-1	1	RU.СЖАН.30272-01
	LX130T-2	1	RU.СЖАН.30272-02
	LX195T-1	1	RU.СЖАН.30272-03
	LX195T-2	1	RU.СЖАН.30272-04
	LX240T-1	1	RU.СЖАН.30272-05
	LX240T-2	1	RU.СЖАН.30272-06
	LX365T-1	1	RU.СЖАН.30272-07
	LX365T-2	1	RU.СЖАН.30272-08
	SX315T-1	1	RU.СЖАН.30272-09
	SX315T-2	1	RU.СЖАН.30272-10
SVP-719	VX330T-1	1	RU.СЖАН.30657-01
	VX330T-2	1	RU.СЖАН.30657-02
	VX485T-1	1	RU.СЖАН.30657-03
	VX485T-2	1	RU.СЖАН.30657-04
	VX690T-1	1	RU.СЖАН.30657-05
	VX690T-2	1	RU.СЖАН.30657-06
SVP-721	VX330T-1	1	RU.СЖАН.30352-01
	VX330T-2	1	RU.СЖАН.30352-02
	VX485T-1	1	RU.СЖАН.30352-03
	VX485T-2	1	RU.СЖАН.30352-04
	VX690T-1	1	RU.СЖАН.30352-05
	VX690T-2	1	RU.СЖАН.30352-06
SVP-722	VX330T-1	1	RU.СЖАН.30503-01
	VX330T-2	1	RU.СЖАН.30503-02
	VX485T-1	1	RU.СЖАН.30503-03
	VX485T-2	1	RU.СЖАН.30503-04
	VX690T-1	1	RU.СЖАН.30503-05
	VX690T-2	1	RU.СЖАН.30503-06
SVP-723	VX330T-1	1 и 2	RU.СЖАН.30136-01
	VX330T-2	1 и 2	RU.СЖАН.30136-02
	VX485T-1	1 и 2	RU.СЖАН.30136-03
	VX485T-2	1 и 2	RU.СЖАН.30136-04
	VX690T-1	1 и 2	RU.СЖАН.30136-05
	VX690T-2	1 и 2	RU.СЖАН.30136-06
SVP-726	VX690T-1	1 и 2	RU.СЖАН.30000-01
	VX690T-2	1 и 2	RU.СЖАН.30000-02
	VX980T-1	1 и 2	RU.СЖАН.30000-03
	VX980T-2	1 и 2	RU.СЖАН.30000-04
	VX1140T-1	1 и 2	RU.СЖАН.30000-05
	VX1140T-2	1 и 2	RU.СЖАН.30000-06
SVP-733	XCKU060	1	RU.СЖАН.30658-01
	XCKU085	1	RU.СЖАН.30658-02
	XCKU115	1	RU.СЖАН.30658-03

Продолжение таблицы на следующей странице

Продолжение таблицы 9-3

Модули FPGA	FPGA микросхема		Код заказа
	Тип	Порядковый номер	
TS-PCIE	K325T-1	1	RU.СЖАН.30659-01
	K325T-1	1	RU.СЖАН.30659-02
	K410T-1	1	RU.СЖАН.30659-03
	K410T-1	1	RU.СЖАН.30659-04

Список литературы

1. SETFabric-Stream PE для PCI Express устройств типа RootComplex. Руководство пользователя. UG-SFBS-00 (цит. на с. 14).
2. IP-ядро «DM контроллер». Техническое описание. DS-IP-DM-CONT (цит. на с. 15).
3. SETFabric-Stream PE. Библиотека пользователя. Руководство программиста. PG-SFBS-00 (цит. на с. 15).
4. Аппаратная платформа для FPGA модулей Core-0. Руководство пользователя. UG-FPGA-00-CORE-0 (цит. на с. 25).