

Особенности построения эффективного взаимодействия между процессором общего назначения и NeuroMatrix® DSP в составе гибридных СБИС

инж. Андрианов А.В.; инж. Мушкаев С.В.

ЗАО НТЦ “Модуль”

В статье рассматривается задача построения эффективного взаимодействия между процессорным ядром общего назначения, работающим под управлением операционной системы Linux и DSP ядрами с архитектурой NeuroMatrix®, разработки ЗАО “НТЦ Модуль”. Для решения этой задачи используется универсальная библиотека асинхронного удаленного вызова процедур (RPC) libaura. В статье подробно рассматриваются особенности RPC библиотеки libaura, используемая система управления памятью ядра Linux, приводятся оценки производительности данного решения в различных сценариях.

В современных СБИС можно встретить возрастающее количество вспомогательных DSP ядер, которые используются для решения все более сложных вычислительных задач. Это делает актуальным вопрос эффективного управления вспомогательными процессорными ядрами и их интеграцию с существующим программным обеспечением. Как правило, в гибридных СБИС содержится одно или несколько процессорных ядер общего назначения, работающих под управлением операционной системы (чаще всего - Linux) и несколько вспомогательных вычислительных DSP ядер. Примером такой СБИС является K1879XB1Я производства ЗАО НТЦ "Модуль".

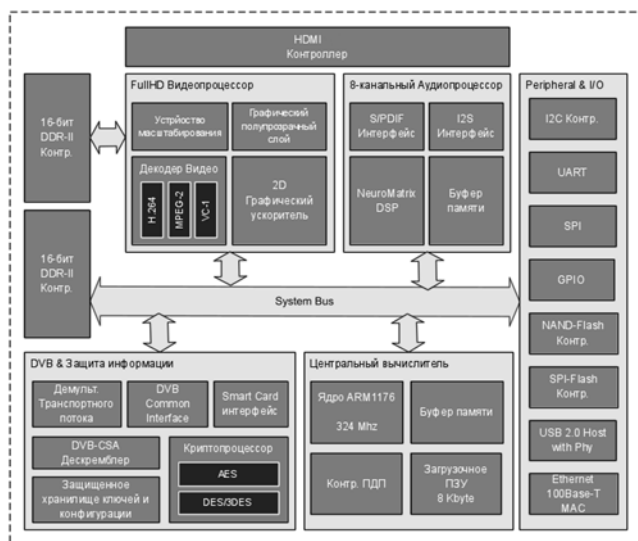


Рис. 1

K1879XB1Я (Рис. 1) содержит одно ядро общего назначения ARM1176JZF-V и одно вычислительное ядро с архитектурой NeuroMatrix®, работающие на частоте 324Mhz. Динамическая память присутствует в виде двух независимых контроллеров внешней DDR-2 памяти и трех блоков накристалльной SRAM памяти.

При построении взаимодействия между процессорным ядром общего назначения, работающим под управлением OS Linux и вычислительными DSP ядрами необходимо учитывать ряд особенностей аппаратуры, оказывающих серьезное влияние на итоговую производительность.

Во-первых, время доступа DSP ядра к разным областям физической памяти может существенно различаться. Время доступа к данным, расположенным в накристалльной SRAM памяти, намного меньше, чем к внешней DDR2 памяти.

Во-вторых, у NeuroMatrix DSP, как и у большинства DSP процессоров, отсутствует блок управления памятью (MMU). Это усложняет или делает невозможным передачу в программу, работающую на DSP виртуального адреса с буфером, выделенным программой в пространстве пользователя через стандартный системный аллокатор памяти.

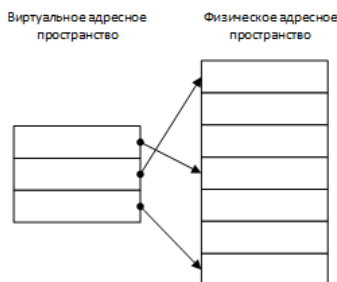


Рис. 2

Страницы памяти буфера, выделенного в пространстве пользователя, могут быть расположены в разных местах физического адресного пространства (см. рис. 2), или могут быть выгружены из

памяти ядром операционной системы (если используется файл/раздел подкачки).

Даже если драйвер ядра зафиксирует страницы буфера в памяти, и передаст адреса страниц программе, исполняемой на DSP ядре, это может существенно снизить производительность алгоритма при переходе от одной странице памяти к другой.

Стандартный аллокатор памяти в пространстве пользователя не позволяет управлять тем, из какой физической области памяти будет произведена аллокация, что может так же снизить производительность.

В ядре OS Linux существуют две стандартных подсистемы, которые способны решить задачу выделения и организации совместного доступа нескольких драйверов/устройств к одному физически непрерывному DMA буферу: ION Memory Manager и DMA-BUF (вместе с аллокатором CMA, Contiguous Memory Allocator).

DMA-BUF является подсистемой ядра Linux, которая позволяет организовать совместный доступ к DMA буферу между несколькими драйверами/устройствами.

ION Memory Manager, активно использующийся в Android, управляет несколькими пулами памяти, некоторые из которых резервируются на стадии загрузки системы, чтобы избежать фрагментации. Некоторые из типов пулов памяти (CMA, DMA) позволяют выделять области физически-непрерывной памяти.

В отличие от DMA-BUF, другой подсистемы ядра Linux, выполняющей схожие функции, ION позволяет выделять память в пространстве пользователя, что делает эту подсистему особенно удобной. ION Memory Manager доступен в ветке ядра Linux 4.4.x LTS для K1879XB1Я.

Для традиционных библиотек удаленного вызова процедур характерны следующие черты, которые при построении сложных приложений могут стать серьезными недостатками:

1. Асимметричность (одна из взаимодействующих сторон является инициатором);
2. Синхронность (выполнение вызывающей процедуры приостанавливается с момента выдачи запроса и возобновляется только после возврата из вызываемой процедуры)

При росте количества вспомогательных ядер, задействованных в приложении, синхронность RPC приводит к тому, что программисту приходится использовать многопоточность, чтобы одновременно задействовать все вспомогательные DSP ядра. Это может серьезно усложнить архитектуру приложения.

В ряде приложений с использованием ядра Neuromatrix®, DSP ядро самостоятельно взаимодействует с периферийными блоками, и может являться инициатором в обмене данными, что делает асимметричность серьезным недостатком, хоть и меньшим, чем синхронность.

AURA (Another Universal RPC, Actually) является универсальной библиотекой RPC, распространяемой под лицензией GNU LGPLv2. Ряд ее особенностей и отсутствие вышеупомянутых недостатков делает ее особенно привлекательной для работы с DSP Neuromatrix®:

- Наличие асинхронного API (Результат выполнения удаленной процедуры можно получить в функции обратного вызова, callback);
- Наличие механизма событий, дающий возможность организовать симметричный обмен данными при использовании асинхронного API;

- Аллокация буфера для передаваемых параметров и возвращаемых значений производится в модуле транспорта. Это позволяет записывать передаваемые параметры сразу в буфер, выделенный через ION Memory Manager;
- Наличие встроенного пула буферов, снижающего количество вызовов аллокатора памяти, которые могут занимать продолжительное время;
- Интеграция с библиотекой асинхронного ввода-вывода libevent;
- Привязки к языку lua, существенно упрощающие разработку хост-части на стадии прототипирования приложения;
- Возможность передачи буферов, выделенных через ION в виде аргументов вызываемой удаленной процедуры

Для тестирования производительности удаленного вызова процедур была проведена серия тестов, проводившихся на микрокомпьютере Module MB77.07, работающим под управлением Debian Jessie.

Первым тестом было измерение среднего времени, которое тратится на удаленный вызов процедуры. Libaiga позволяет вызывать удаленную процедуру как по внутреннему номеру, так и по имени. Вызов по имени немного медленнее, так как влечет за собой поиск в хэш-таблице. Усредненные результаты представлены в Таблице 1.

Имя теста	Вызов удаленной процедуры по номеру (включен пул буферов)	Вызов удаленной процедуры по имени (включен пул буферов)	Вызов удаленной процедуры по номеру (пул буферов выключен)	Вызов удаленной процедуры по имени (пул буферов выключен)
Время, нс	268'562	255'670	1'245'904	1'257'496

Таблица 1

Как видно из таблицы, включение встроенного пула буферов существенно увеличивает производительность RPC. Аллокация непрерывной памяти ядром OS является достаточно дорогой операцией, и, как видно из этих тестов, занимает до 80% времени.

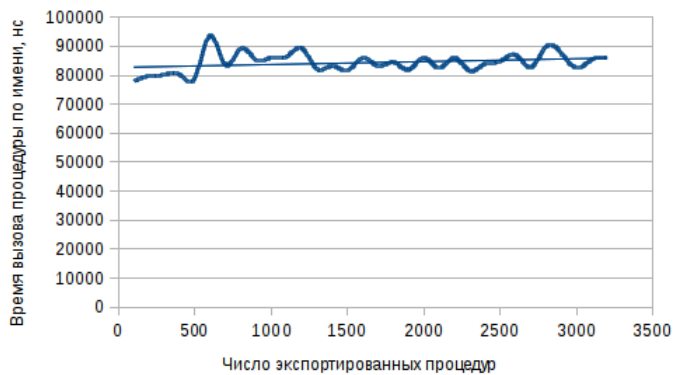


Рис. 3

именем. На графике (Рис. 3) приведена зависимость среднего времени вызова удаленной процедуры от количества экспортированных процедур.

На графике (Рис. 4) приведена зависимость потребления библиотекой оперативной памяти в зависимости от количества экспортированных процедур.

Вторым тестом была оценка производительности используемой в libaiga хэш-таблицы, используемой для поиска имени вызываемой процедуры в таблице экспорта и замер потребляемой памяти. Для этого теста вместо модуля транспорта для Neuromatrix DSP использовался специальный тестовый транспорт-заглушка, экспортировавший различное количество процедур со случайным шестнадцатисимвольным

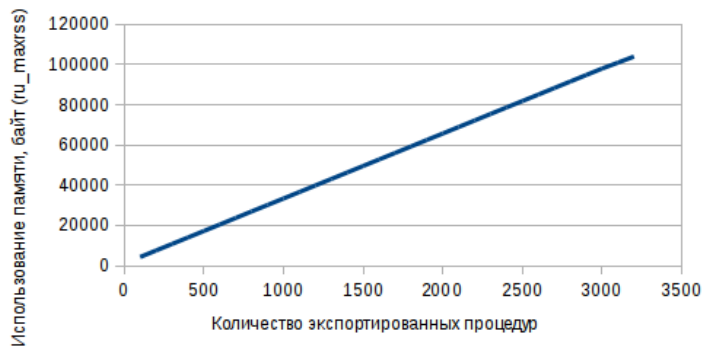


Рис. 4

Как видно из графиков, время вызова процедур практически не растет при значительном увеличении количества экспортированных процедур, в то время как потребление памяти растет линейно.

Последним тестом была оценка производительности RPC библиотеки при передаче в виде параметров процедуры больших объемов данных.

Для этого проводился замер времени выполнения RPC вызова при различных размерах передаваемых параметров.

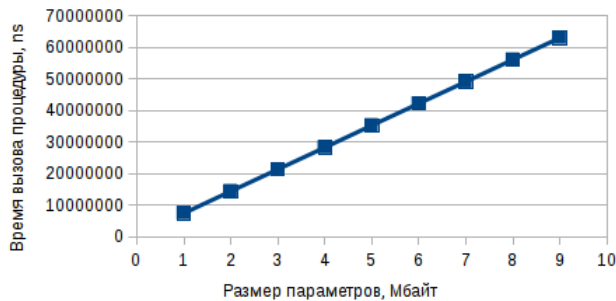


Рис. 5

Для этого теста вызывалась удаленная процедура, принимающая единственный параметр: бинарные данные фиксированного размера и проводился замер среднего времени вызова. Какая либо работа с данными на стороне DSP ядра в этом тесте не проводилась. Как видно из графика (Рис. 5) зависимость линейная, так как данные копируются в буфер один единственный раз при заполнении буфера, выделенного через ION Memory Manager.

Литература

1. Исходные коды libaura; <https://github.com/nekromant/aura>
2. Исходные коды ядра Linux для K1879XB1Я; <https://github.com/RC-MODULE/linux-3.10.x>
3. Шевченко П.А., Презентация доклада «Отечественная СБИС декодера цифрового телевизионного сигнала уже существует» // Тринадцатая международная конференция CSTB'2011, секция бизнес-семинаров, Москва 2011, (рус). <http://www.module.ru/>
4. “Нейропроцессор NM6403. Введение в архитектуру” М.: НТЦ “Модуль”, 1998 <http://www.module.ru/>
5. “Микросхема интегральная NM6406. Руководство по эксплуатации” М.: НТЦ “Модуль”, 2011 <http://www.module.ru/>