Введение (слайд 2)

Эта глава посвящена техническим деталям архитектуры набора команд (Instruction Set Architecture, ISA) RISC-V и ее отличиям от других ISA. В ней рассмотрены некоторые правила и нормы, регулирующие работу RISC-V International, а также то, как разрабатываются расширения и стандарты ISA.

В этой главе вы узнаете:

- о процессе, который используется для разработки RISC-V ISA и расширений;
- о различиях базовой, расширенной и стандартной ISA RISC-V;
- об основах Непривилегированной (Unprivileged) и Привилегированной (Privileged) спецификаций.

Определение архитектуры набора команд (ISA) (слайд 3-4)

Архитектура набора команд (ISA, Instruction Set Architecture) – это абстрактная модель компьютера. Ее также называют архитектурой или архитектурой компьютера. Реализация ISA, например, центральный процессор (CPU, Central Processing Unit), называется имплементацией. Существуют такие распространенные ISA как x86, ARM, MIPS, PowerPC или SPARC. Все эти ISA требуют лицензии для их имплементации. В свою очередь, RISC-V ISA предоставляется под лицензией с открытым исходным кодом, которая не требует платы за использование.

Чем отличается ISA RISC-V

Наиболее заметным отличием RISC-V от других ISA является то, что RISC-V разрабатывается организацией с открытым членством, в которую можно вступить совершенно бесплатно, и лицензировать свои ISA на основе разрешительных лицензий с открытым исходным кодом. Это означает, что любой может внести свой вклад в спецификации, и ни одна компания или группа компаний не может определять направление развития стандартов.

RISC-V International управляется Советом директоров. Совет состоит из членов, избранных для представления всех классов членства, чтобы обеспечить стратегический голос на всех уровнях. Кроме того, Технический руководящий комитет (Technical Steering Committee, TSC) обеспечивает руководство техническими инициативами по разработке долгосрочной стратегии, формированию тактических комитетов и рабочих групп и утверждение технических результатов для ратификации или выпуска спецификаций.

Модель совместного развития

Спецификация RISC-V начинает свою жизнь как только целевая группа, утвержденная Техническим руководящим комитетом (TSC). После того, как целевая группа получила утвержденный устав, она начинает публичную работу на GitHub, работая над своими документами в формате AsciiDoc. Эти репозитории на GitHub могут получать запросы на исправление только от членов RISC-V International, но при этом работа ведется публично и прозрачно. Для групп, решивших вести протоколы, протоколы заседаний целевых групп также публикуются в открытом доступе. Любой пользователь GitHub может свободно отправлять вопросы в репозиторий, чтобы дать раннюю обратную связь по любой спецификации. Спецификации и стандарты, не относящиеся к ISA (например, трассировка процессора, архитектурные тесты, оверлей программного обеспечения), разрабатываются аналогичным образом.

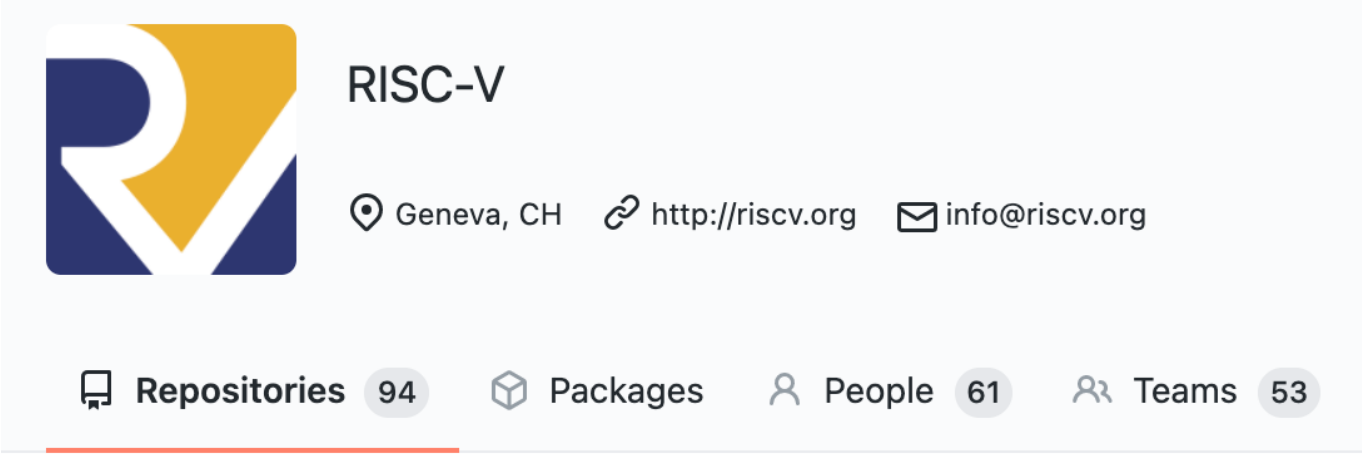

Рисунок 1. RISC-V на GitHub.

Спецификации RISC-V размещены на GitHub и находятся рядом с десятками программных проектов. Список ратифицированных спецификаций и ссылки на их репозитории можно найти на GitHub.

Создание и курирование открытых спецификаций

Процесс разработки спецификаций обычно возглавляет архитектор аппаратного обеспечения в одной из организаций-членов RISC-V International. Архитектор может непосредственно не разрабатывать текст, но выступает в качестве председателя целевой группы, контролирующей разработку спецификации. Для завершения разработки спецификации группе может потребоваться от нескольких месяцев до более одного года. Жизненный цикл расширения будет описан далее.

Открытость процесса разработки зависит от трех ключевых фактов:

1. Список рассылки целевой группы является общедоступным.
2. Документ спецификации находится в открытом доступе, и к нему можно оставлять комментарии.
3. Существует общедоступный список рассылки, куда любой желающий может отправить электронное письмо (isa-dev@groups.riscv.org).

Используя эту методологию, даже те, кто не являются членами организации, могут участвовать в разработке любой спецификации или стандарта, задавая вопросы, внося предложения или просто следя за ходом работы. Более того, в процессе ратификации существует 45-дневный период, когда все работы над спецификацией должны быть заморожены, а сама спецификация опубликована для ознакомления сообществом разработчиков. В это время все желающие могут высказать свои замечания, и все вопросы будут решены до голосования за ратификацию.

Хотя стать членом RISC-V International – самый простой способ внести свой вклад в открытые спецификации, это не единственный способ. Любой может внести свой вклад, взаимодействуя с целевыми группами на публичных форумах, таких как список рассылки и GitHub.

Жизненный цикл расширений RISC-V (слайд 5-6)

Каждое расширение RISC-V проходит несколько этапов на пути к ратификации (рис. 2). В этом разделе кратко рассмотрен каждый этап этого пути:

1. **План.**

   Целевая группа разрабатывает окончательный устав и устанавливает некоторые временные рамки.
2. **Разработка.**

   Группа выпускает несколько версий, которые считаются нестабильными.
3. **Заморозка.**

   Группа готовит полный окончательный вариант спецификации без существенных неизвестных и ожидаемых изменений (только для исправления проблем, но без новых возможностей).
4. **Готовность к утверждению.**

   Проект спецификации рассылается на общественное рассмотрение, рассматриваются любые комментарии или вопросы общественности, и Технический Руководящий Комитет (TSC) ставится в известность о необходимости голосования.
5. **Развитие экосистемы.**

Расширение ратифицировано и поддерживается как часть RISC-V ISA. Сообществу еще предстоит решить множество задач. Например, спецификация Vector была ратифицирована, но добавление автовекторизации в компиляторы является частью списка задач по развитию экосистемы этой спецификации.

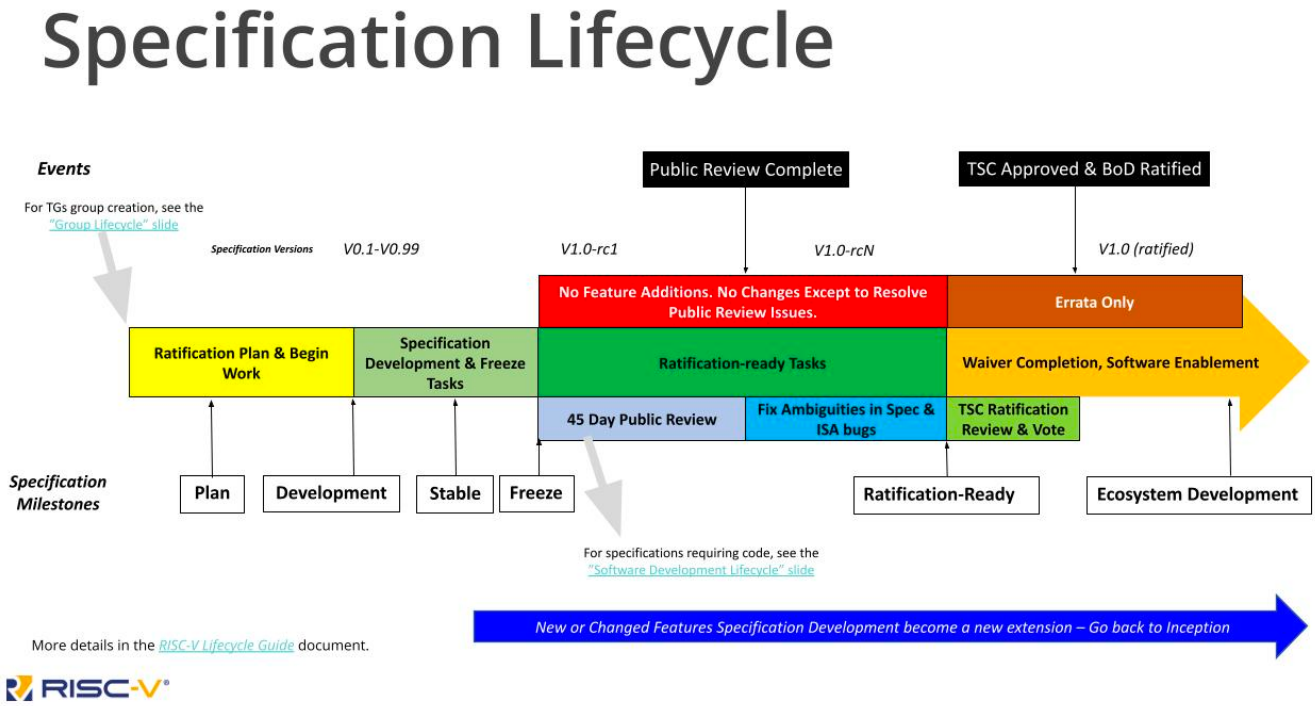

Рисунок 2. Жизненный цикл спецификации.

После ратификации расширения оно добавляется либо в непривилегированную, либо в привилегированную спецификацию. Иногда спецификация создается как часть отдельного документа (наиболее распространенным примером является спецификация отладки). Но это редкий случай и обычно указывает на то, что расширение не является частью ISA, а представляет собой «стандарт» или «не-ISA спецификацию». Далее рассмотрены непривилегированная и привилегированная спецификации более подробно.

Непривилегированная спецификация (слайд 7-9)

Организация спецификаций (слайд 7-8)

RISC-V ISA состоит из двух частей:

- часть 1, непривилегированная спецификация;
- часть 2, привилегированная спецификация.

Чтобы понять, почему спецификация делится на две разные части, необходимо сначала немного разобраться в компьютерной архитектуре и безопасности. Исторически процессоры использовали иерархические домены защиты, часто называемые кольцами защиты, для защиты данных и кода от злоумышленников.

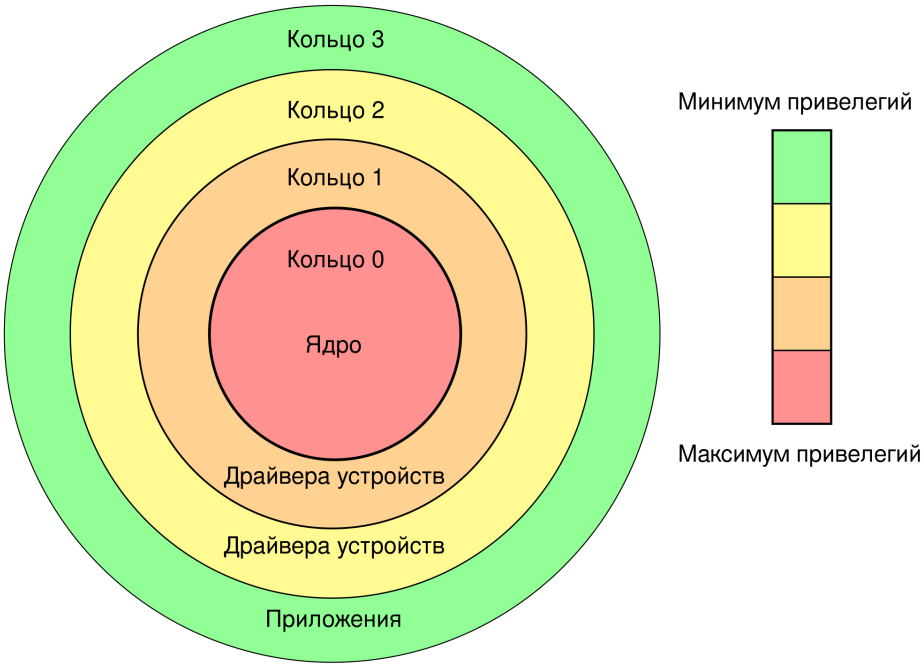

Рисунок 3. Кольца привилегий.

Наиболее привилегированный код работает в «кольце 0» и имеет доступ ко всей системе. Процессор решает, какие привилегии предоставить исполняемому коду, в зависимости от уровня привилегий. Например, доступ к памяти по физическому адресу может быть ограничен «кольцом 0», так что другие кольца должны ссылаться на виртуальное адресное пространство. Обычно процессор может одновременно работать только в одном из режимов привилегий, а для перехода между режимами существуют специальные команды. Все эти детали могут меняться от системы к системе, но они должны соответствовать правилам, установленным в документах спецификации конкретной архитектуры.

В настоящее время RISC-V имеет три уровня привилегий: режим пользователя (U-mode, User Mode), режим супервизора (S-mode, Supervisor Mode) и машинный режим (M-mode, Machine Mode). Их можно представить как «кольцо 2», «кольцо 1» и «кольцо 0» соответственно. Другие режимы, такие как режим гипервизора (H-mode), вероятно, будут добавлены в ближайшем будущем. Как и на рисунке выше, режим U предназначен для пользовательских процессов, режим S – для ядра и/или драйверов устройств, а режим M – для загрузчика и/или микропрограммного обеспечения. Каждый уровень привилегий имеет доступ к определенным регистрам управления и состояния (Control and Status Registers, CSR), и более высокие уровни привилегий могут получить доступ к CSR менее привилегированных уровней.

Внутри непривилегированной спецификации (слайд 9)

Непривилегированная спецификация описывает элементы, не относящиеся к машинному режиму (M-mode) или режиму супервизора (S-mode). В нее входят базовая ISA, а также ее расширения, такие наборы команд как: целочисленные (I, integer), с плавающей точкой (F, float), двойной точности (D, double), упакованные (C, compressed) и многие другие.

Базовые наборы команд описывают формат команд, основные целочисленные команды, команды загрузки и хранения, а также другие фундаментальные детали ISA. Они бывают на нескольких видов:

- RV32I – целочисленный, 32 бита;
- RV32E – сокращенный RV32I для встраиваемых систем;
- RV64I – целочисленный, 64 бита;
- RV128I – целочисленный, 128 бит.

Все эти "базовые ISA" либо уменьшают, либо расширяют базовый набор команд RV32I. Например, RV64I расширяет целочисленные регистры и поддерживаемое адресное пространство пользователя до 64 бит. Это означает, что команды LOAD и STORE работают немного иначе, чем в RV321. Непривилегированная спецификация содержит раздел, объясняющий эти различия.

Расширения базового ISA

Непривилегированная спецификация также содержит описания расширений этих базовых ISA. Любое расширение, которое не требует M-режима для работы, может быть описано в непривилегированной спецификации.

Каждое расширение базового ISA разрабатывается и поддерживается целевой группой:

- Crypto Task Group работает над криптографическими расширениями, которые могут перенести многие сложные криптографические алгоритмы в аппаратное обеспечение, повышая надежность и скорость его работы;
- Extension Task Group работает над расширениями для работы с битами, которые могут ускорить выполнение многих распространенных математических задач;
- Vector (V) Extension Task Group работает над векторными командами, которые лежат в основе многих вычислений при обработке графики.

После ратификации эти расширения добавляются в непривилегированную спецификацию. Ниже перечислены некоторые из ратифицированных расширений, которые можно встретить в процессоре RISC-V.

Расширение M

Глава 7 непривилегированной спецификации описывает, как должно выполняться умножение и деление целых чисел. В ней описывается поведение каждой из команд умножения (MUL, MULH, MULHU, MULHU, MULW), какие регистры используются для множителя и множимого, и где хранится результат. То же самое делается и для деления, поскольку функционально можно рассматривать деление как обратную операцию умножению. Может показаться странным, что это расширение не вошло в стандартную спецификацию. Но для многих встроенных процессоров умножение можно выполнять программно, если оно требуется нечасто или вообще не требуется. Удаление этой логики из процессора позволяет сэкономить деньги на разработке и снизить стоимость конечного продукта.

Расширение F

В главе 11 непривилегированной спецификации описано, как добавляются вычислительные команды с плавающей точкой одинарной точности, соответствующие стандарту арифметики IEEE 754-2008. Существует множество ресурсов, описывающих детали арифметики с плавающей точкой в вычислениях. В этой главе описывается, как этот процесс реализован в RISC-V, и дополняется главой 12 (расширение D), которая описывает вычислительные команды двойной точности с плавающей точкой. А в главе 13 рассматривается расширение стандарта Q для 128-битных двоичных команд с плавающей точкой четверной точности. Все эти три расширения соответствуют

стандартам IEEE. Многие встроенные приложения не требуют логики с плавающей точкой, и поэтому это расширение не является частью базовых ISA.

Расширение C

В главе 16 описано расширение сжатого набора команд, которое уменьшает статический и динамический размер кода путем добавления коротких 16-битных кодировок команд для общих операций. Как правило, 50%-60% команд RISC-V в программе могут быть заменены командами RVC, что приводит к уменьшению размера кода на 25%-30%. Расширение C совместимо со всеми другими стандартными расширениями команд. Оно позволяет свободно смешивать 16-битные команды с 32-битными, причем последние могут начинаться с любой 16-битной границы. Таким образом, при добавлении расширения C в любую систему ни одна команда не вызывает исключения, связанные с выравниванием адресов команд.

Приведенные выше расширения охватывают большинство ратифицированных в настоящее время расширений в непривилегированной спецификации. Важно отметить, что многие расширения включены в спецификацию на стадии "черновика" или "заморозки". Как уже было сказано в разделе «Жизненный цикл расширений RISC-V», эти спецификации еще не ратифицированы, и любая реализация должна избегать их использования в производстве.

Привилегированная спецификация (слайд 10)

Обзор (слайд 10)

Как следует из названия, привилегированная спецификация содержит описания RISC-V ISA, которые работают в машинном режиме (M-mode) или режиме супервизора (S-mode). Эти режимы имеют повышенные привилегии и поэтому описаны в совершенно отдельном документе от базового ISA и стандартных расширений. Эта спецификация также содержит дополнительные функциональные возможности, необходимые для запуска полнофункциональных операционных систем, таких как Linux.

В первой части каждой главы привилегированной спецификации привилегий подробно описаны регистры управления и состояния (CSR), доступ к которым возможен только из M- и S-режима. Мы не будем рассматривать эти детали здесь, а сосредоточимся на других деталях, специфичных для этих двух режимов.

ISA машинного уровня (M-режим), версия 1.11

В этой главе описаны функции машинного уровня, доступные в машинном режиме. M-режим используется для низкоуровневого доступа к аппаратной платформе и является первым режимом, который включается при сбросе, когда процессор завершает инициализацию и готов к выполнению кода. M-режим также может использоваться для реализации функций, которые слишком сложно или дорого реализовывать в аппаратном обеспечении напрямую. Хорошим примером может служить сторожевой таймер, реализованный в низкоуровневом программном обеспечении (микропрограмме), который помогает системе восстанавливаться после сбоев. Далее рассмотрены три важные особенности M-режима, описанные в спецификации: немаскируемые прерывания, атрибуты физической памяти и защита физической памяти.

Немаскируемые прерывания (Non-Maskable Interrupts, NMI)

Немаскируемые прерывания (NMI) используются только в условиях аппаратных ошибок. При срабатывании они вызывают немедленный переход к обработчику NMI, работающему в M-режиме, независимо от того, как установлен бит разрешения прерывания для этого аппаратного потока. Другими словами, это прерывание будет обслуживаться без возможности его заблокировать. Каждое NMI будет иметь связанный с ним регистр «mcause». Это позволяет разработчикам

самостоятельно решать, как они хотят обрабатывать эти прерывания, и позволяет им определять множество возможных причин. NMI не сбрасывают состояние процессора, что позволяет диагностировать, сообщать и, возможно, локализовать аппаратную ошибку.

Атрибуты физической памяти (Physical Memory Attributes, PMA) (слайд 12)

Физическая карта памяти системы включает в себя такие диапазоны адресов, как: области памяти, управляющие регистры с привязкой к памяти и пустые области в адресном пространстве. Некоторые области памяти могут не поддерживать чтение, запись или выполнение; некоторые могут не поддерживать доступ к подслову или подблоку; некоторые могут не поддерживать атомарные операции; а некоторые могут не поддерживать когерентность кэша или иметь различные модели памяти. В системах RISC-V эти свойства и возможности каждого региона физического адресного пространства называются атрибутами физической памяти (PMA).

PMA некоторых областей памяти фиксируются во время проектирования микросхемы – например, для ПЗУ на кристалле. Другие фиксируются во время проектирования платы, в зависимости, например, от того, какие другие микросхемы подключены к внешним шинам. Некоторые устройства могут быть конфигурируемыми во время работы для поддержки различных применений, которые подразумевают различные PMA. Например, оперативная память на кристалле может быть приватным кэшем одного ядра в одном приложении, или использоваться как общий некэшируемый блок памяти в другом конечном приложении. В большинстве систем потребуется, чтобы по крайней мере некоторые PMA динамически проверялись аппаратно на более поздних этапах конвейера выполнения после того, как будет известен физический адрес, поскольку некоторые операции не будут поддерживаться по всем физическим адресам памяти, а некоторые операции требуют знания текущей настройки конфигурируемого атрибута PMA.

Для RISC-V выделяется спецификация и проверка PMA в отдельную аппаратную структуру – устройство контроля PMA (PMA checker). Во многих случаях атрибуты известны на этапе проектирования системы для каждой области физического адреса и могут быть непосредственно подключены к устройству контроля PMA. Если атрибуты настраиваются во время выполнения, то могут быть предусмотрены специфические для платформы регистры управления с привязкой к памяти для задания этих атрибутов с гранулярностью, соответствующей каждой области памяти (например, для статической памяти с произвольным доступом (SRAM) на кристалле, которая может быть гибко разделена между кэшируемым и некэшируемым использованием).

Подробное описание PMA может занять целую главу изложения. Здесь не рассматриваются PMA с упорядочиванием памяти, PMA с идемпотентностью, PMA с когерентностью или PMA с возможностью кэширования. Подробности реализации PMA описаны в разделе 3.5 Привилегированной спецификации.

Защита физической памяти (PMP) (слайд 13-15)

Общей особенностью большинства современных процессоров является способ выполнения безопасных удаленных вычислений или «надежная среда выполнения». Примеры этой технологии включают Intel Software Guard Extensions (SGX), AMD Secure Encrypted Virtualization (SEV) и Arm TrustZone. Хотя RISC-V ISA не предоставляет комплексного решения для реализации надежной среды выполнения, возможности защиты физической памяти (PMP) являются прочной основой, на которой можно построить такую систему.

RISC-V PMP ограничивает физические адреса, доступные программному обеспечению, работающему на жестком диске (аппаратный поток). Дополнительный модуль PMP предоставляет регистры управления машинным режимом для каждого модуля, позволяющие задавать привилегии доступа к физической памяти (чтение, запись, выполнение) для каждой области физической памяти. Значения PMP проверяются параллельно с проверками PMA, которые были рассмотрены в предыдущем разделе. Степень детализации настроек управления доступом PMP зависит от платформы и внутри платформы может варьироваться в зависимости от области физической памяти, но стандартное кодирование PMP поддерживает области размером до четырех байтов. Привилегии определенных регионов могут быть жестко закреплены (например, некоторые регионы могут быть видны только в машинном режиме, но не на уровнях с более низкими привилегиями).

Записи PMP описываются 8-битным регистром конфигурации и одним 32-битным (или 64-битным) адресным регистром. Поддерживается до 16 записей PMP. Если реализованы какие-либо записи PMP, то должны быть реализованы все CSR PMP, но любые поля CSR PMP могут быть установлены в ноль. CSR PMP доступны только в M-режиме.

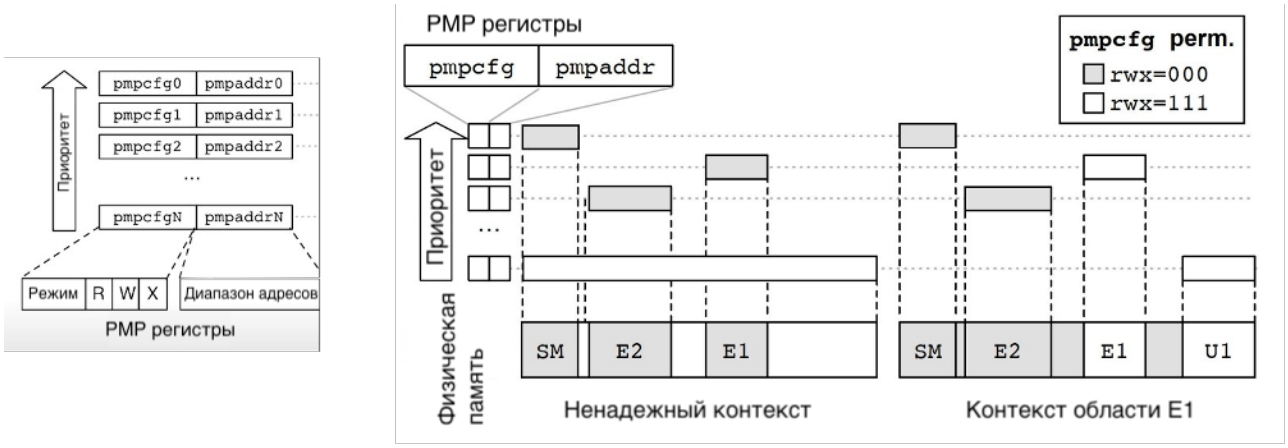

Рисунок 4. Пример настройки разных контекстов.

На рисунке 4 показан пример того, как можно настроить два разных контекста: один ненадежный и один с доступом к «Области E1». В этом примере приложение запускается в пользовательском контексте U1. Это приложение имеет доступ только к своей собственной памяти и памяти внутри Области E1. Память внутри Области E2 и в мониторе безопасности (Security Monitor, SM) недоступна пользовательскому приложению. Таким образом, конфиденциальность данных обеспечивается за счет того, что монитор безопасности (работающий в M-режиме) может изменять настройки PMP, разрешая или запрещая доступ к областям памяти в соответствии с конфигурацией PMP.

ISA уровня супервизора (S-Mode) ISA, Version 1.11 (слайд 16)

В этой главе описывается архитектура уровня супервизора RISC-V, которая содержит общее ядро, используемое с различными схемами трансляции и защиты адресов уровня супервизора. Режим супервизора для поддержки чистой виртуализации намеренно ограничен с точки зрения взаимодействия с базовым физическим оборудованием, таким как физическая память и прерывания устройств. В соответствии с этим, некоторые возможности супервизора, включая запросы на прерывания таймера и межпроцессорные прерывания, предоставляются механизмами, специфичными для конкретной реализации. В некоторых системах исполнительная среда супервизора (Supervisor Execution Environment, SEE) предоставляет эти возможности в виде бинарного интерфейса супервизора (Supervisor Binary Interface, SBI). В других системах эти возможности предоставляются напрямую, через какой-либо другой механизм, определяемый реализацией.

RISC-V поддерживает страничную 32-битную, 39-битную и 48-битную адресацию виртуальной памяти на основе страниц. Для синхронизации обновлений структур данных управления памятью в

памяти с текущим выполнением используется супервизорная команда доступа к памяти (SFENCE.VMA). Выполнение этой команды гарантирует, что все предыдущие сохранения, уже видимые текущему RISC-V аппаратному потоку (hardware thread), будут упорядочены перед всеми последующими неявными обращениями этого аппаратного потока к структурам данных управления памятью.

Виртуальная память — это довольно сложная концепция, и она выходит за рамки этого курса. Достаточно понимать, что RISC-V поддерживает виртуальную страничную память с несколькими размерами, и что существует специальная команда S-режима, используемая для синхронизации обновлений между аппаратными потоками.

Спецификации, не входящие в ISA

Целевые группы также могут работать над программным обеспечением или стандартами, которые не являются частью ISA. Например, следующие группы работают над проектами, которые не приводят к созданию спецификаций, а скорее являются стандартами, стимулирующими сообщество разрабатывать свои продукты на общей основе:

- рабочая группа по отладке работает над поддержкой и стандартами внешней отладки;
- целевая группа по соблюдению требований работает над тестами и фреймворками соответствия RISC-V ISA;
- целевая группа по структуре конфигурации работает над тем, как представить структуру конфигурации данной аппаратной реализации как в удобном для чтения человеку формате, так и в двоичном формате.