**SCR1**

После более или менее успешных экспериментов с разработкой собственных вариантов микроархитектур RISC-V полезно посмотреть на «профессиональные» софт-ядра. Для рассмотрения было выбрано достаточно мощное, судя по описанию, и отлаженное ядро – SCR1 от компании Syntacore.

SCR1 представляет собой IP-ядро 32-х разрядного процессора архитектуры RISC-V микроконтроллерного класса, предназначенное для использования в составе различных встраиваемых и других систем.

Плюсы:

- SCR1 -это открытое RV32I|E[MC] ядро высокого качества от компании Syntacore, очень приятным бонусом является то, что разрешено неограниченное коммерческое использование (достаточно редко встречается в случае ядер разрабатываемых и поддерживаемых коммерческими компаниями).

Минусы:

- SCR1 – ядро не любительского уровня и требует некоторых временных затрат на знакомство с его «экосистемой».

Ядро написано на SystemVerilog и предоставляется со всем необходимым пакетом документации и верификационным окружением. Поддерживается и обновляется по мере развития спецификаций RISC-V. Хорошо подходит для образовательных целей и используется в промышленности.

**Поддерживаемые наборы команд:**

RV32I - базовая целочисленная система команд

- минимальный необходимый набор команд;

- 32-битное адресное пространство;

- 31 регистр общего назначения по 32-бит;

- мсе инструкции ширины 32-бит.

RV32E (E=Embedded) базовая целочисленная система команд

- аналогично RV32I, но представлено только 15 регистров общего назначения;

-способствует реализациям, чувствительным к стоимости регистрового файла.

Расширение“M”

- стандартное опциональное расширение для операций целочисленного умножения и деления;

Расширение“C”

- стандартное опциональное расширение для сокращенных инструкций;

- дополнительный набор инструкций ширины 16-бит;

Ядро SCR1 SCR1 работает только в режиме «Machine Privilege Level» (Machine Level -режим с наивысшими привилегиями, любой исполняемый код воспринимается как надежный(profile -embedded without protection)) - можно сказать – типовой/родной режим для микроконтроллеров. Практически всегда программное обеспечение микроконтроллера работает или в режиме bare-metal – без какой-либо операционной системы, или под управлением небольшой ОС, часто – операционной системы реального времени.

Моменты, не сильно явно прописанные стандартами (да и не сильно важные в случае ядер именно микроконтроллерного класса).

Особые инструкции:

▪ECALL: генерирует исключительную ситуацию environment\_call

▪EBREAK: генерирует исключительную ситуацию breakpoint

▪ERET: возврат из trap

▪WFI: остановка ядра до возникновения прерывания

Control and Status Registers (CSRs) -набор специализированных регистров для просмотра состояния и управления ядром и подсистемами:

▪информационные регистры (только для чтения);

▪настройка и обработка аппаратного отладчика (Trap);

▪счетчики/Таймеры (отображаются в память);

▪регистры Debug;

▪регистры IPIC.

Структурная схема ядра представлена на рис. 1.

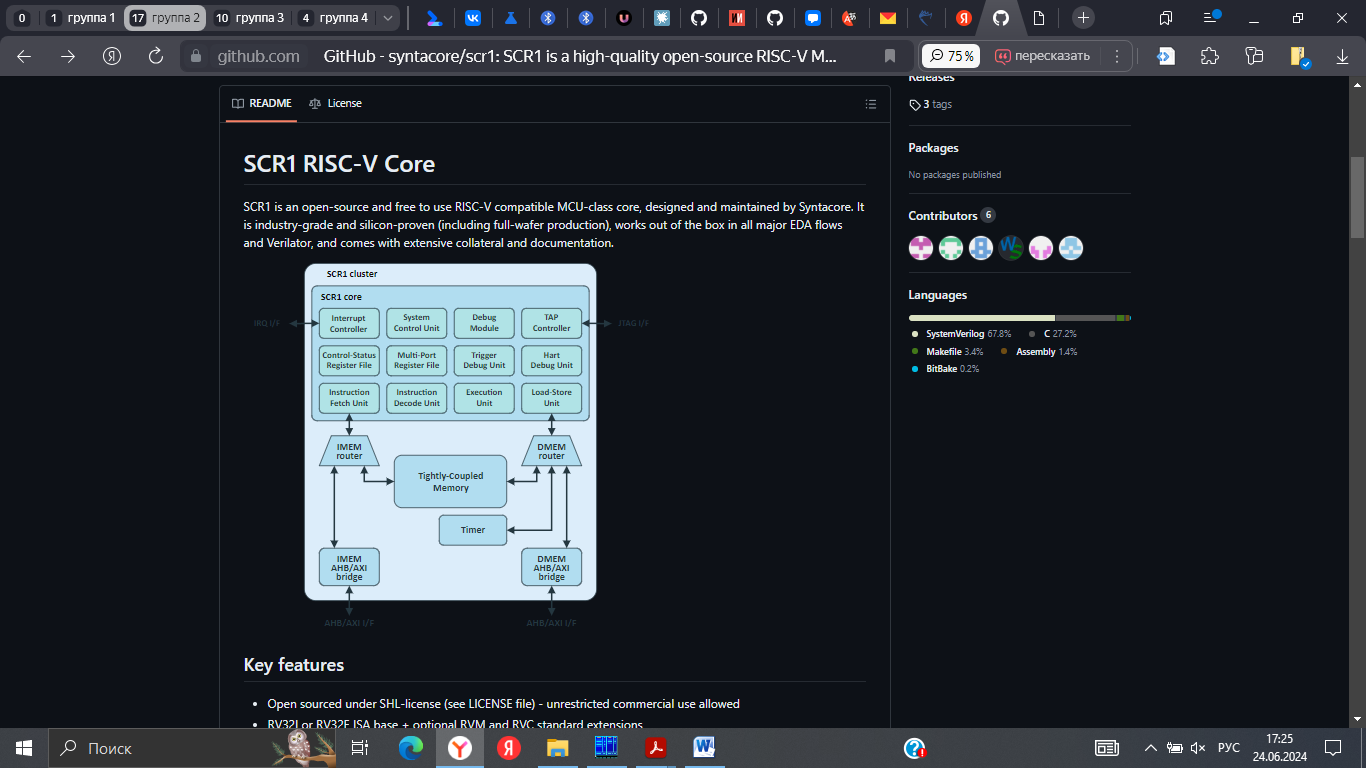


Рис.1 Структурная схема процессорного ядра SCR1.

Как можно видеть, значительная часть модулей ядра предназначена для работы в режиме отладки приложений. Это, можно сказать, ставит ядро в разряд IP-ядер, предназначенных для достаточно серьезных задач и проектов. И это весьма чувствительно отличает данное ядро от многих более «легких» ядер.

Взаимодействие с «внешним миром» у SCR1 организовано:

- через отладочный интерфейс JTAG;

- линии прерываний;

- шинный интерфейс AXI или AHB (в зависимости от предпочтений, или требований проекта разработчик может выбирать между двумя вариантами top-файлов ядер).

SCR1 работает в модели памяти little-endian, 32-битная адресация, выравнивание на границы 32-битных слов при работе с наборами инструкций RV32I|E[M], и на границу 16-битных слов для RV32C.

Если программный код не большой (менее 64КБ), то в принципе, он может выполняться очень быстро. Достигается это за счет, так называемой, «тесносвязанной памяти» (Tightly-Coupled Memory), которую можно рассматривать, как своеобразную кэш-память.

Тесно связанная память или Tightly-Coupled Memory (TCM) -небольшая близкая к ядру память с быстрым откликом:

- гарантированное время отклика -1 такт;

- может использоваться для секций инструкций и данных;

- два независимых интерфейса доступа к инструкциям и данным;

- размер TCM до 64 kBytes;

- базовый адрес 0x00480000.

Карта памяти ядра SCR1 представлена на рис.2.



Рис.2 Карта памяти SCR1

Внешний интерфейс памяти- AXI4 Interface или AHB-Lite Interface.

Ядро поставляется в виде исходных текстов на Verilog (System Verilog), доступно для прямого скачивания, или через Git. Структура репозитория представлена в таблице. Помимо самих исходников в репозитории представлено верификационное окружение и документация.

Таблица 1

Структура репозитория ядра SCR1

| **Директория** | **Описание** |
| --- | --- |
| ├─ riscv-tests | Исходные файлы тестов на соответствие RISC-V ISA |
| ├─ riscv-compliance | Общие исходные файлы для тестов на соответствие RISC-V |
| └─ coremark | Исходные файлы тестов производительности EEMBC's CoreMark® benchmark |
| **Docs** | **Документация SCR1** |
| ├─ scr1\_eas.pdf | Спецификация внешней архитектуры SCR1 |
| └─ scr1\_um.pdf | Руководство пользователя SCR1 |
| **Sim** | **Тесты и скрипты для симуляции** |
| ├─ tests/common | Общие исходные файлы для тестов |
| ├─ tests/riscv\_isa | RISC-V ISA тестирует исходные файлы для конкретной платформы |
| ├─ tests/riscv\_compliance | Исходные файлы тестов на соответствие стандарту RISC-V |
| ├─ tests/benchmarks/dhrystone21 | Исходные тексты тестов Dhrystone 2.1 |
| ├─ tests/benchmarks/coremark | Исходные тексты тестов EEMBC's CoreMark® |
| ├─ tests/isr\_sample | Пример программы-с обработчиком прерываний "Interrupt Service Routine" |
| ├─ tests/hello | Пример простой программы - "Hello" |
| └─ verilator\_wrap | Оболочки для симулятора Verilator |
| **Src** | **SCR1 RTL исходные тексты и тестбенчи** |
| ├─ includes | Заголовочные файлы |
| ├─ core | Исходные тексты компонентов ядра |
| ├─ top | Исходные файлы модулей верхнего уровня процессорного ядра |
| └─ tb | Тестбенчи |

Требуемые ресурсы – порядка 15 kGates в базовой конфигурации RV32EC

Возможности по конфигурации:

- Конфигурируемый конвейер от 2 до 4 стадий;

- Конфигурируемый IPIC на 16 линий прерываний;

- Конфигурируемый RISC-V Debug модуль с JTAG интерфейсом.

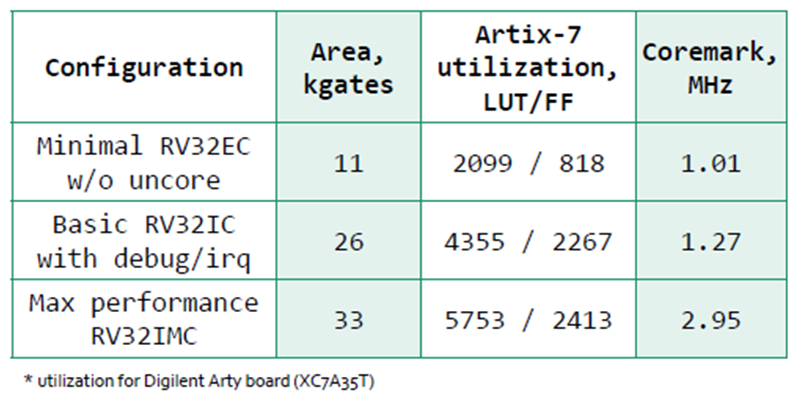


Рис. 3 Результаты тестирования ядра SRC1 на плате Digilent Arty XC7A35T.

Интегрированный Программируемый Контроллер Прерываний (Integrated Programmable Interrupt Controller -IPIC) может быть опционально включен в состав проекта SCR1

- доступно до 16 векторов/линий прерываний;

- вектора прерываний имеют фиксированные приоритеты;

- вектор с наименьшим номером имеет наивысший приоритет;

- IPIC настраивается через специальные регистры IPIC CSR.

Подсистема Debug может быть опционально включена в состав проекта SCR1

- совместима с RISC-V External Debug Support Spec Ver. 0.13.2

- поддержка стандартных функций отладки:

- Reset;

- Halt / Resume / Step;

- Доступ к GPR, CSR, MEM;

- HW Breakpoint / Watchpoint с выбором количества;

- интерфейс JTAG IEEE Std 1149.1-2013.

SCR1 имеет суммарно 33 конфигурируемые опции:

1. Архитектура системы команд:

- RV32I или RV32E база;

- расширения M и C.

1. Микроархитектурные опции ядра(оптимизация площади/производительности)

- стадии конвейера;

- конфигурация умножителя;

- тактирование;

1. Адресные константы:

- адрес вектора сброса - RST vector;

-MTVEC base;

- адрес обработчика таймера – Timer;

- TCM;

1. Опции вне ядра:

- включение подсистемы Debug;

- включение HW Breakpoints/Watchpoints + выбор количества;

- включение IPIC;

- включение TCM;

- добавление регистров на входах/выходах AXI/AHB;

1. Симуляционные опции:

- включение SVA и Tracelog.

**Литература и ссылки**

<https://github.com/syntacore/scr1>

<https://syntacore.ru/page/products/processor-ip/scr1x32ru>

<https://riscv.expert/media/riscv_moscow_2019/SCR1%20Tutorial%20Symposium%20Moscow.pdf>