

АНАЛИЗ ПРИМЕНЕНИЯ ПРОГРАММНЫХ ЭМУЛЯТОРОВ В МАРШРУТЕ ПРОЕКТИРОВАНИЯ СнК

Кишко Д. В., Алехин А. П.
ООО Ядро Микропроцессоры
Москва, Россия
E-mail: d.kishko@yadro.com

В работе проведен сравнительный анализ программных эмуляторов для систем на кристалле (СнК). Рассмотрены способы интеграции эмуляторов с основными инструментами разработки в маршруте проектирования и верификации СнК. Показано, что представленный подход позволяет снизить время на разработку и отладку функциональных сценариев за счет включения в работу на ранних стадиях проектирования и переноса наиболее затратных процедур отладки в эмулятор.

ВВЕДЕНИЕ

Для обеспечения высокого качества проектирования СнК требуются значительные усилия в области функциональной верификации дизайна интегральной схемы. Процесс верификации позволяет определить соответствие разрабатываемого дизайна предъявляемым требованиям, и выявить на ранних этапах скрытые проблемы функционирования как отдельных частей СнК, так и в целом интегральной схемы. Исследования последних лет показывают, что более 50% времени дизайн-компании затрачивают на процессы верификации СнК [1]. Существуют разные подходы к функциональной верификации СнК, отличающиеся сложностью, стоимостью и глубиной осуществляемых проверок. Согласно исследованиям, представленным в [2] и [3], к методам функциональной верификации относят: формальную верификацию, поведенческую, потактовую и событийную симуляцию (HDL симуляцию), использование аппаратных ускорителей и прототипирования на ПЛИС. Сам процесс верификации включает не только планирование, разработку и отладку функциональных сценариев, но также запуск этих сценариев на выполнение, который может занимать значительное время.

Программные эмуляторы СнК позволяют значительно снизить общее время отладки, запуска и выполнения функциональных сценариев, в связи с этим представляют значительный интерес для внедрения в маршрут проектирования СнК.

I. МАРШРУТ ПРОЕКТИРОВАНИЯ СнК

Обобщенный маршрут проектирования СнК представлен на рисунке 1. Как правило, он включает в себя следующие этапы: исследование рынка и формирование продуктовых требований к СнК, архитектурную проработку, этапы логического и физического дизайна, выпуск СнК, процедуры отбраковки и приемки изготовленной интегральной схемы. Основные задачи функциональной верификации СнК начинаются на этапе проработки архитектуры и заканчиваются началом этапа проработки физического дизайна.

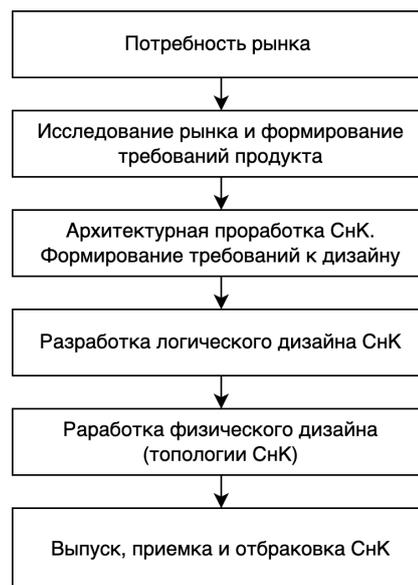


Рис. 1 – Обобщенный маршрут проектирования СнК

Как правило разработка и верификация самой программной модели СнК происходит значительно быстрее, чем разработка логического дизайна СнК. При этом использование программных эмуляторов позволяет сократить время на функциональную верификацию СнК ввиду того, что отладка функциональных сценариев на эмуляторе занимает гораздо меньше времени, чем отладка с использованием HDL симулятора. Это достигается в первую очередь за счет применения высокоуровневых языков программирования, использования готовых библиотечных моделей составных частей СнК. Также большинство программных эмуляторов системного уровня позволяют осуществить запуск целевой операционной системы и произвести валидацию пользовательских сценариев, что является дополнительным преимуществом в повышении качества проектируемого дизайна. На рисунке 2 представлена взаимосвязь процессов разработки логического дизайна, функциональных сценариев и программной модели.

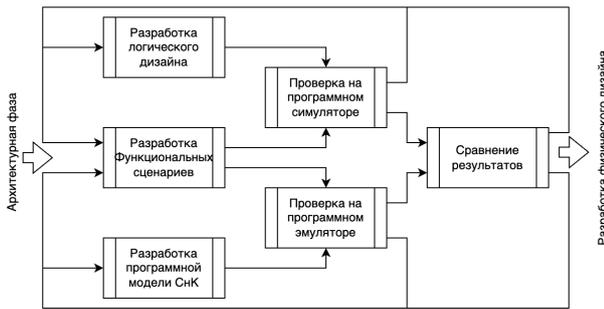


Рис. 2 – Процессы разработки и функциональной верификации СнК

Создание программной модели разрабатываемого СнК, позволяет решить ряд задач, основными из которых являются:

- Разработать функциональные сценарии использования СнК, необходимые для верификации логического дизайна;
- Верифицировать функциональные сценарии;
- Выработать требования к программной модели функциональных блоков СнК;
- Сформировать артефакты (отладочный вывод, трассы выполнения CPU) для сравнения с результатами работы симулятора.

II. АНАЛИЗ СОВРЕМЕННЫХ ЭМУЛЯТОРОВ

В начале разработки логического дизайна СнК его программная модель может иметь достаточно ограниченный функционал, позволяющий проводить минимальные функциональные проверки. Однако в процессе проектирования СнК и доработки его модели, перечень проверок расширяется, обеспечивая тем самым полноту проводимых проверок. Это позволяет подойти к концу этапа логического дизайна с практически полным функционально верифицированным дизайном СнК. Неоспоримым преимуществом эмуляторов является возможность их совместного использования вместе с HDL симуляторами с помощью DPI, а также интеграция с аппаратными ускорителями на базе FPGA [3].

В зависимости от сложности проектируемого СнК к эмуляторам могут предъявляться различные требования, основные из которых представлены ниже:

- T1. Открытый исходный код;
- T2. Быстродействие эмулятора;
- T3. Возможность создания модели вычислительного кластера, состоящего из большого количества ядер, в том числе гетерогенных;
- T4. Возможность создания и подключение моделей периферийных устройств;
- T5. Быстрая и гибкая настройка адресного пространства и карты прерываний;
- T6. Полная документация и наличие методических материалов;

- T7. Гибкость получения трасс выполнения CPU и отладочной информации;
- T8. Возможность сохранения среза архитектурного состояния СнК;
- T9. Возможность программной отладки (GDB).

В таблице 1 представлены сравнительные характеристики программных эмуляторов СнК. Знаком + показано соответствие требованию, знаком - - несоответствие требованию, и частичное соответствие - знаком ±.

Таблица 1 – Сравнение платформ эмуляции СнК

Платформа	T1	T2	T3	T4	T5	T6	T7	T8	T9
Simics	-	+	+	+	+	+	±	+	+
OVPSim	±	+	+	+	+	+	±	+	+
QEMU	+	+	-	±	±	-	±	-	+
Spike	+	±	+	+	+	-	+	-	+
riscv-rust	+	±	±	+	+	+	+	-	-
tinyemu	+	±	+	-	-	+	+	-	-
gem5	+	+	+	-	-	-	-	-	-
whisper	+	-	+	-	-	-	+	-	-
RVVM	+	±	-	±	±	-	±	-	-
nemu	+	-	+	-	-	-	+	-	-
Renode	+	+	+	+	+	+	+	+	+

Разработка программной модели СнК позволяет не только внести существенный вклад в верификацию СнК и проверить различные функциональные сценарии, но и на ранних этапах проработать программную реализацию драйверов для внутренних узлов будущей интегральной схемы и при ее выпуске обеспечить готовность программных библиотек и инструментов для заданных условий эксплуатации.

III. ЗАКЛЮЧЕНИЕ

В работе представлен анализ использования программных эмуляторов применительно к маршруту проектирования СнК, рассмотрены особенности применения в совокупности с предъявляемыми требованиями. В зависимости от подходов применяемых для верификации анализ позволяет на начальной стадии выбрать наиболее удобный инструмент для создания программной модели СнК.

IV. СПИСОК ЛИТЕРАТУРЫ

1. Wilson Research Group functional verification study IC/ASIC functional verification trend report 2020. Mode of access: <http://www.inform.ind.edu/PBIO/>. – Date of access: 10.10.2023.
2. Слинкин Д. И. Анализ современных методов тестирования и верификации проектов сверхбольших интегральных схем. Международный журнал Программные продукты и системы. – 2000. – No 10. Режим доступа: – Режим доступа: <http://www.swsys.ru/index.php?page=article&id=4308&lang=> – Дата доступа: 10.10.2023.
3. Солодовников А. П., Переверзев А. Л., Силантьев А. М. Программно-аппаратный комплекс для ускорения функциональной верификации систем на кристалле // Изв. вузов. Электроника. 2023. Т. 28. No 4. С. 441-451. <https://doi.org/10.24151/1561-5405-2023-28-4-441-451>. – EDN: NULBAQ