

ИРИС – инструментарий разработки интегральных схем в среде C++

Гасанов Э.Э., проф., зав. каф. МаТИС

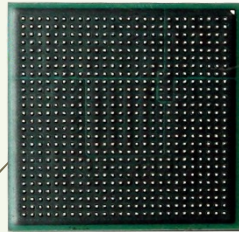
МГУ имени М.В. Ломоносова,

FPGA-Systems, Москва, 1 июня 2024 г.

ИРИС - история создания и использования

- ▶ Начало разработки 2008 год.
- ▶ Использовался в компании LSI при создании аппаратных БЧХ кодеков для твердотельных накопителей, которые были поставлены компаниям Pliant, Sandisk, Samsung.
- ▶ Использовался в компании LSI при разработке сопроцессора помехоустойчивого кодирования для беспроводных сетей 3G и 4G (FEC-сoproцессор), который поддерживал стандарты беспроводной связи LTE, WiMAX, WCDMA (с любыми скоростями передачи данных в рамках этих стандартов).
- ▶ Использовался в компании Huawei при разработке LDPC кодеков для стандарта беспроводной связи 5G.
- ▶ Используется в компании Крафтвэй для создания российского твердотельного накопителя информации и сетевого контроллера со встроенной поддержкой IPSec.
- ▶ Используется в компании Специальный технологический центр для создания специальных помехоустойчивых кодеков для космической связи.

МИКРОСХЕМА ДИСКОВОГО КОНТРОЛЛЕРА ТНИ – ПЕРВЫЙ ПРОДУКТ ДИЗАЙН-ЦЕНТРА KRAFTWAY



- Kraftway - единственная отечественная компания, разработавшая собственную микросхему контроллера ТНИ PCIe NVMe;
- Аналогичные разработки имеют всего 12 компаний из 5 стран мира;
- Все основные работы по созданию микросхемы проведены в России:
 - Создана математическая модель (7 патентов на изобретения);
 - Спроектированы архитектура и алгоритмы работы;
 - Разработаны все основных IP-блоки и интерфейсы;

ФОРМ-ФАКТОРЫ ТНИ



Накопитель в форм-факторе HHHL PCIe



ТНИ в форм-факторе U.2 (в разработке)



Внешний ТНИ с интерфейсом USB 3.0

МИКРОСХЕМА ДИСКОВОГО КОНТРОЛЛЕРА ТНИ ВТОРОЕ ПОКОЛЕНИЕ

ЛИНЕЙКА НАКОПИТЕЛЕЙ KRAFTWAY ТНИ-2 (2024)

Интерфейс	PCIe 4.0 x4, NVMe 1.4			
Параметры	Форм-фактор	M.2 <i>(Client/Mobile)</i>	PCIe/U.2 <i>(Server/Workstation)</i>	EDSFF E1.S/E1.L <i>(Enterprise/DC)</i>
	Тип и интерфейсы NAND-памяти	3D SLC/MLC/ TLC ONFI 4.0/Toggle 3.0 <i>(Micron, Kioxia, YMTC)</i>		
	Объем, ТБ	1/2/4	2/4/8	8/16
Технологии	Garbage Collection, S.M.A.R.T., WDT/PVT-monitor; RISC-V, soft-LDPC, RAID over NAND, Power management			

SSD NVMe ASIC

- Корпус BGA676
- Техпроцесс TSMC 28nm
- Размер ~ 9x9 мм
- Потребление ~3W



M.2
(2280, 2260, 2242)



U.2



PCIe HHL



EDSFF

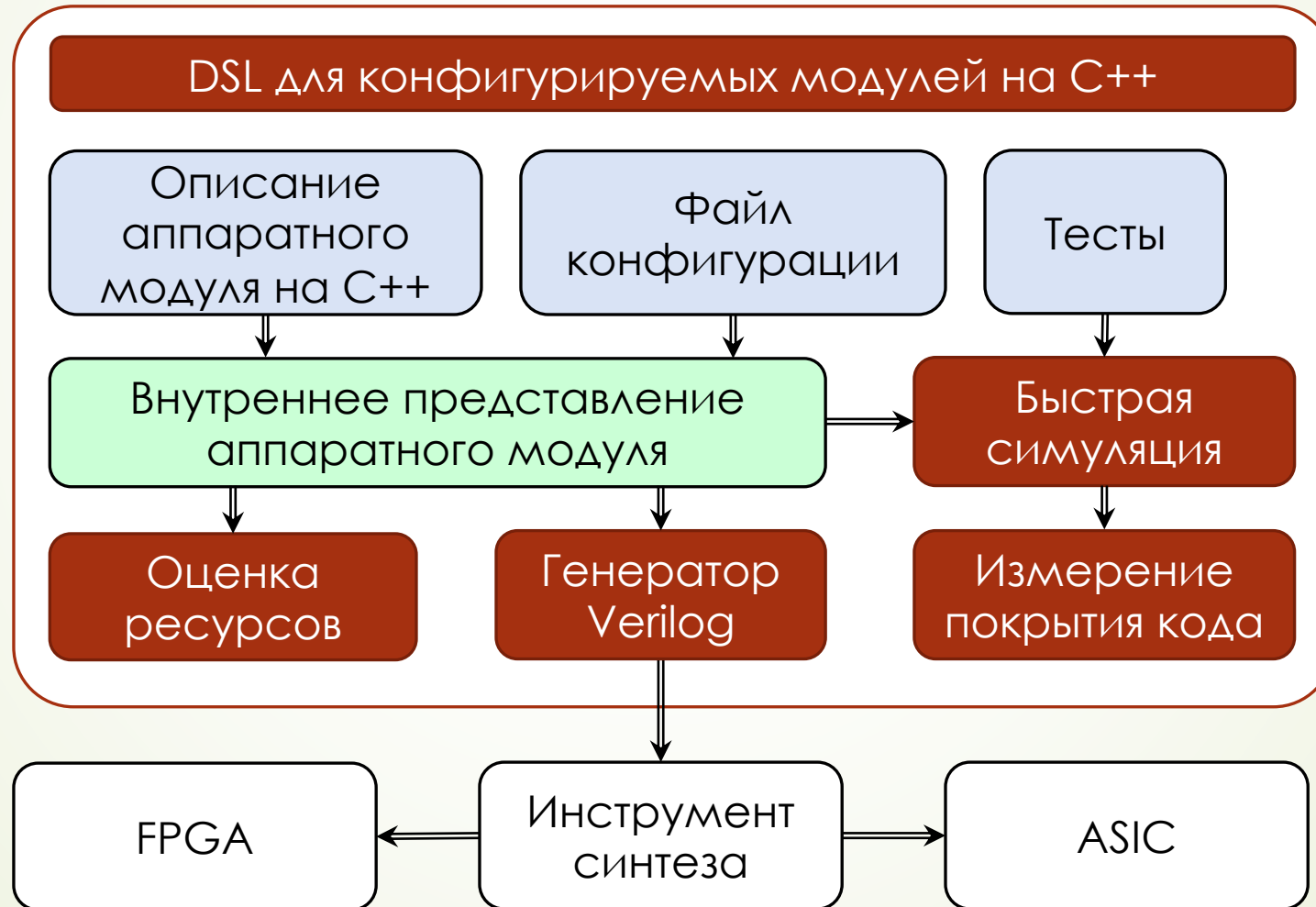
**при доступности TLC/QLC-памяти высокой плотности и ёмкости*

МИКРОСХЕМА СЕТЕВОГО КОНТРОЛЛЕРА СО ВСТРОЕННОЙ ПОДДЕРЖКОЙ IPSEC

- ▶ Пропускная способность до 40 Gb в полудуплексном режиме.
- ▶ Программируемый процессорный кластер ядер RISC-V со встроенным MMU, контроллером прерываний и кэшем L2.
- ▶ Возможность offloading на внешний CPU для поддержки до 100 Security Association.
- ▶ Блок обработки пакетов
 - ▶ Программируемый пакетный процессор для каждого из интерфейсов предназначенный для разбора заголовков пакетов их маршрутизации.
 - ▶ Блок переупорядочивания пакетов предназначен для восстановления порядка следования пакетов
 - ▶ Криптографический модуль реализующий шифрование/дешифрование и имитозащиту пакетов согласно алгоритмам ГОСТ 34.12-2015, создания цифровой подписи используется алгоритм ГОСТ Р 34.10-2012, вычисления хэш функции ГОСТ Р 34.11-2012.
- ▶ Режимы работы:
 - ▶ Сетевой адаптер. Варианты исполнения: PCIe карты 4x1G и 2x10G, интегрированный на плату
 - ▶ Smart NIC с поддержкой IPSec. Варианты исполнения аналогичные.
 - ▶ Высокоскоростной криптоускоритель ГОСТ 34.12-2015

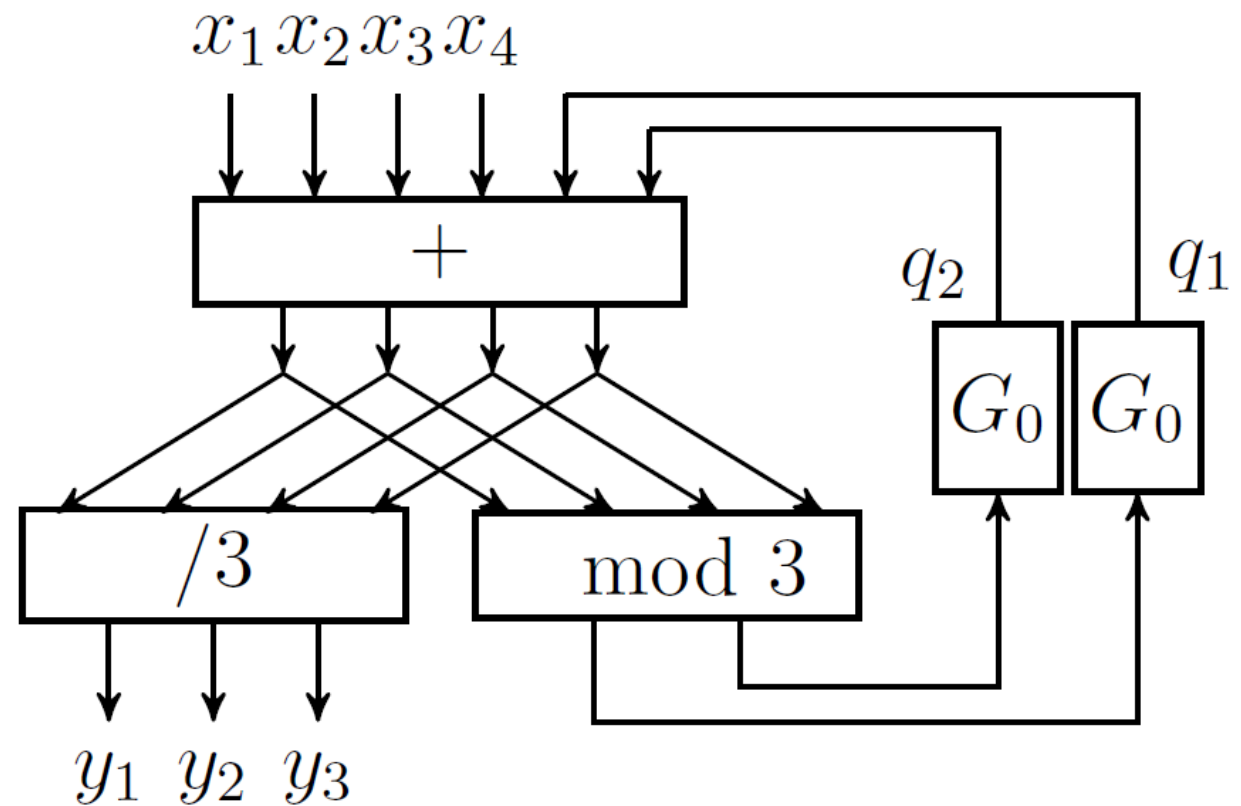


Инструментарий разработки интегральных схем (ИРИС)

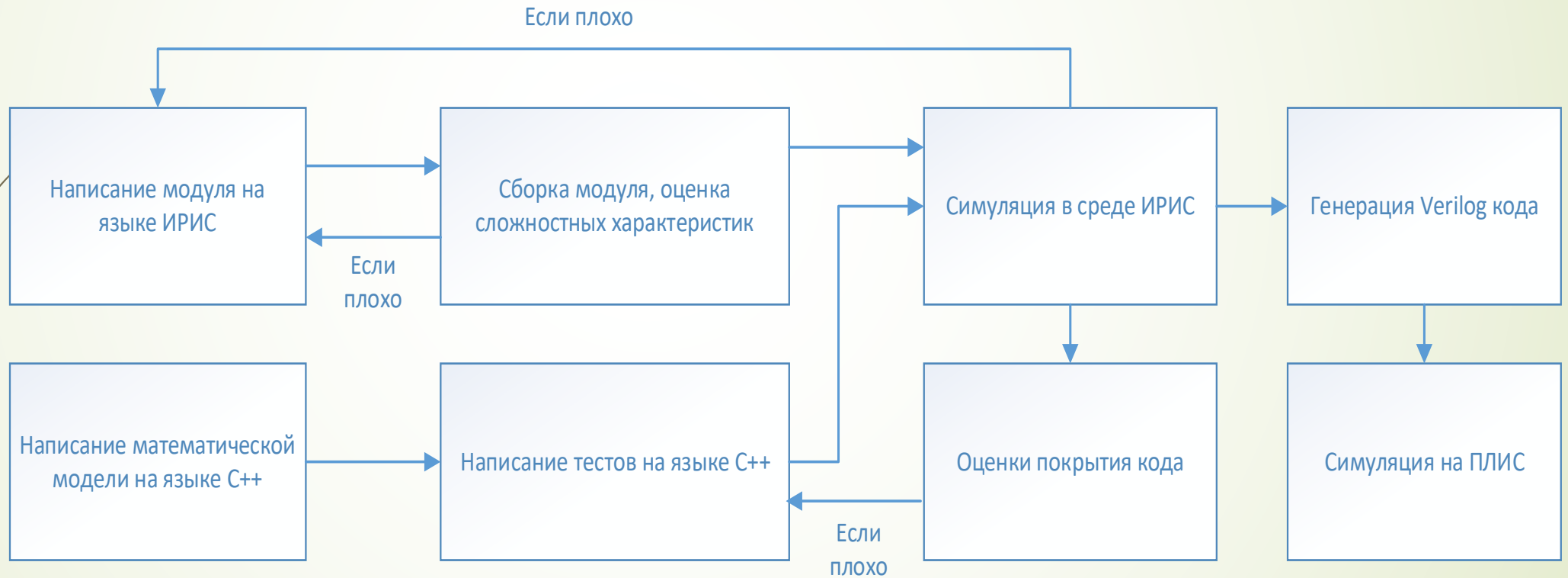


Язык ИРИС

- Основные сущности:
WIRE, REG, SUBMODULE,
E-выражение.
- WIRE(x, 4); REG(q, 2);
WIRE(c, 4); WIRE(y, 3);
- $c = E(x) + E(q)$;
- SUBMODULE(MOD3, ins1);
- ins1.init(4);
- ins1.IN = E(c);
- $q = E(\text{ins1.OUT})$;
- SUBMODULE(DEL3, ins2);
- ins2.init(4);
- ins2.IN = E(c);
- $y = E(\text{ins2.OUT})$;



Разработка СФ-блоков в среде ИРИС



Пример, который трудно реализовать на Verilog'e – создание LDPC декодера для средств связи

- ▶ Надо создать LDPC декодер, который может работать с большим количеством LDPC матриц и на ходу переключаться между ними.
- ▶ Пишем на C++ программу, которая симулирует работу LDPC декодера на одной матрице.
- ▶ Из конфигурационного файла узнаем каталог, где лежат LDPC матрицы и прогоняем программу на каждой из матриц.
- ▶ В результате симуляции формируем содержимое памяти команд LDPC декодера и определяем все параметры LDPC декодера.
- ▶ Сформированные данные используем в качестве параметров для генерации требуемого LDPC декодера.
- ▶ Симуляционную программу можем использовать для тестирования и верификации созданного аппаратного модуля.

Ключевые преимущества ИРИС

Система **ИРИС** является высокоуровневым языком описания аппаратуры (**HDL**), нацеленным на **быстрое** создание и тестирование сложных аппаратных модулей.

В отличие от **систем высокоуровневого синтеза** (Vivado HLS, Google XLS, ...) в системе ИРИС разработчик имеет **полный контроль** (на уровне гейтов) над создаваемой аппаратной схемой.

Важное преимущество ИРИС перед другими высокоуровневыми языками описания аппаратуры (SystemVerilog, Chisel, ...) заключается в том, что она реализована в виде **библиотеки C++**, одного из самых популярных языков программирования, для которого существует большой выбор высокоэффективных компиляторов (Visual Studio, GCC, Clang).

Система ИРИС обладает **встроенным сверхбыстрым** симулятором, позволяющим избежать использования внешних средств (Verilator, Questa Sim, Vivado, ...), и работающим в **десятки раз быстрее** чем встроенный симулятор Chisel.

Система ИРИС

Система **ИРИС** является **полностью российской разработкой**, что позволяет с легкостью корректировать вектор ее развития, и в дальнейшем **интегрировать** ее в стек российских средств разработки микроэлектроники.

Выбор языка популярного языка C++ особенно важен в данном ключе, поскольку это значительно **понижает порог входа** для большинства инженеров, и так хорошо знакомым с ним.

Создание аппаратной модели (RTL Verilog) является наиболее **затратным и долгим** этапом (1.5-2 года) при проектировании.

Многочисленные примеры использования ИРИС при проектировании сложных микросхем показали, что время на проектирование сокращается в **несколько раз**.

Подробное сравнение ИРИС с другими системами

Возможность	ИРИС	System Verilog / VHDL	Chisel	HLS (Vivado HLS/Google XLS)
Полный контроль на уровне гейтов при синтезе	Есть	Есть	Есть	Нет (язык высокого уровня транслируется в RTL)
Гибкая конфигурируемость	Есть	Нет	Есть	Есть
Типизация проводов	Есть	Есть	Есть	Есть
Встроенная быстрая оценка сложности хардвера	Есть	Медленная, но более точная (требуется синтез)	Нет	Есть (по крайней мере в Vivado HSL)
Встроенные средства тестирования	Есть	Ограниченные	Есть	Есть

Скорость симуляции (LDPC декодер для ТНИ)

Вид симуляции	ИРИС	Questa Sim	Vivado	Icarus Verilog	Verilator (с опцией -O3)
Простая симуляция	7	13.7	187.5	470	10
С измерением Toggle coverage	9	75			
С измерением Branch coverage	11	18			

Скорость симуляции (Mersenne Twister, N бит за такт)

	ИРИС	Chisel (встроенный симулятор)	Verilator (с опцией -O3)
$N=32$			
дамп – входы и выходы, ~300КВ	0.01 (усреднение по 100 запускам)	4.5	0.047 (усреднение по 100 запускам)
$N=32768$			
дамп – входы и выходы, ~660МВ	1.0	100	1.2



Текущие разработки

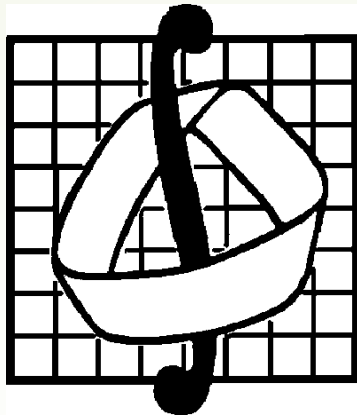
- ▶ Разработка средства оценки количества логических элементов и задержек, а также глубины разрабатываемых схем.
- ▶ Разработка средств для создания модулей с несколькими тактовыми сигналами.
- ▶ Разработка средств оптимизации логики схем.
- ▶ Разработка средств выявления критических путей и добавления на них регистров и последующую простановку регистров на остальные пути с целью сохранения корректности функционирования.
- ▶ Разработка дополнительных модулей библиотеки готовых модулей.
- ▶ Разработка программной документации на программное изделие.



Планы



- Разработать транслятор с языка Verilog во внутренние структуры средства ИРИС.
- Разработать компилятор с языка Verilog на язык ИРИС.
- Разработать средства совместных отладки и тестирования модулей, написанных на языке ИРИС и языке Verilog.
- Разработать высокоскоростное средство симуляции модулей, написанных на языке Verilog (создать отечественный verilator).



Спасибо за внимание!

FPGA-Systems, Москва, 1 июня 2024 г.



Генеральный партнёр конференции FPGA-Systems 2024.1



Первая современная отечественная САПР,
реализующая сквозной цикл проектирования печатных плат



www.aremex.ru

Где найти FPGA / RTL / Verification комьюнити?

FPGA-Systems.ru

Сайт комьюнити

[FPGA-Systems Magazine \(FSM\)](#)

Первый журнал о программируемой логике

[@fpgasystems](#)

Телеграм чат

admin@fpga-systems.ru

Электронная почта

Youtube.com/c/fpgasystems

Youtube канал

