Развитие Средств Проектирования Микроэлектронных устройств

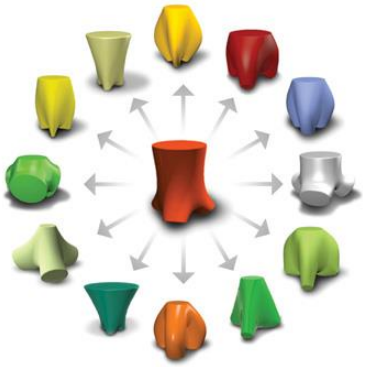
Зав. лабораторией МПАСС, МФТИ
Д.т.н., проф. Александр Дроздов

Сентябрь, 2020

Мировые тенденции в микроэлектроники (и не только)

ПОСТ-индустриальная реальность

Характеристики ПОСТ-индустриальной экономики



1. Исключение человека из производственного процесса: задача человека - описать функциональность в форме, удобной для перехода в беспилотный технологический цикл
2. Основным результатом здесь являются технологии для производства и модификации всего спектра решений и услуг
3. **Кастомизация**: «массовые» товары и услуги исчезают. Каждый потребитель получает уникальный продукт или услугу
4. Прибыль переносится с «физического» уровня производства на производство интелектуальной собственности (IP)

Микроэлектроника - авангард новой реальности

- Технологии проектирования, производства и бизнеса от микроэлектроники распространяются на другие отрасли

Рынок систем на кристалле (SoC)

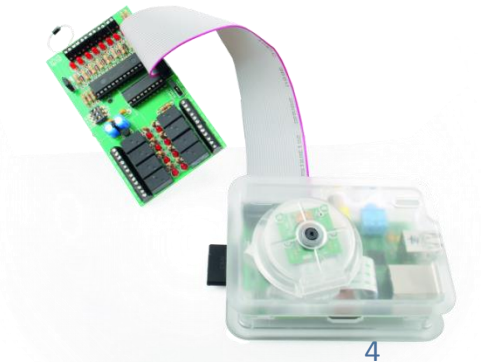
- Процессоры общего назначения (CPU, GPU, DSP)
  - Рынок принадлежит лидерам
  - Установленные каналы продаж
  - **$$B** инвестицый в R&D

- Растущие рынки
  - Интернет вещей **(IoT)**
  - Умные дома
  - Роботы, БПЛА
  - Виртуальная и дополненная реальность
  - Искусственный интеллект
  - Носимая электроника

- **Окно Возможностей**

  Специализированнные SoCs

  для новых технологий и устройств



Техпроцесс и производительность

- Процесс **350nm** (1995) → **50x** / **20 лет** → **7nm** (2019)

- **Рост реальной производительности <50x**
  - Многоядерные системы используют 25-50% от пиковой производительности
  - Почему 8-ядерные смартфоны лучше, чем 4-ядерные?

- **Окно Возможностей**: кастомизация и новое поколение САПР
  - SoC кастомизация для приложений позволяет достичь лучшую производительность с минимальной стоимостью и минимальным энергопотреблением
  - Новый системный уровень САПР для достижения лучшего времени выхода на рынок в индустрии

Эволюция современных архитектур

Ядра микропроцессоров

- Виды ядер по назначению
  - **CPU** – центральный процессор, процессор общего назначения
  - **GPU** (graphical processing unit) – процессор ускорения отрисовки графики
  - **DSP** (digital signal processor) – процессор цифровой обработки сигналов, для телекоммуникаций, работы с видео и звуком.
  - Новый тип: **TPU** (tensor processing unit) – процессор для ускорения алгоритмов ИИ.

- Самые популярные архитектуры CPU
  - Семейство архитектур x86
    - ПК, сервера, ноутбуки.
  - Семейство архитектур ARM
    - Смартфоны, планшетные компьютеры, встроенные системы, IoT
  - Семейство архитектур MIPS (на спаде)
    - Маршрутизаторы, игровые консоли, цифровые приставки.
  - Перспектива: открытая архитектура RISC-V.
    - Возможность свободного использования, любой может реализовать и вывести на рынок собственную реализацию
    - Пока только начало внедрения, архитектура претендует на ниши ARM.

Архитектурные технологии

- Ключевые архитектурные технологии
  - **CISC** (Complete Instruction Set Computer) или **RISC** (Reduces Instruction Set Computer)
    - CISC – большое количество сложных, в том числе много-тактовых инструкций. Пример x86
    - RISC – минимизированный набор инструкций, большинство одно тактовые. Пример ARM
  - **VLIW** (very long instruction word) – длинное слово из нескольких команд, исполняемых параллельно
  - **Out of order** – система «внеочередного исполнения команд»,
    - Процессор динамически выбирает одну или несколько команд на исполнение по мере готовности, при этом порядок исполнения команд может изменяться.
  - **SIMD** (single instruction, multiple data) – параллельное исполнение одной инструкции над несколькими данными.
  - **HyperThreading** (технология Intel) – аппаратная многопоточность
    - Несколько потоков исполняются внутри одного ядра.
    - Каждый поток имеет собственный регистровый файл, вычислительные ресурсы (ALU) разделяются
- Направления эволюции архитектур
  - Комбинирование технологий: VLIW + SIMD, или out of order + аппаратная многопоточность.
  - Кастомизация набора инструкций: RISC дополняется специализированными инструкциями для ускорения конкретных алгоритмов (или групп похожих алгоритмов)

Бинарная трансляция

Бинарная трансляция - эмуляция одного набора инструкций на другом за счет **трансляции** машинного кода.

Бинарная трансляция – способ обеспечения совместимости с другими архитектурами

- Например: поддержка ПО для ARM на собственных ядрах.

Типы бинарной трансляции

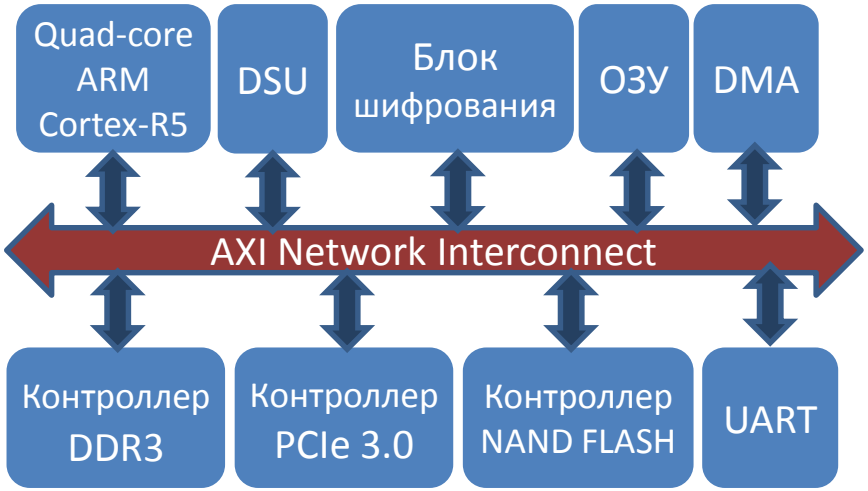
- **Статическая –** весь код транслируется перед запуском программы
- **Динамическая –** код транслируется в процессе исполнения

GIP – указатель на гостевую инструкцию

Архитектуры современных микропроцессоров

Структура системы на кристалле (СнК)

- Одно или несколько ядер CPU
- Опционально: ядра GPU или DSP
- Подсистема памяти
  - Интерконнект
  - Память ПЗУ, OTP – однократно программируемая память
  - Память ОЗУ
  - Память кэш
  - Flash- память
  - Интерфейсы внешней памяти (DDR)
- Интерфейсы для общения с внешним миром
  - PCI, USB, Ethernet, UART, SPI, I2C
- Аппаратные ускорители
  - Сжатие и обработка видео
  - Шифрование и вычисление ЭЦП
  - Специализированные алгоритмы

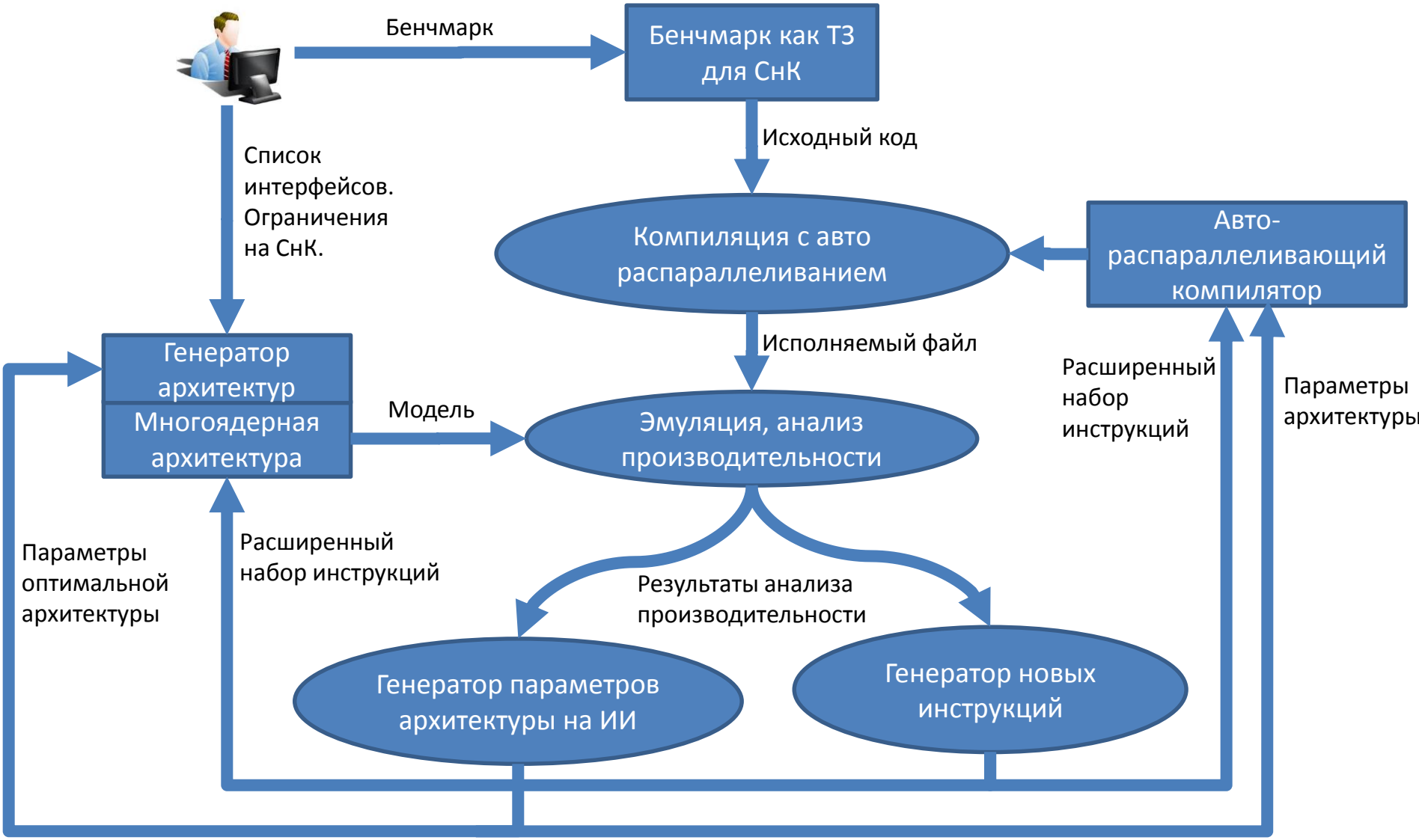


Этапы разработки СнК

- Выбор базовых блоков (CPU, GPU, интерфейсные блоки, элементы памяти)
- Кастомизация блоков: расширение набора инструкций, подпор параметров
- Разработка ускорителей – перенос алгоритмов с уровня ПО на уровень аппаратуры.
- Сборка СнК из отдельных блоков
- Логический и физический синтез, запуск в производство
- Верификация готового чипа.

Современные технологии проектирования

Автоматическое проектирование СнК



Автоматическое проектирование СнК (2)

- **Входные данные для проектирования СнК**
  - Приложение или набор приложений (бенчмарки)
  - Список интерфейсов для коммуникаций с внешним миром (PCI, USB, SPI, GPIO и т.д)
  - Аннотация кода: метки на функции, которые могут быть реализованы как аппаратные ускорители
- **Варианты критериев оптимизации СнК**
  - Минимальное время работы бенчмарков при ограничении по размеру кристалла и\или энергопотреблению
  - Минимальные размер или энергопотребление при заданном максимальном времени работы бенчмарков
- **Результат проектирования**
  - Синтезируемая RTL-модель СнК
  - Программные модели
  - Системное ПО и средства разработки

Стратегия генерации оптимальной СнК

- Выбор оптимальных параметров СнК: количество ядер, размеры кэш и\или обычной памяти, количество и разрядность шин, размеры VLIW и SIMD инструкций.
- Синтез новых инструкций на основе результатов профилирования

- **Компоненты системы проектирования**
  - Высокоуровневые языки описания аппаратуры с возможностью параметризации моделей и генерации различных сущностей (эмулятор, RTL, драйвера и т.д.)
  - Оптимизирующий, авто-распараллеливающий компилятор
  - Система анализа производительности
  - Компиляторы высокоуровневых языков программирования в ПЛИС\СБИС
  - Алгоритмы генерации моделей СнК (подбор параметров) на основе анализа производительности с применением ИИ.
  - Алгоритмы генерации системного ПО (драйвера, HAL) и средств разработки
  - Алгоритмы генерации новых инструкций.
  - Скрипты для итеративного автоматического проектирования СнК

Универсальная Библиотека Трансляции

Задача – разработка Компонентная Технология Оптимизирующей Трансляции (КТОТ) – подхода, абстрагирующего алгоритмы анализа и оптимизаций от конкретного промежуточного представления компилятора

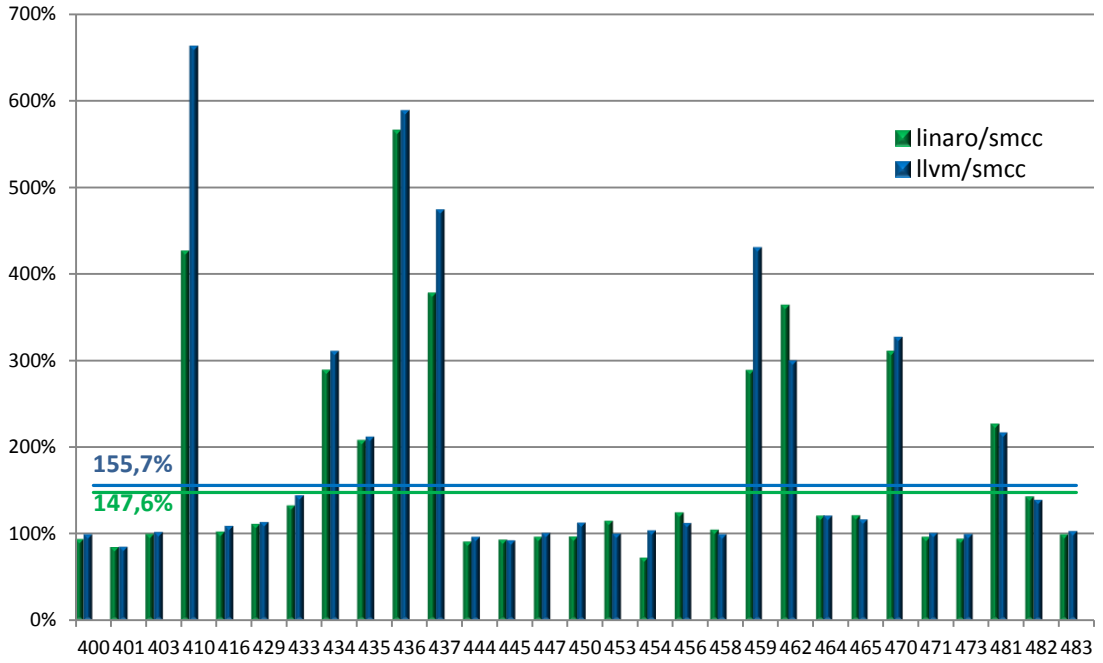
Универсальная Библиотека Трансляции (UTL) – реализация КТОТ

УБТ апробирована в составе:

- GNU
- LLVM
- QEMU (x86->ARM)
- В JIT-компиляторе компании Soft Machines Inc.

На основе UTL и LLVM создан авто-распараллеливающий и авто-векторизующий компилятор **smart-cc**

На пакете SPEC/CPU2006 на ARM **8 ядер** выигрывает **47%** и **55%** у Linaro-GCC и LLVM, на MIPS 2 ядра **13%** и **20%**



Язык PPDL

Внутренняя разработка

Задача – разработка языка описания архитектуры с принципом «одно описание – много реализаций», что дает

- Автоматическая генерация компонент ядра (синтезируемое RTL-описание, ассемблер, компилятор, отладчик и т.д.)
- MIMD-параметризация ядра

Язык PPDL – Processor Description Language обладает следующими преимуществами

1. Сокращение затрат на разработку процессоров в 3-7 раз
2. Сокращение затрат на модификацию в 7-10 раз
3. Как следствие предельно низкой стоимости модификаций – неограниченные возможности для итеративного поиска оптимальных аппаратных решений

Внедрено: Для автоматической генерации компонент средств разработки для NeuroMatrix

Дальнейшее развитие PPDL:

- Развитие компилятора PPDL в сторону генерации Гиперядерных ядер.
- Генерация back-end компилятора (llmv, gcc)
- Автоматическая генерация новых инструкций на основе данных анализа производительности приложений

Технология компиляции C\C++ в ПЛИС

Задача: обеспечить минимальное время переноса C\C++ программ в ПЛИС или СБИС (компилято C2FPGA)

Недостатки существующих решений

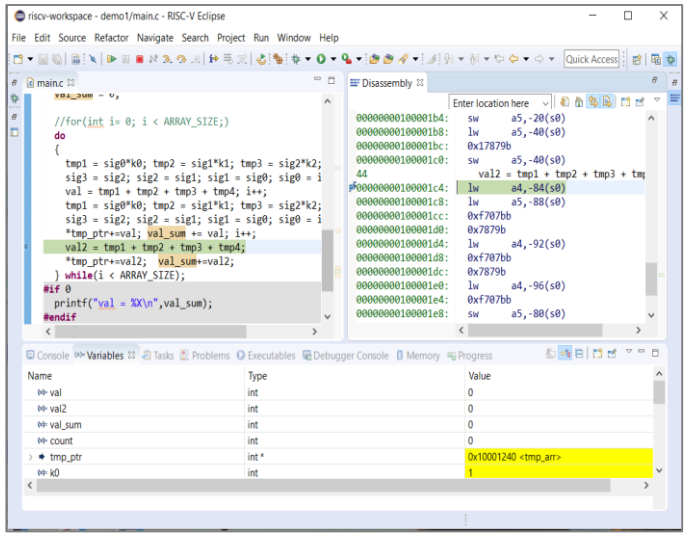
- Множество ограничений на исходный код
- Отсутствует отладка на ПЛИС на уровне исходного кода
- Нет эффективного автораспарраллеливания на уровне функций
- => Решения подходят для переноса только небольших функций, перенести большое приложение - проблема

Виртуальный СнК + автораспараллеливающий компилятор

+ Основа – «виртуальный микрокомпьютер» с виртуальной периферией, стандартными библиотеками и классическими средствами разработки и отладки.
+ Приложения компилируются авто распараллеливающим компилятором нашей разработки
+ Предусмотрена система отладки ПО как для виртуального СнК

Преимущества

- Возможность портировать сложные приложения
- Нет ограничений на исходный код
- Дружественная к программистам среда разработки
- Подходит для разработки встроенных систем, систем на кристалле и высокопроизводительных решений (ПЛИС-кластеров)

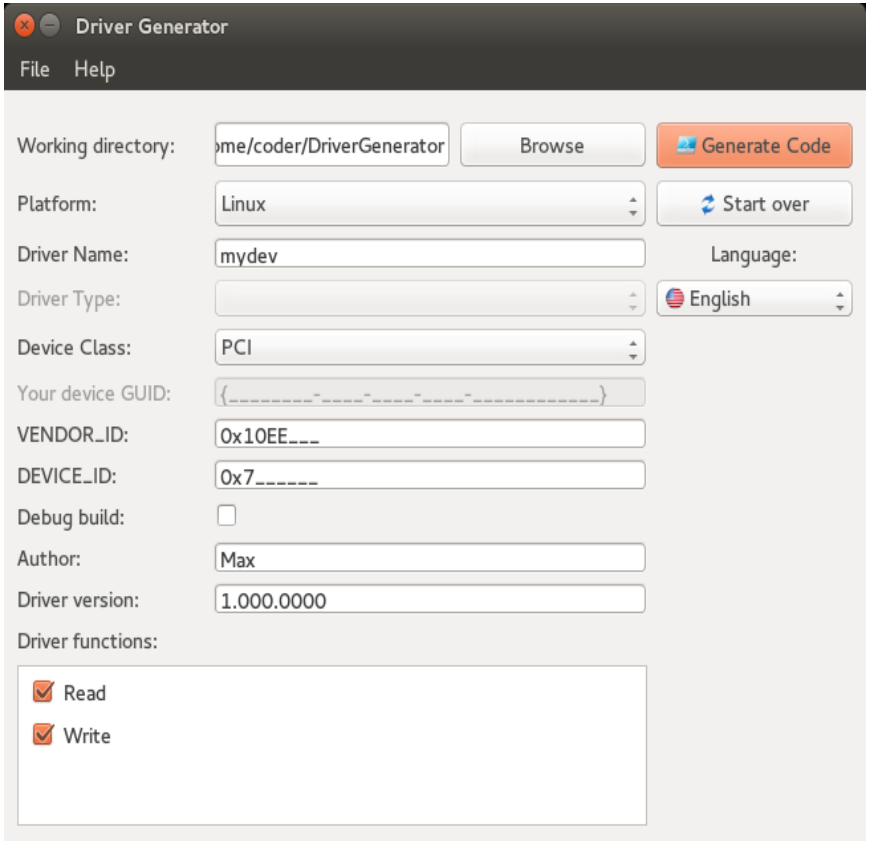


Генератор Драйверов

Задача – создание программного инструмента, облегчающего разработку драйверов одновременно для ОС Windows и Linux

- Аналог: Jungo WinDriver
- Конкурентные преимущества – разработка драйвера одновременно для нескольких ОС

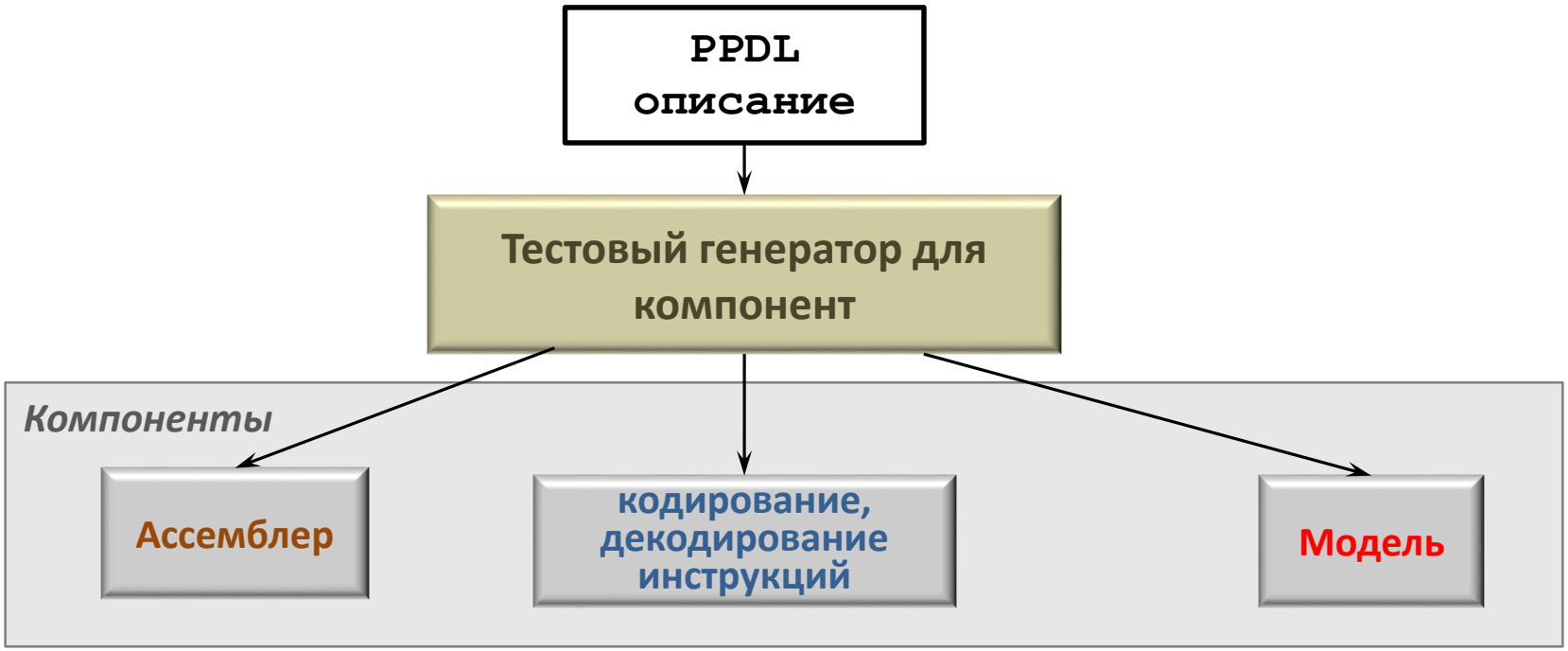
Статус проекта: тестирование, внедрение



Верификация

Разработан уникальный подход Генератора Случайных Тестов, используется для:

- **HW & SW тестирование**
  - Генерация тестов для высокого покрытия при функциональной верификации на симуляторе
  - Верификация HW дизайна
  - Отладка на кристалле
  - Стрессовое тестирование
- **Разработка направленных тестов для воспроизведения HW/SW ошибок**



IP – блоки (примеры)

IP библиотека стандартных периферийных блоков.

- Контроллер QSPI.
- Контроллер интерфейса i2c (master/slave).
- Таймер общего назначения.
- Контроллер GPIO.
- Контроллер UART.
- Контроллер Ethernet 1000BASE-T (MAC)
- Контроллер AXI4 DMA.
- Контроллер DDR4 SDRAM памяти (в разработке)

IP библиотека аппаратных блоков криптографических функций.

Хэш-функции:

- Семейство MD
- Семейство SHA
- Семейство RIPEMD
- Семейство Whirlpool
- Семейство Стрибог
- Параметризуемый блок HMAC

Блочные шифры:

- Семейство DES
- Семейство AES
- Семейство RC
- Blowfish
- Семейство Кузнечик

Спасибо за внимание!