

Введение в LabVIEW FPGA

Технология FPGA

Блоки памяти

Хранят наборы данных или значения в определяемой пользователем области памяти

Конфигурируемые логические блоки (CLBs)

Реализуют логику, используя триггеры и таблицы преобразования (LUTs)

Блоки умножения и цифровые сигнальные процессоры (DSPs)

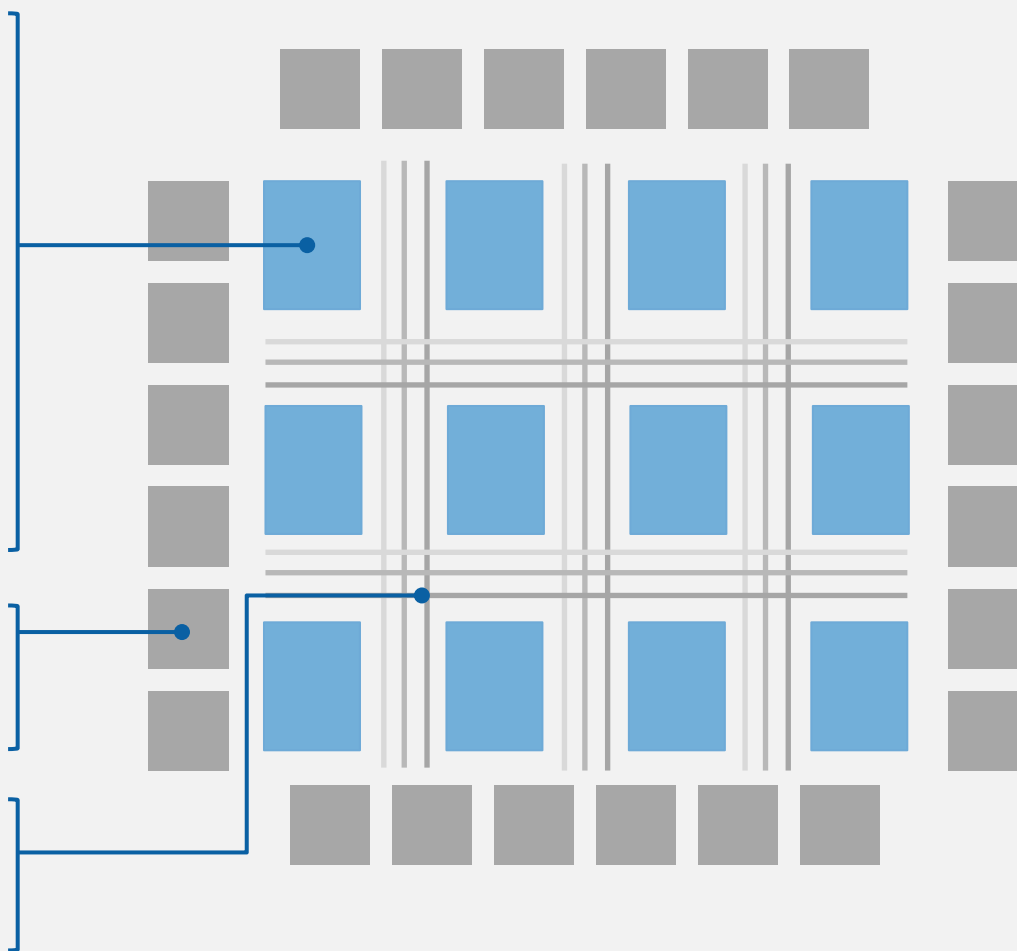
Реализуют обработку сигналов используя схемы умножителей и умножителей-накопителей

Блоки ввода-вывода

Обеспечивают непосредственный доступ к линиям цифрового и аналогового ввода-вывода

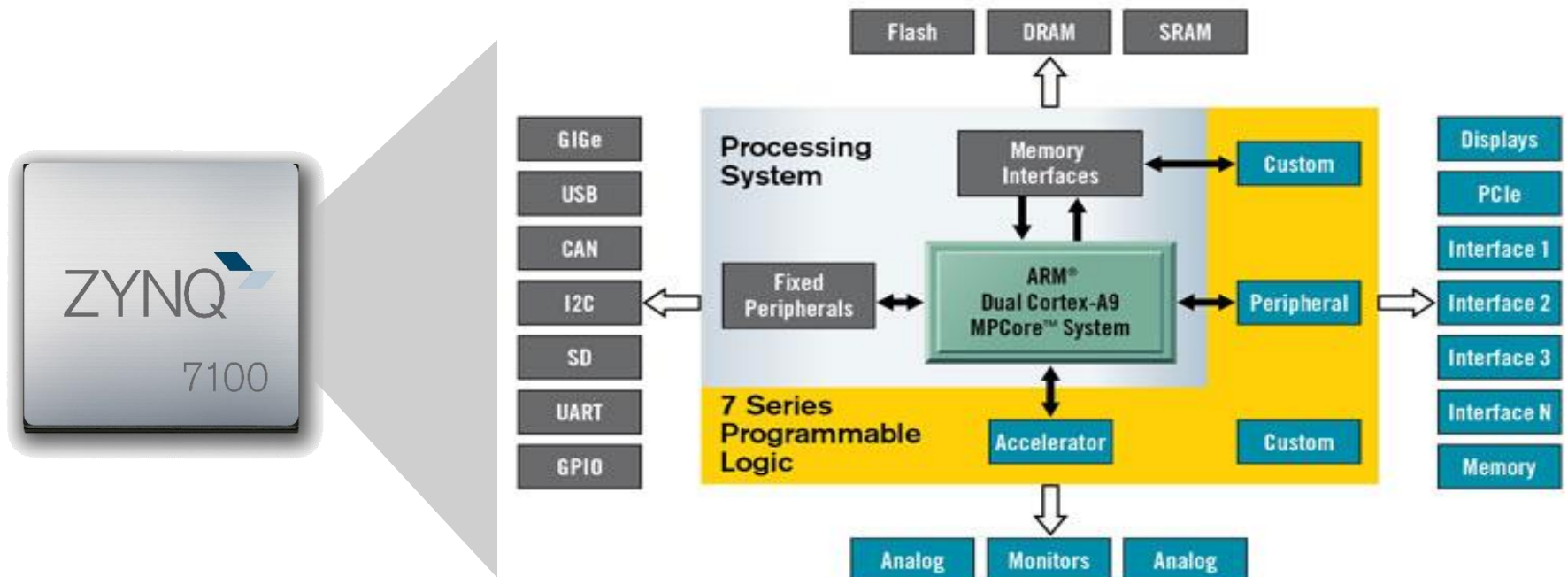
Программируемые межсоединения

Создают маршруты распространения сигналов в матрице FPGA



Будущее FPGA:

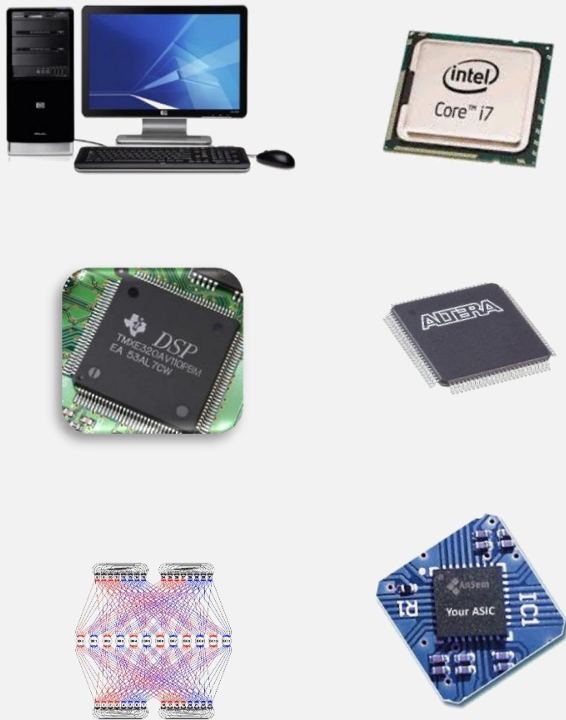
Гетерогенные, с массовым параллелизмом системы на кристаллах



- Снижение потребляемой мощности
- Уменьшение общей площади на плате
- Улучшение реконфигурирования
- Снижение стоимости

Эволюция подходов к технологии проектирования

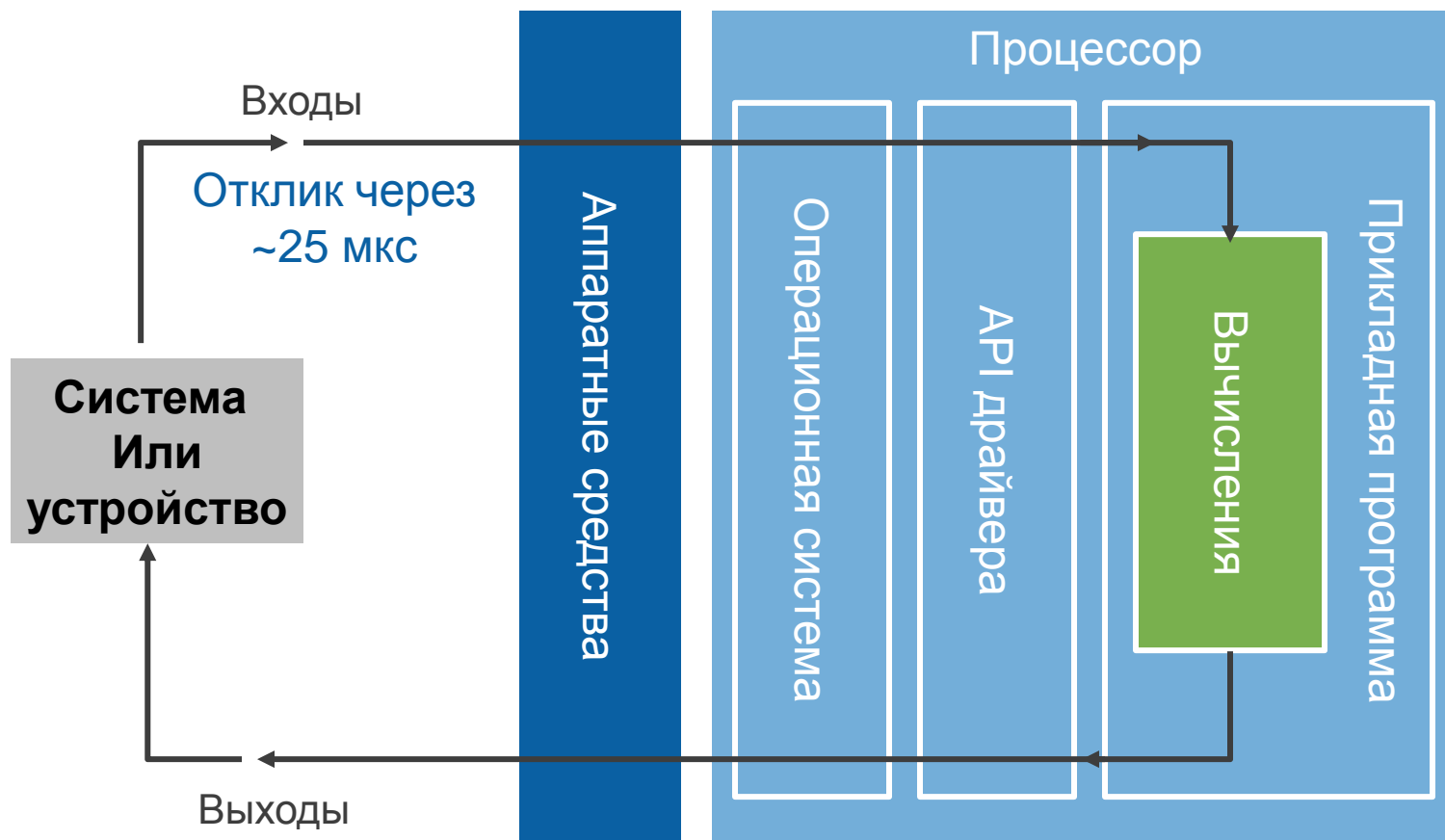
Традиционные компоненты



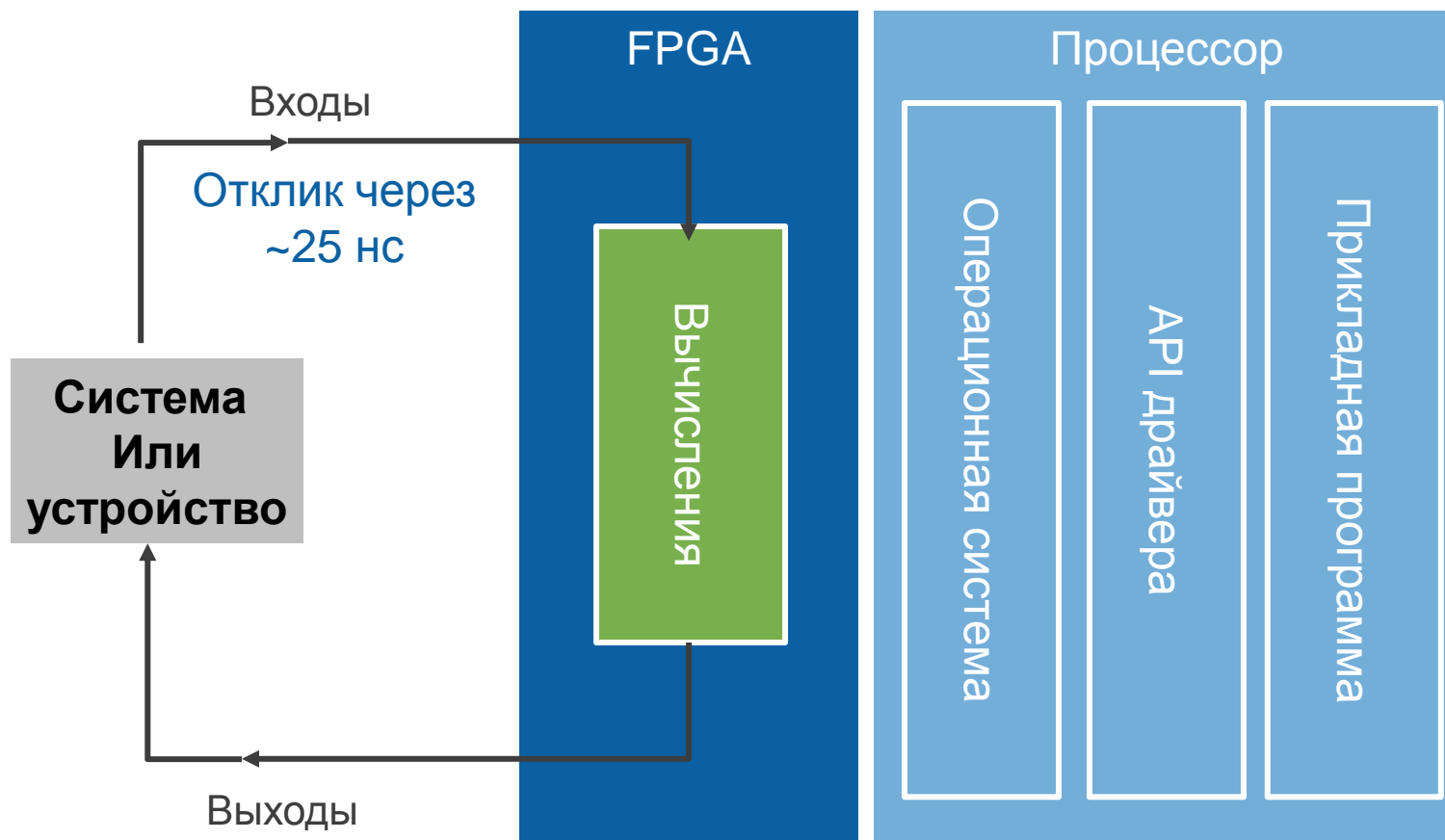
Более совершенные



Метод, основанный на процессоре

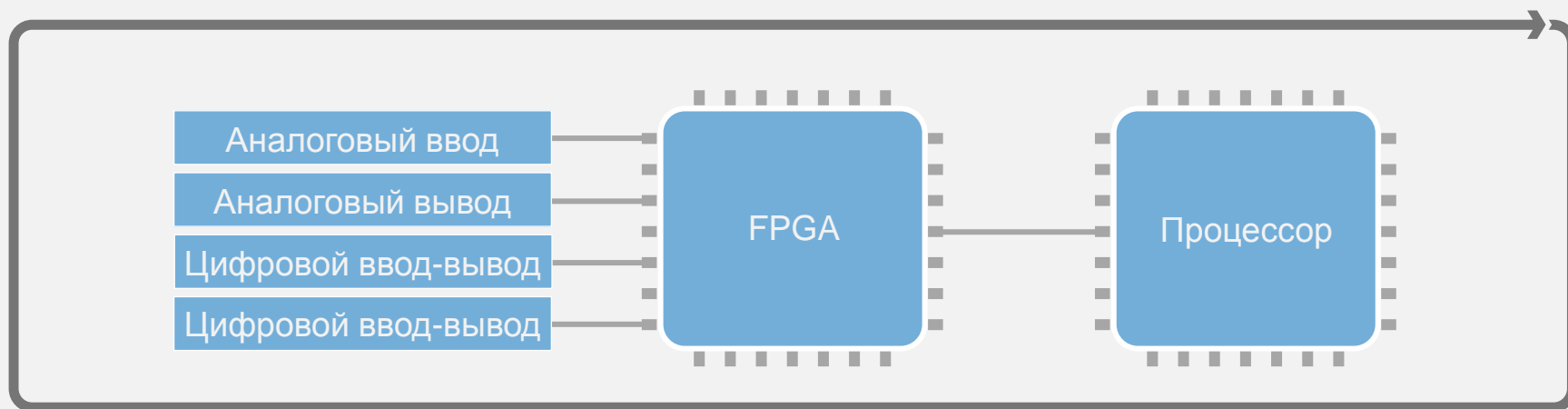


Принятие решений в схеме FPGA



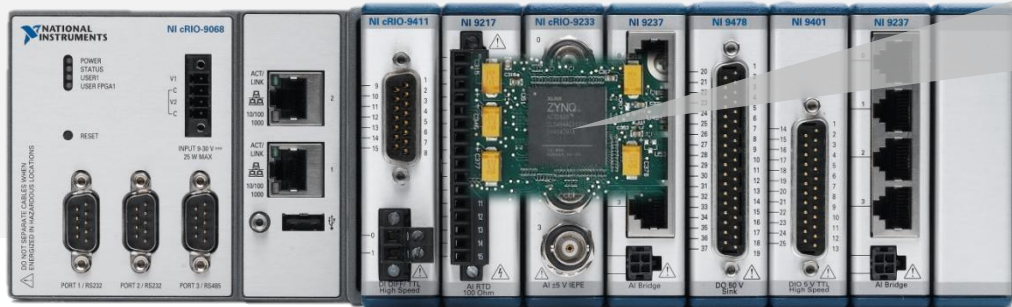
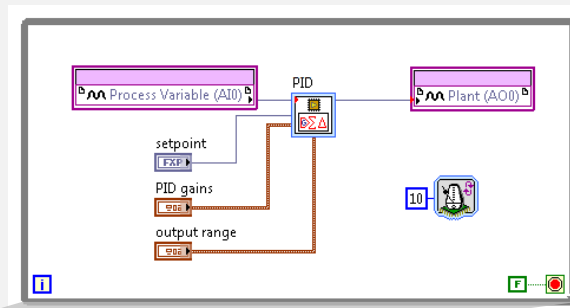
Подход NI

Архитектура NI LabVIEW RIO



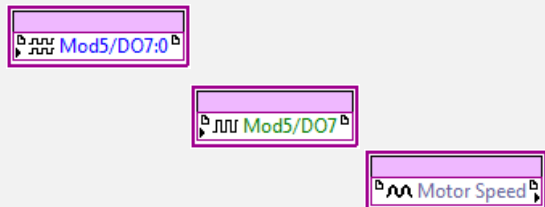
Модуль LabVIEW FPGA

- Использование LabVIEW для разработки аппаратных средств
- Разгрузка наиболее важных частей вашего приложения
 - Высокоскоростное управление
 - Обработка сигналов в реальном времени
 - Пользовательские протоколы
 - Специализированное тактирование, запуск и синхронизация
 - Быстрое тестирование методом стимул/отклик

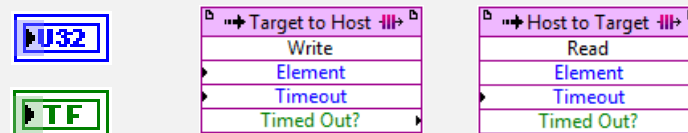


Элементы LabVIEW FPGA

Интерфейс ввода-вывода



Обмен данными



Тактирование

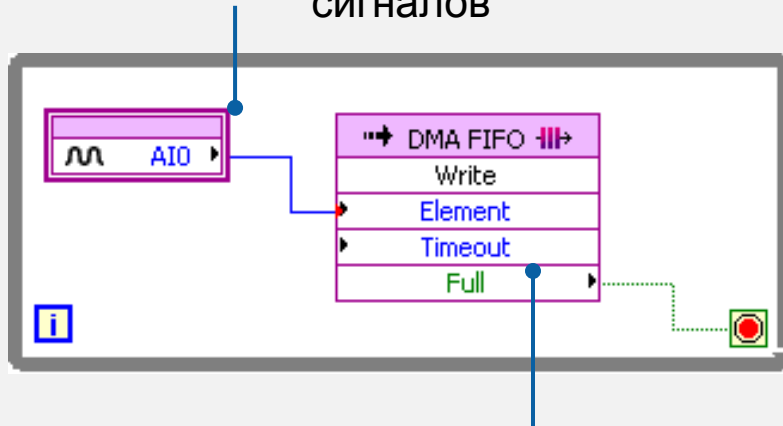


Управление



Абстрагирование от сложности аппаратных средств

Поточечное измерение аналоговых сигналов



Прямая передача аналоговых данных в память компьютера через FIFO для регистрации данных, визуализации и т.п.

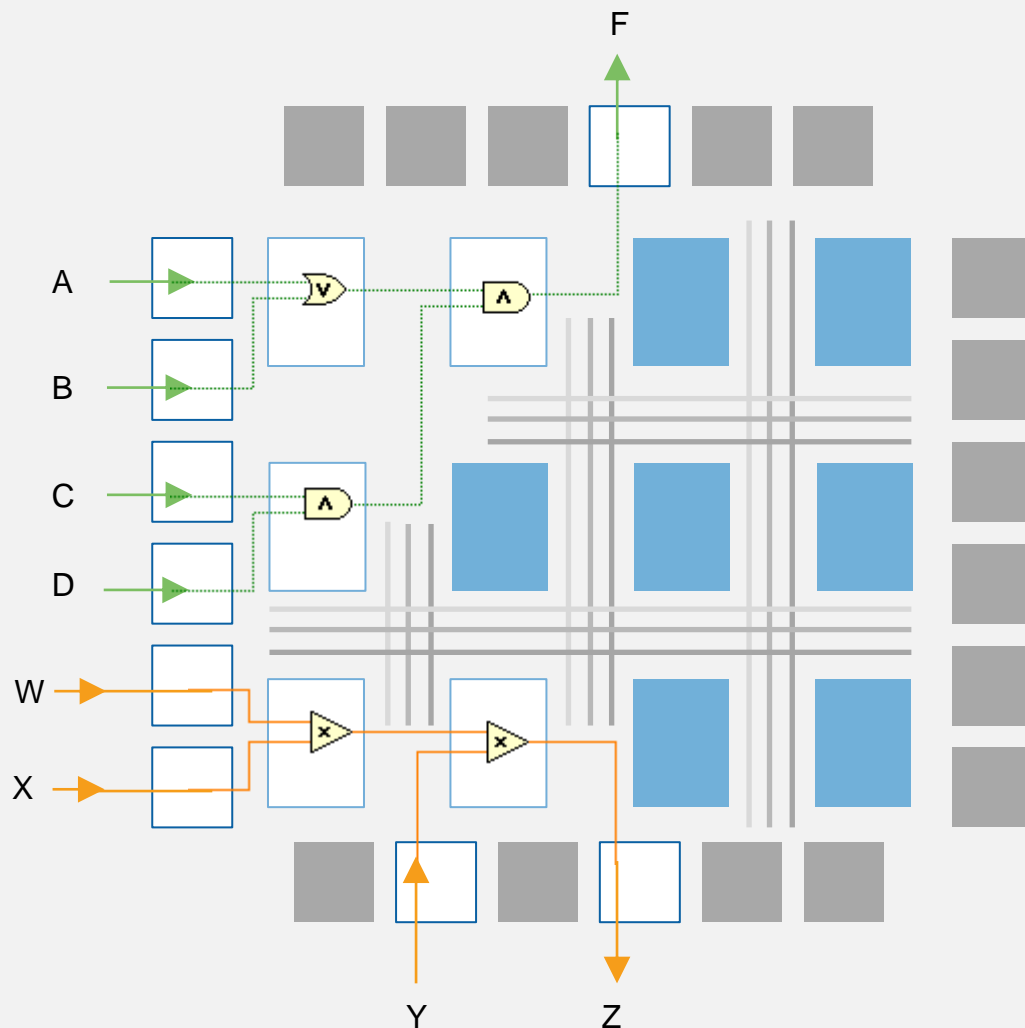
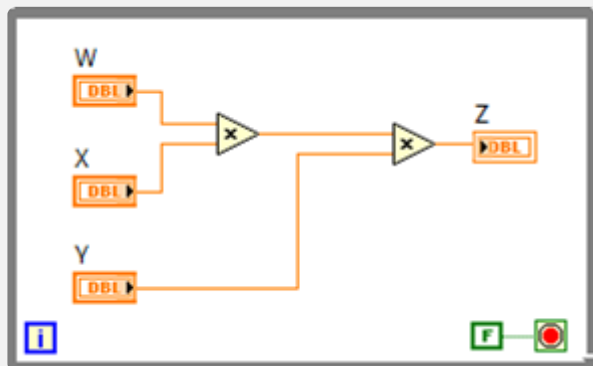
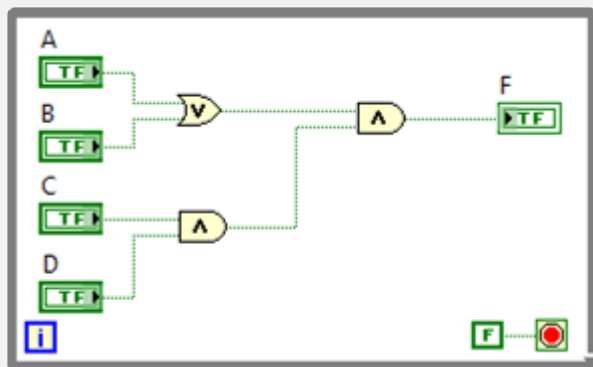
LabVIEW FPGA

сравните

~4000 строк VHDL

VHDL

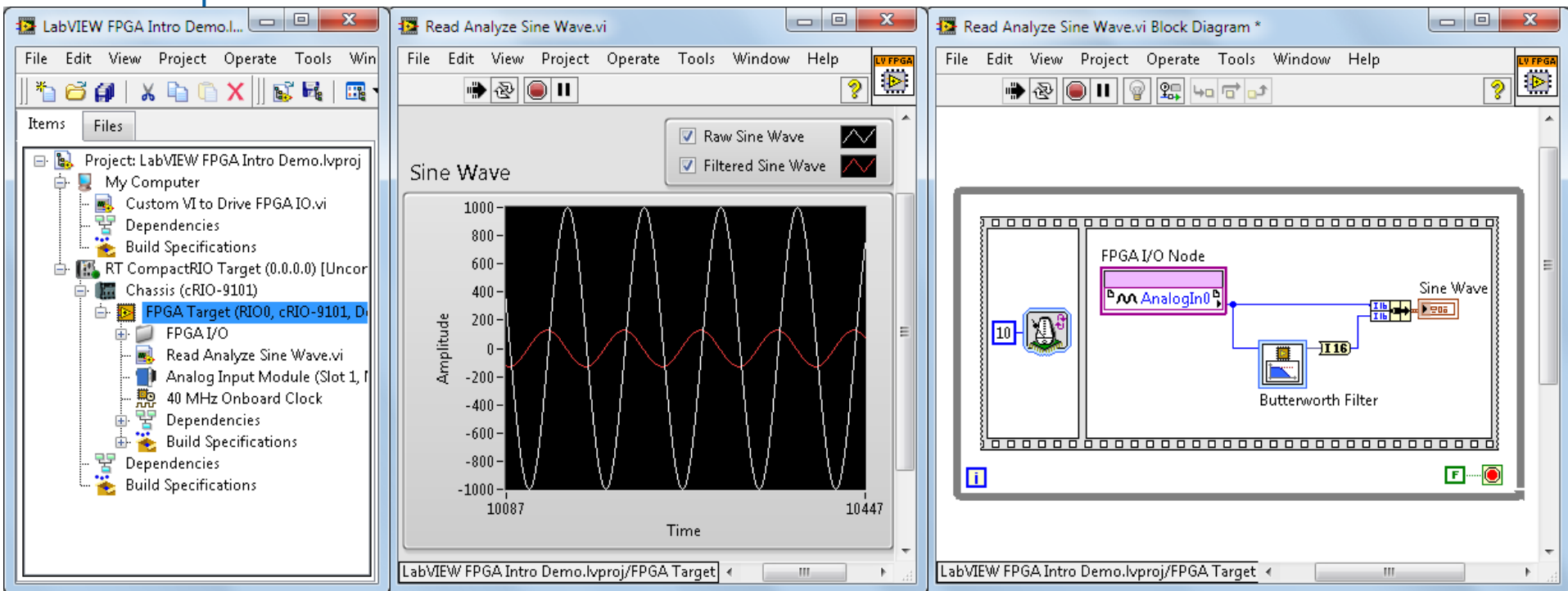
Отображение LabVIEW на FPGA



Среда графического программирования LabVIEW

“Проект” = Конфигурация
системы

“VI” = Приложение



“Лицевая панель” = Элементы интерфейса

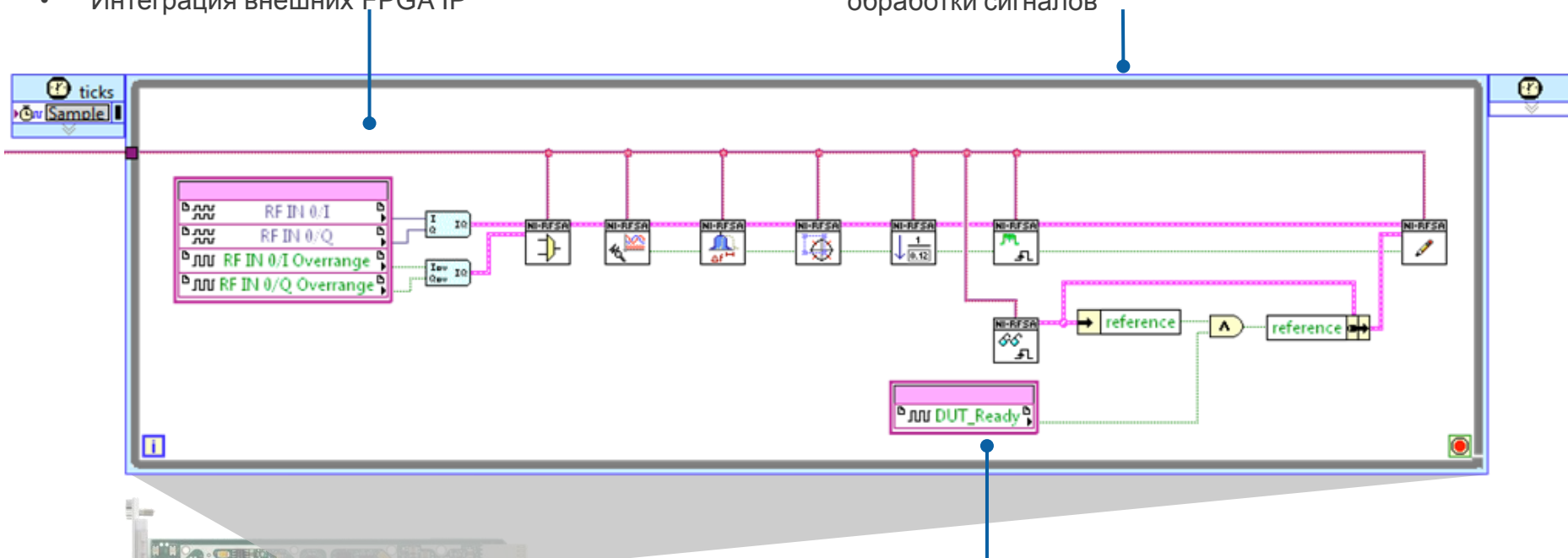
“Блок-диаграмма” = Код

Программирование с LabVIEW FPGA

- Элементы программирования, знакомые по LabVIEW
- Программирование, симуляция, отладка, компиляция и развертывание в LabVIEW
- Интеграция внешних FPGA IP

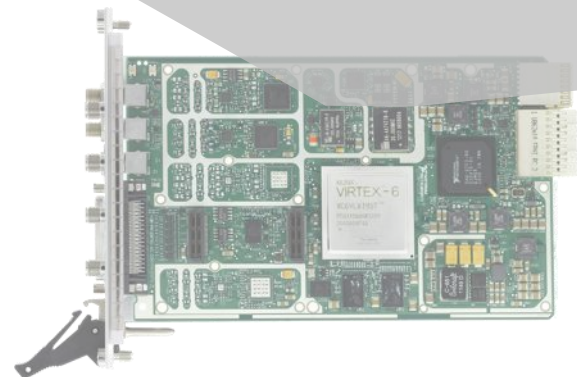
Высокоэффективные свойства

- Высокопроизводительные математические функции
- Улучшенное управление тактированием
- Доступ к оптимизированным ядрам цифровой обработки сигналов



Доступ к вводу-выводу и периферии

- Простые API для ввода-вывода на лицевую панель
- Высокая пропускная способность потокового обмена данными с хостом или другими устройствами PXI через PCI Express
- Чтение/запись в DRAM с произвольной адресацией



В программно проектируемых измерительных приборах FPGA, программируемые пользователем, предоставляют возможность :

1. Выполнять измерения и генерацию стимулирующих воздействия в FPGA
2. Тестировать в контуре с обратной связью или в соответствии с протоколом
3. Реализовать пользовательский запуск и сжатие данных
4. Детерминировано выполнять тесты и управлять тестируемым объектом
5. Учитывать индивидуальные особенности тестируемого устройства или специализированного приложения



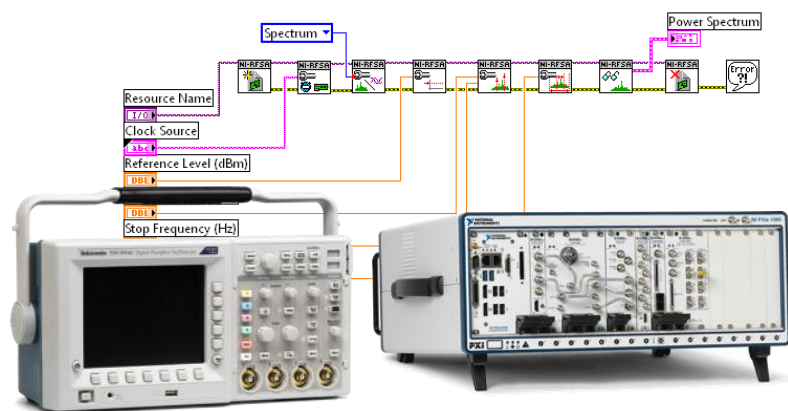
Наивысшая
производительность
тестирования

Перспективы многократного
использования аппаратных
средств

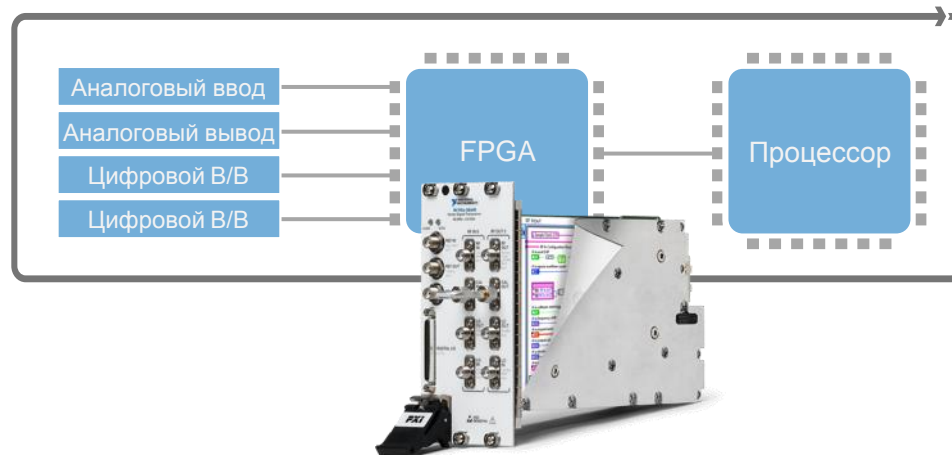
Новое, инновационное
тестирование

Минимальная стоимость тестирования

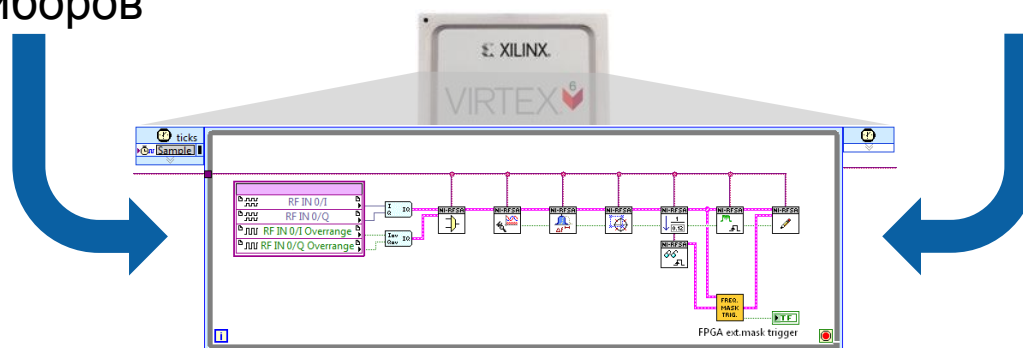
FPGA – расширения драйвера прибора



Совместимость с драйверами
стандартных промышленных
приборов



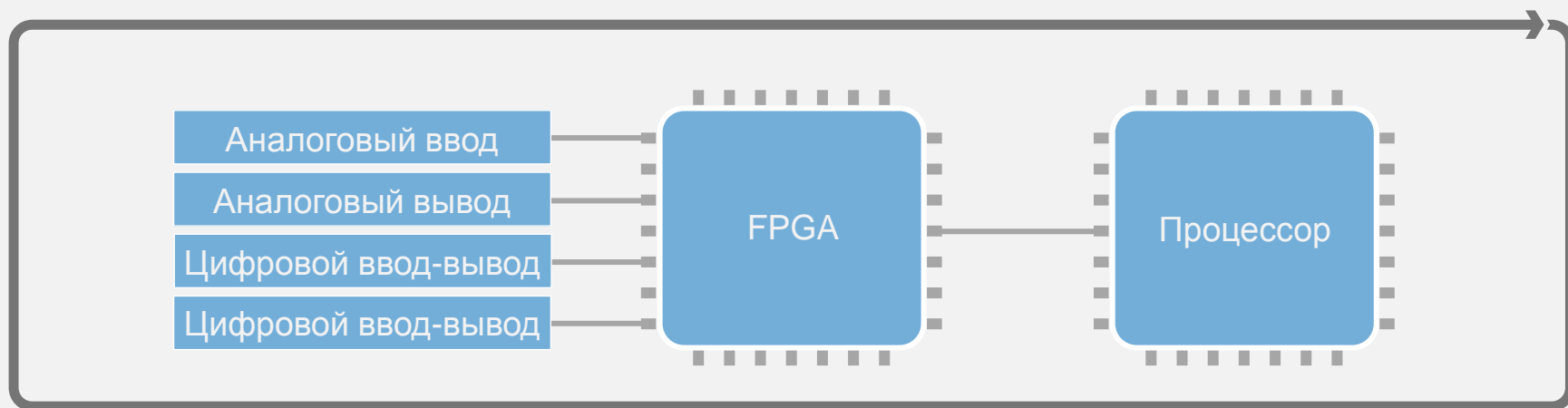
Гибкость архитектуры LabVIEW
RIO



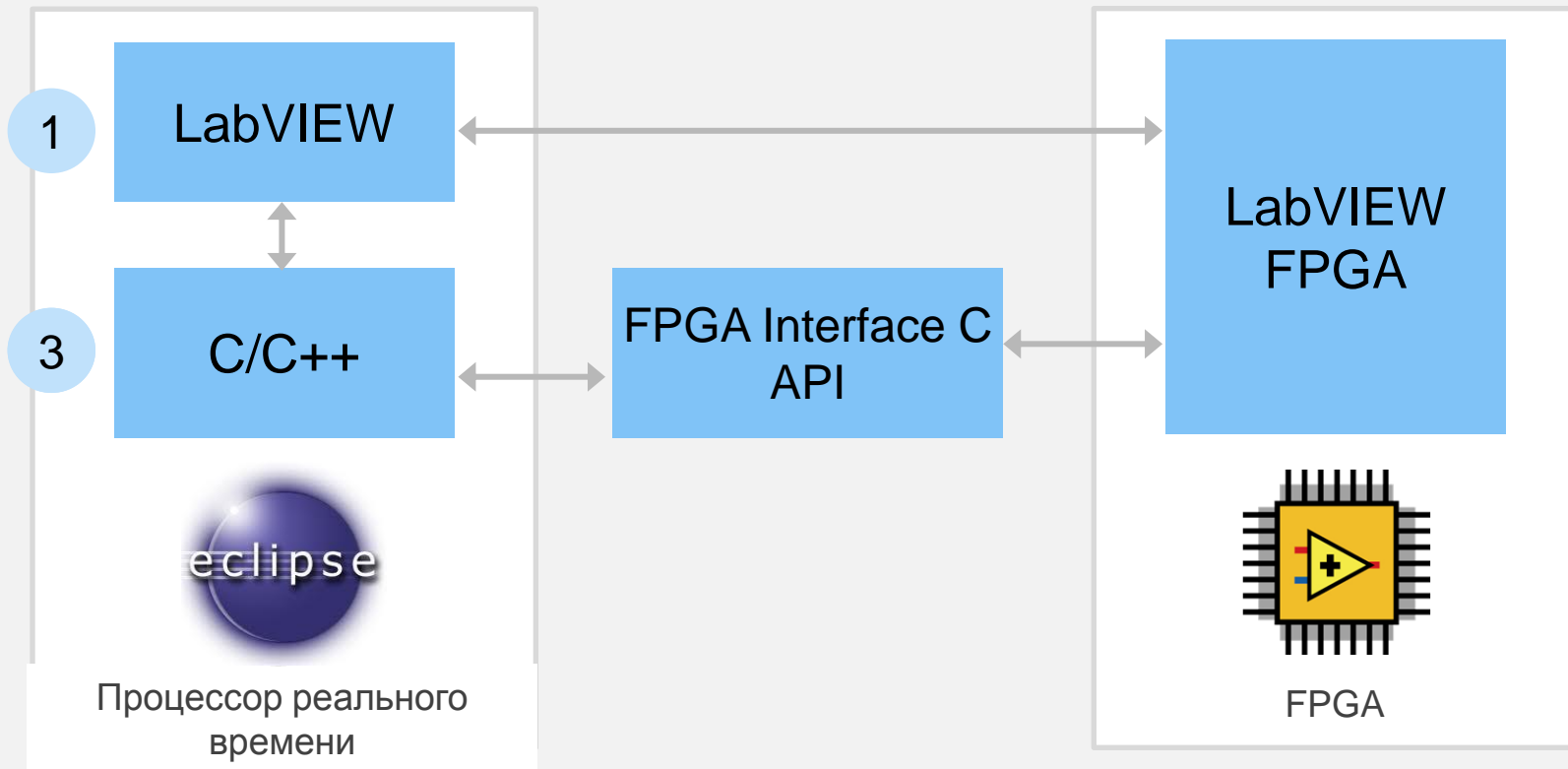
FPGA-расширения драйвера прибора

LabVIEW FPGA для встроенных систем

Архитектура NI LabVIEW RIO



NI Linux Real-Time: гибкая программная архитектура



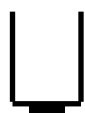
1 LabVIEW RT и FPGA

2 Приложение LabVIEW RT для ввода-вывода, вызов приложений или библиотек C/C++ app or library

3 C/C++ на процессоре RT, C API интерфейс для обмена данными между FPGA и LabVIEW FPGA

Демонстрация – измерение сигнала с датчика в LabVIEW FPGA

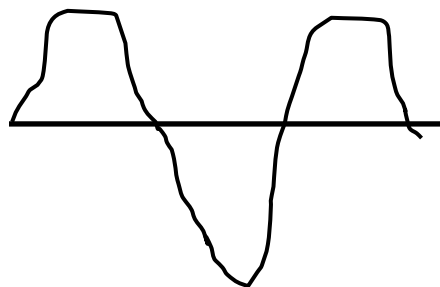
1



Датчик
приближения



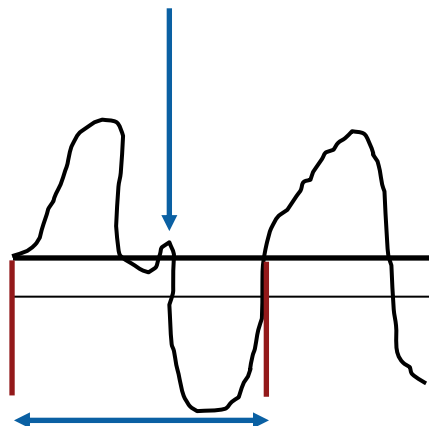
Ротор



Выходной
сигнал
датчика
тахометра

2

Порог отсечения
помех

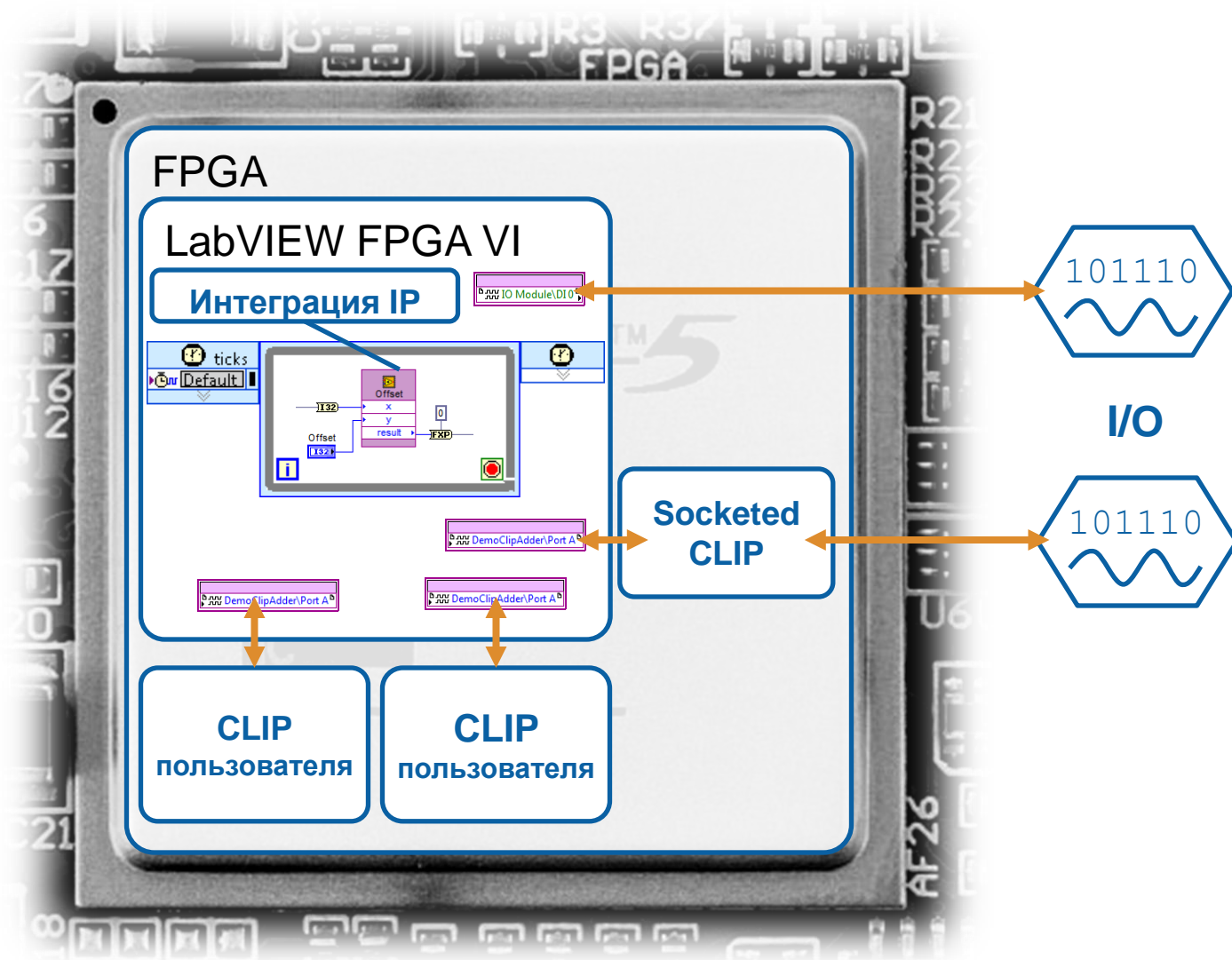


Время полного оборота
одного зубца датчика
приближения

3

Преобразование
периода (мкс/импульс) в
обороты/минуту

Механизмы интеграции HDL

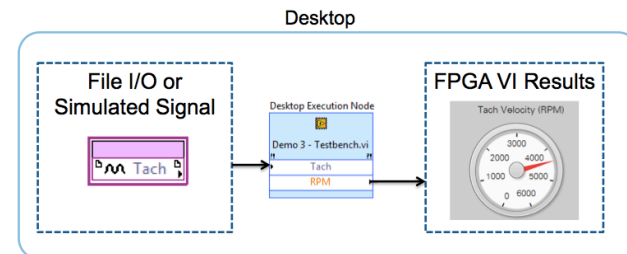


Работайте эффективнее с LabVIEW FPGA

Проверяйте быстрее

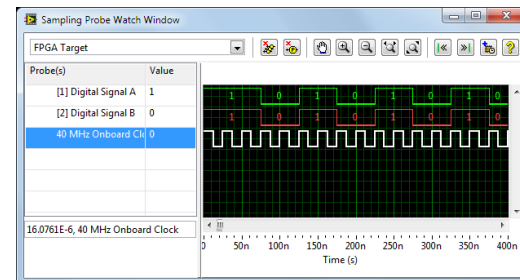
Проверяйте код, симулируя ввод-вывод

Используйте узел **Desktop Execution Node** для проверки кода путем создания тестовых средств и используя симулированные или генерируемые из файла данные ввода-вывода



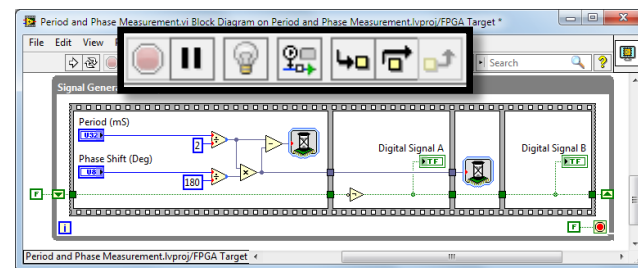
Проверяйте временные свойства сигнала с помощью пробника Waveform Probe

Используйте пробник **Digital Waveform Probe** для наблюдения одних сигналов относительно других и просмотра их предистории



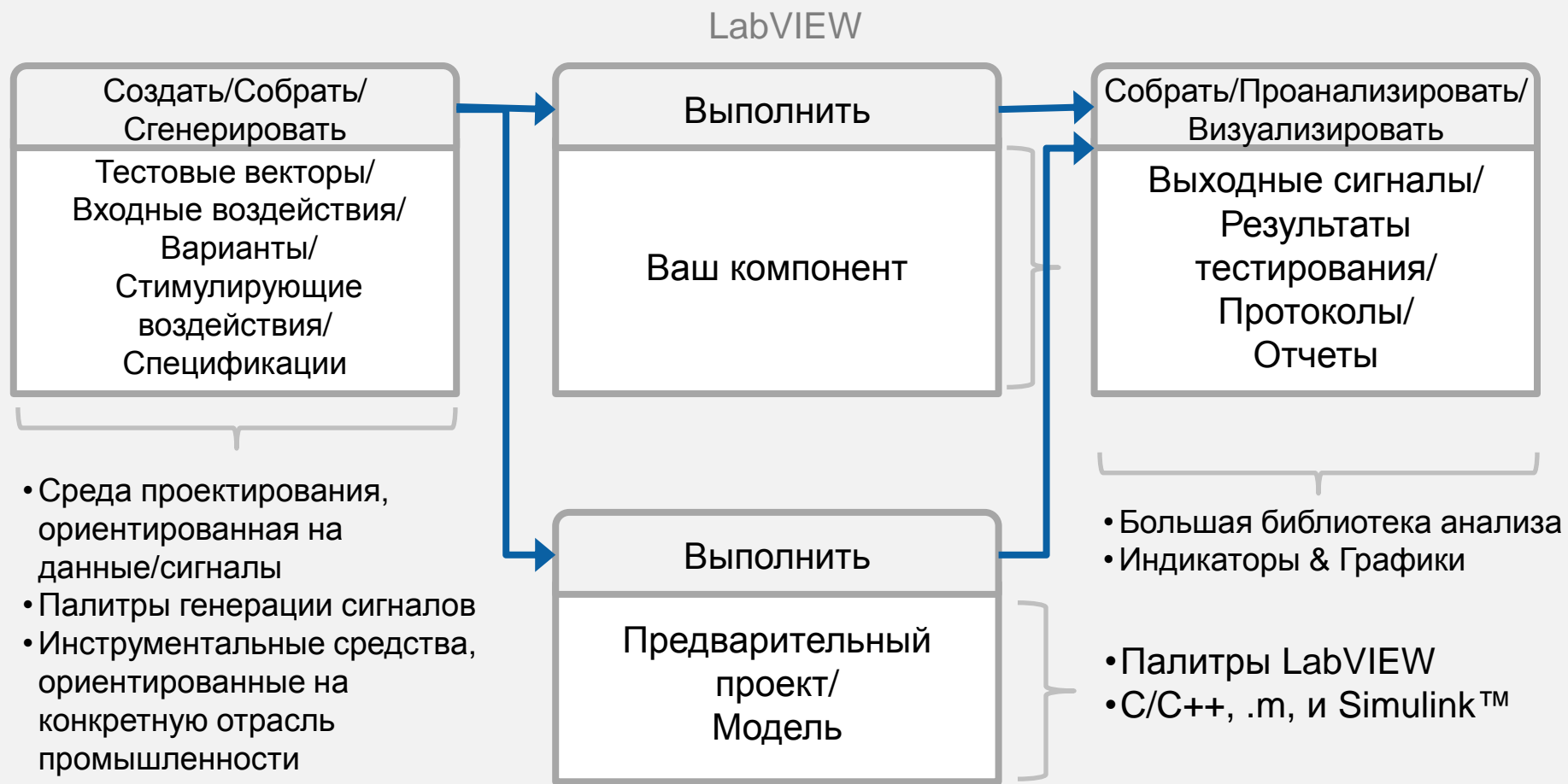
Выполняйте отладку, используя при симуляции стандартные средства LabVIEW

Подсветку выполнения, контрольные точки, пошаговое выполнит



Тестируйте ваши компоненты с помощью LabVIEW

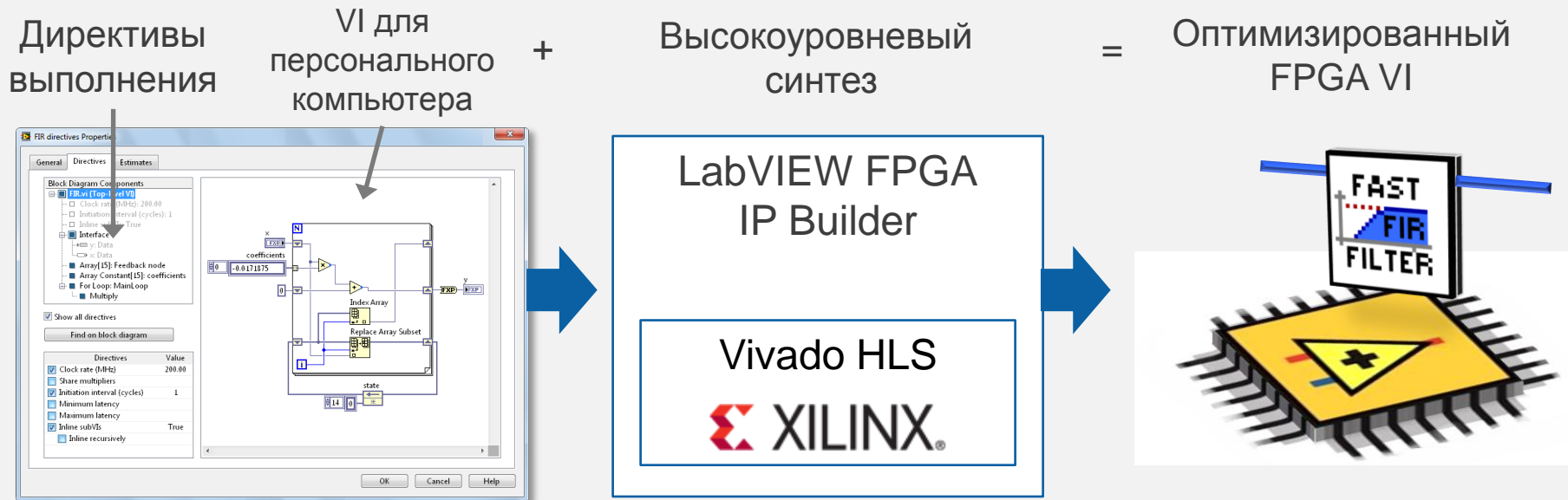
Создание, выполнение, анализ и представление результатов тестирования в одной среде



Высокоуровневый синтез FPGA

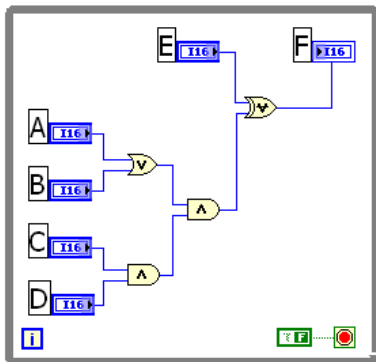
LabVIEW FPGA IP Builder

- Автоматическая оптимизация ваших LabVIEW FPGA VI
- Простое портирование кода LabVIEW персонального компьютера на FPGA
- Повторное использование ваших IP для решения различных проблем, модификация не требуется



Процесс компиляции

Код LabVIEW FPGA



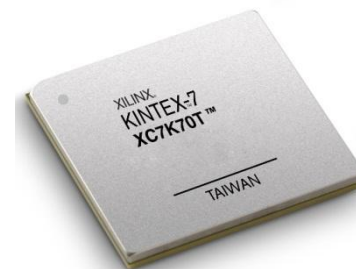
Компилятор Xilinx VHDL

```
-- VHDL code for the LabVIEW FPGA block diagram
-- process SynchronizationFFs;
-- Then we keep track of what the digital input was on the previous
-- clock cycle by inserting another flip flop
PreviousDigitalInputFF;
process( areset, Clk )
begin
    if areset then
        cPrevDigitalInput <= false;
    elsif rising_edge(Clk) then
        cPrevDigitalInput <= cdigitalInput;
    end if;
end process PreviousDigitalInputFF;

-- Then we have a little combinatorial logic to detect a rising edge
cRisingEdgeDetected <= cdigitalInput and not cPrevDigitalInput;

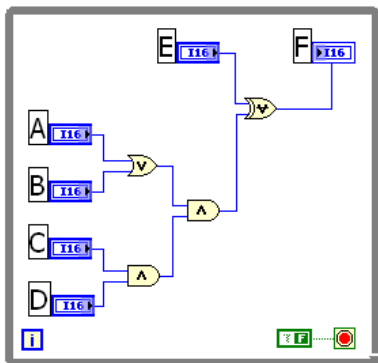
-- And finally we have a register that increments when that rising
-- edge is detected.
CounterRegister:
process( areset, Clk )
begin
    if areset then
        CounterRegister <= 0;
    elsif rising_edge(Clk) then
        CounterRegister <= CounterRegister + 1;
    end if;
end process CounterRegister;
```

Реализация логики в FPGA



Процесс компиляции

Код LabVIEW FPGA



Компилятор Xilinx VHDL

```
process SynchronizationFFs;
-- Then we keep track of what the digital input was on the previous
-- clock cycle by inserting another flip flop
PreviousDigitalInputFF;
process( areset, Clk )
begin
    if areset then
        cPrevDigitalInput <= false;
    elsif rising_edge(Clk) then
        cPrevDigitalInput <= cdigitalInput;
    end if;
end process PreviousDigitalInputFF;
-- Then we have a little combinatorial logic to detect a rising edge
cRisingEdgeDetected <= cdigitalInput and not cPrevDigitalInput;
-- And finally we have a register that increments when that rising
-- edge is detected.
CounterRegister:
process( areset, Clk )
begin
    if areset then
        CounterRegister <= 0;
    elsif rising_edge(Clk) then
        if cRisingEdgeDetected then
            CounterRegister <= CounterRegister + 1;
        end if;
    end if;
end process CounterRegister;
```

Реализация логики в FPGA



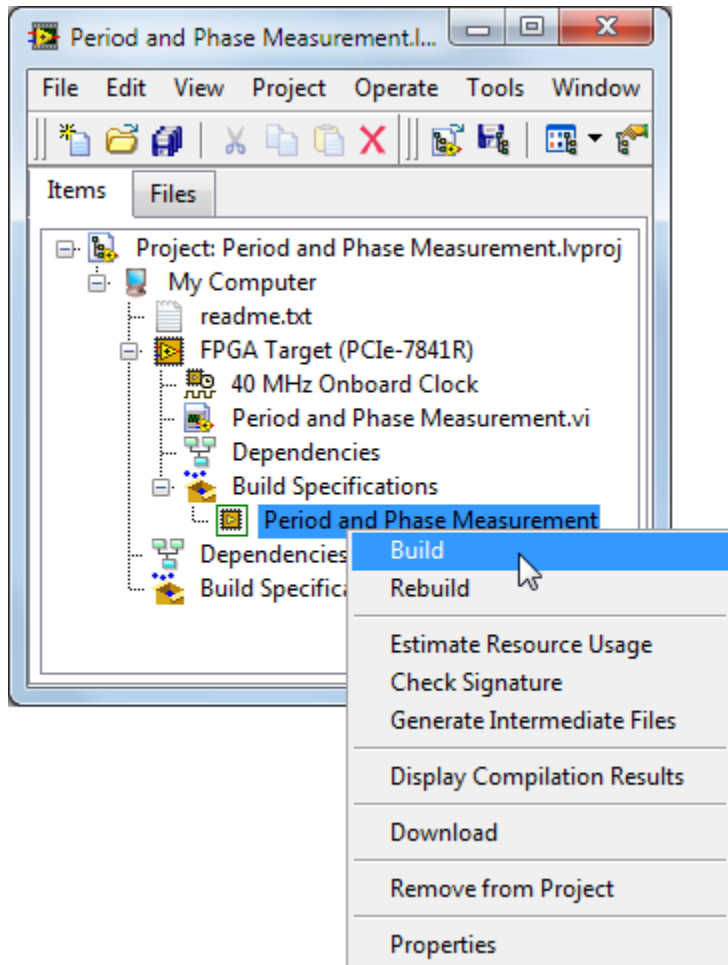
Трансляция
Генерация
кода VHDL

Оптимизация
Минимизация
логической
функции

Синтез
Размещение и
соединение
Проверка
временных
параметров

**Генерация
двоичного
кода**
Загрузка и
выполнение

Компиляция и развертывание одним щелчком



Компьютер
разработчика



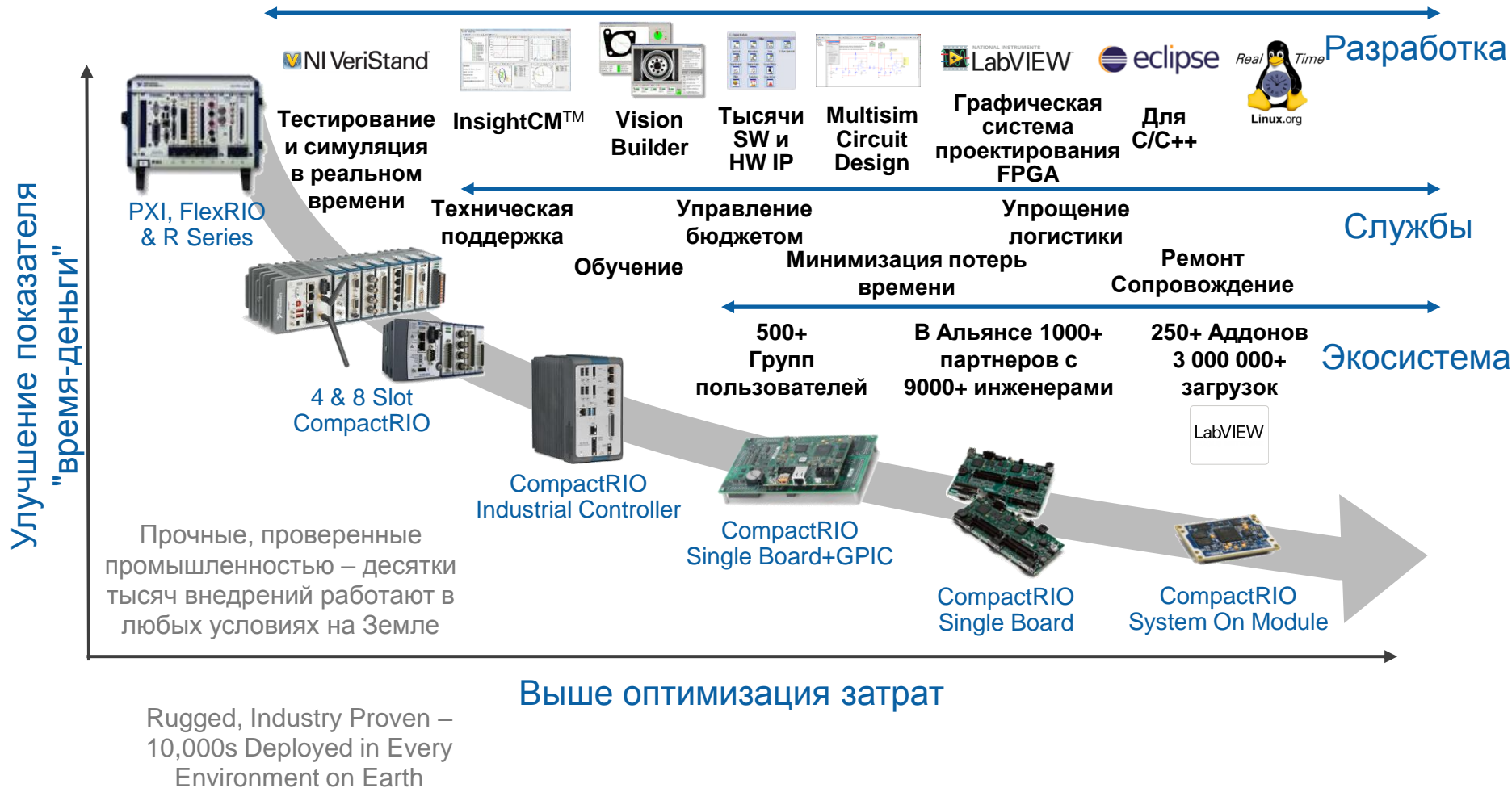
Сервер и
менеджеры
компиляции



Облачный сервис
высокой
производительности

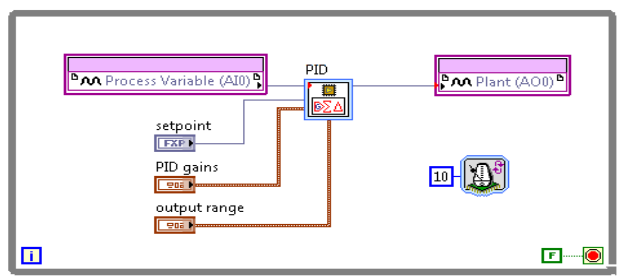


The NI Solution

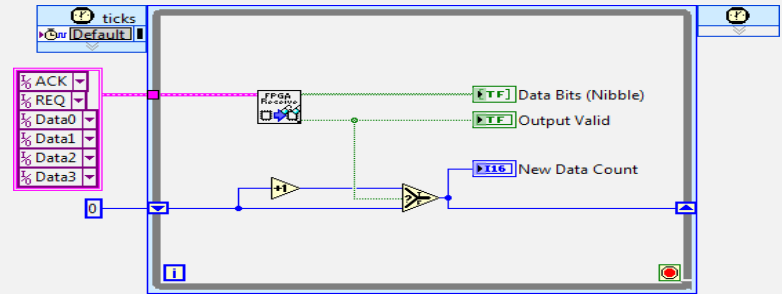


Модуль LabVIEW FPGA

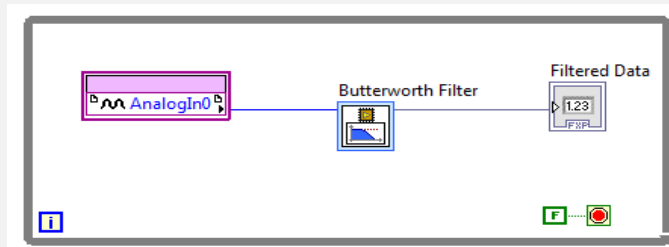
Разгружает самые критические части вашего приложения



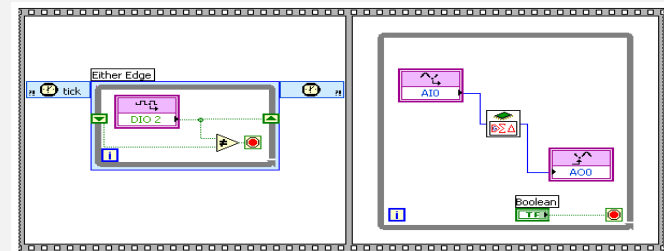
Высокоскоростное управление



Специализированные протоколы



Встроенная обработка
сигналов



Пользовательские тактирование,
запуск и синхронизация

Дополнительные ресурсы

ni.com/fpga

- Основы FPGA
- Изучить подробнее инструментальные средства
 - LabVIEW FPGA
 - Аппаратура на основе

Опробывание LabVIEW FPGA

- <http://www.ni.com/rioeval/>
- Варианты только для программиста и комплект разработчика

Руководство разработчика систем на основе CompactRIO

- <http://www.ni.com/compactriodevguide/>

Измерительные приборы, реализованные программно
<http://www.ni.com/software-designed-instruments/>

NI FPGA

