

## ВСТУПЛЕНИЕ

Методология сверхбыстрого проектирования представляет собой набор передовых методов, рекомендованных компанией Xilinx для максимального повышения производительности и сокращения количества выполняемых операций по проектированию сложных систем, включая встроенные процессорные подсистемы, модули аналоговой и цифровой обработки, высокоскоростной связи и сетевой обработки. Дополнительную информацию смотрите в Руководстве по методологии сверхбыстрого проектирования для Vivado Design Suite (UG949).

Контрольный список методологии проектирования UltraFast (ХТР301) содержит общие положения, касающиеся типичных областей, которые напрямую влияют на принимаемые проектные решения, а также акцентируют внимание на потенциальных проблемах, которые зачастую игнорируются или данные о которых отсутствуют. Он обеспечивает простой доступ к взаимосвязанным объектам. Контрольный список доступен так же через Xilinx Documentation Navigator (DocNav).

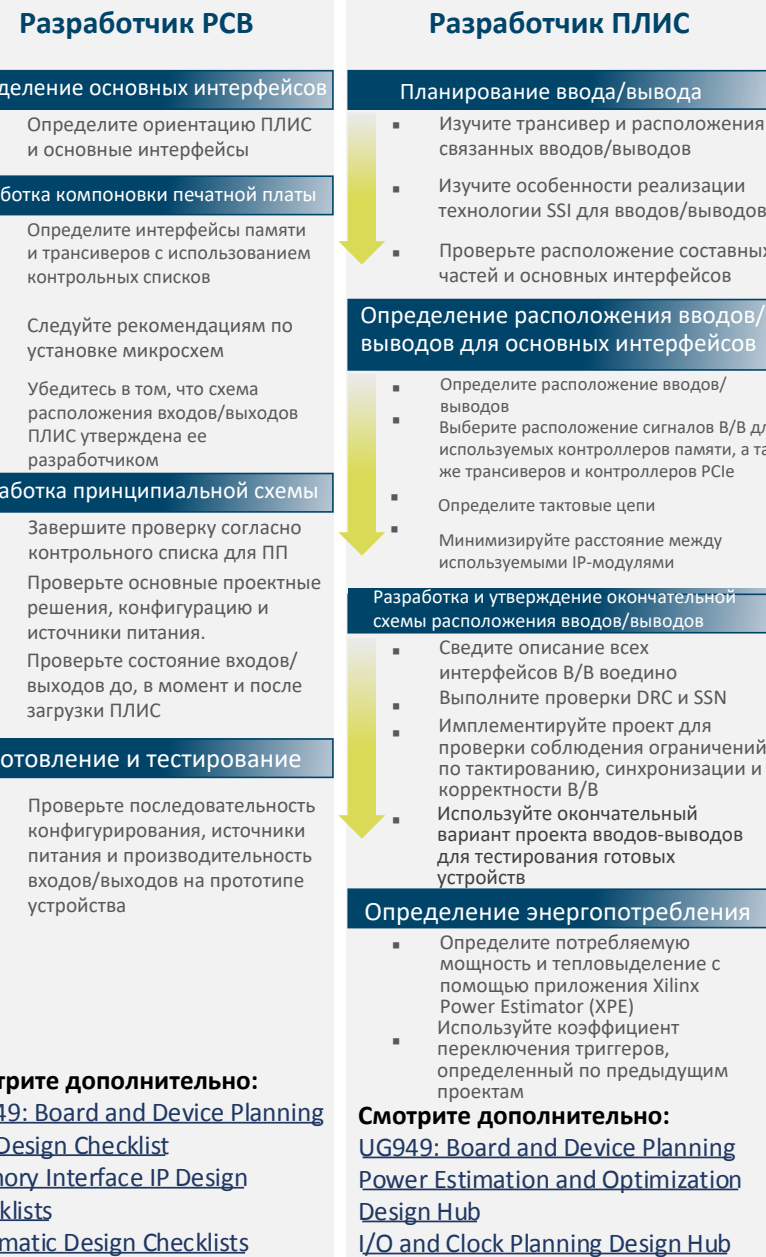
Данное краткое справочное руководство содержит основные методологии проектирования для ускорения процессов интеграции систем и реализации проектов, а также достижения максимальной эффективности от использования устройств и компонентов Xilinx®. Также здесь приведены ссылки на совместно используемые сторонние продукты. Указанные в этом руководстве основные задачи проектирования включают:

- Проектирование устройств и печатных плат
- Разработку функциональности устройства и его имплементацию
- Оценку качества проекта в целом
- Детальный анализ проекта
- Доработку проекта

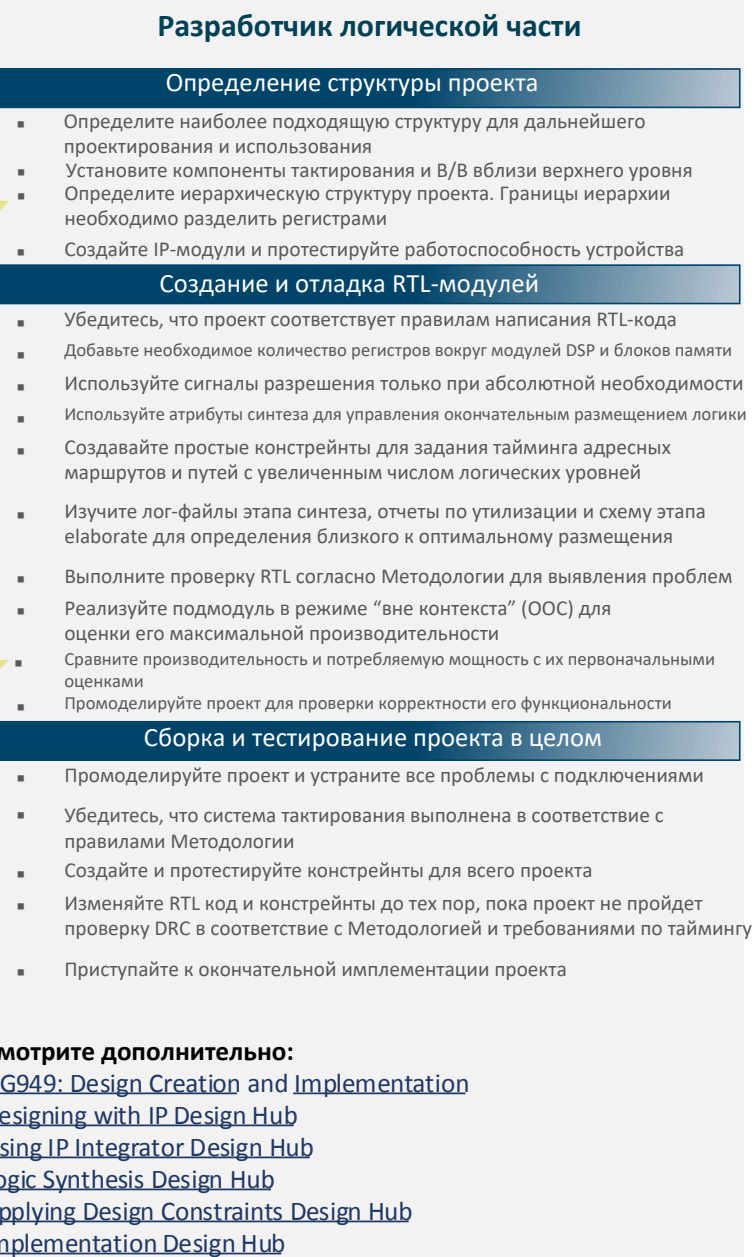
Для ознакомления с различными видами проектов и совместно используемыми устройствами перейдите в раздел "Проектирование на системном уровне" руководства "Методология сверхбыстрого проектирования", доступного в системе DocNav.



## РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ УСТРОЙСТВ И ПЛАТ



## РАЗРАБОТКА ФУНКЦИОНАЛЬНОСТИ И ИМПЛЕМЕНТАЦИЯ ПРОЕКТА





ПРОВЕРКА ПРОЕКТА НА СООТВЕТСТВИЯ ЗАДАНЫМ ТРЕБОВАНИЯМ (КОНСТРЕЙНТАМ)

Задание и проверка базовых констрейнтов

- Проверьте, выполняются ли требования по таймингу на начальных этапах проектирования после подключения большинства блоков и IP-модулей
- Задайте только основные констрейнты:
  - Все ограничения для IP-ядер
  - Задайте правдоподобные констрейнты для входных и сгенерированных тактовых сигналов
  - Задайте констрейнты, описывающие пересечение тактовых доменов
  - Где требуется, добавьте констрейнты для многоцикловых сигналов
  - Не задавайте констрейнтов для сигналов ввода/вывода на данном этапе
- Убедитесь в обоснованности установленных требований
- Проверьте выполнение условия  $WNS \approx 0.0$  на каждом этапе с помощью отчета “report\_timing\_summary”:
  - После синтеза
  - Перед размещением
  - До и после трассировки
- Определите причины нарушения тайминга на самых ранних стадиях
- Устраните проблемы тайминга в RTL-модулях и ошибки синтеза начиная с наиболее значимых

Проверка соответствия требованиям по таймингу

- Запустите отчеты “report\_timing\_summary”, либо “check\_timing”, чтобы удостовериться в том, что все тактовые частоты были определены, а констрейнты на регистры и входные/выходные порты были заданы
- Запустите отчет “report\_methodology” и выявите источники ошибок по таймингу в XDC-файле
- С помощью “report\_clock\_interaction” и проверьте полноту и корректность описания переходов между всеми парами тактовых сигналов
- Запустите “report\_cdc” для проверки констрейнтов для всех пересекающихся тактовых домены асинхронных сигналов и проверки безопасности используемой схемы синхронизации
- Запустите “report\_exceptions” для определения пересекающихся, игнорируемых или неэффективных исключений по таймингу
- Убедитесь в отсутствии критических предупреждений в загруженном проекте после наложения констрейнтов

Смотрите дополнительно:

[UG949: Design Closure](#)

- [Checking That Your Design is Properly Constrained](#)
- [Baselining The Design](#)

[Applying Design Constraints Design Hub](#)

[Timing Closure and Design Analysis Design Hub](#)

АНАЛИЗ ПРОЕКТА И ЗАВЕРШЕНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ

Определение источников нарушения тайминга

- Используйте “report\_qor\_suggestions” для автоматического анализа тайминга рекомендаций по достижению требуемых временных параметров
- Используйте “report\_timing\_summary” или “report\_design\_analysis” для поиска источников нарушения тайминга
- По задержкам типа setup проверьте:
  - Задержку в элементах (7-я серия > 25%, устройства UltraScale > 50%)
  - Задержку в путях (7-я серия > 75%, устройства UltraScale > 50%)
- Время удержания должно быть больше 0 нс
- Выполните проверку на превышение допустимых значений временного перекоса (>500 пс) и/или временной неопределенности (>200 пс) тактовых сигналов

Снижение задержек в логических элементах

- Измените RTL-код для использования параллельных и/или наиболее эффективных операторов
- Добавьте регистры для конвейеризации ретайминг при синтезе
- Добавьте регистры на выходах блоков DSP и блоках памяти
- Используйте директиву LUT\_REMAP для уменьшения количества логики на длинных путях
- Удалите сдвиговые регистры со входов/выходов
- Удалите из проекта директивы KEEP/DONT\_TOUCH/MARK\_DEBUG

Снижение задержек между элементами

- Проверьте констрейнты откорректируйте топологию проекта
- Оптимизируйте пути с большим фанаутом
- Найдите причины нехватки путей для трассировки, если уровень выше 4-го:
  - В таблице “placer congestion отчета” report\_design\_analysis”
  - В лог-файле с первоначальной оценкой разводки цепей

Снижение влияния временного перекоса

- Используйте параллельные буферы вместо каскадных
- Используйте CLOCK\_DELAY\_GROUP между синхронными тактовыми частотами от одного входа или PLL
- Установите исключения по таймингу между асинхронными тактовыми сигналами

Снижение влияния неопределенности тактовой частоты

- Оптимизируйте параметры для блоков MMCM
- Разделите тактовые сигналы с помощью BUFGE\_DIV в устройствах UltraScale™

Смотрите дополнительно:

[UG949: Design Closure](#)

- [Understanding Timing Reports](#)
- [Identifying Timing Violations Root Cause](#)
- [Timing Closure and Design Analysis Design Hub](#)

Снижение числа сигналов управления

- Избегайте задания MAX\_FANOUT в сигналах разрешения
- Увеличьте предел фанauta при синтезе сигналов разрешения
- Объединение эквивалентные управляющие сигналы с помощью команды opt\_design

Оптимизация цепи с большим фанаутом

- Используйте основанную на иерархической структуре проекта репликацию регистров в RTL-коде
- Используйте команду opt\_design-merge\_equivalent\_drivers-hier\_fanout\_limit для увеличения эффективности репликации
- Везде, где возможно, используйте глобальные тактовые сигналы
- Усиьте репликацию с помощью команды phys\_opt\_design

Устранение узких мест при разводке

- Уменьшите процент утилизации и сбалансируйте утилизацию SLR
- Попробуйте использовать директивы AltSpreadLogic\* и/или SSI\_Spread\*
- Определите испытывающие недостаток цепей для разводки модули командой report\_design\_analysis -complexity –congestion
- Для этих модулей используйте альтернативную стратегию синтеза на уровне блока, уменьшите MUX\*/CARRY\* использованием команды opt\_design или используйте директиву CELL\_BLOAT\_FACTOR
- Используйте глобальные тактовые сигналы для цепей с высоким фанаутом в “узких” местах
- Используйте схемы размещения DSP и блоков RAM из предыдущих успешных сборок с низкими показателями нагрузки

Настройка процесса компиляции

- Попробуйте использовать несколько директив place\_design
- Используйте стратегии блочного синтеза для построения оптимального нетлиста
- Задайте дополнительные требования для критических тактовых сигналов при размещении и физической оптимизации с помощью директивы set\_clock\_uncertainty
- Используйте инкрементную компиляцию при незначительных изменениях в проекте для сохранения параметра QoR и уменьшения времени выполнения

Анализ и оптимизация энергопотребления

- Установите ограничения для активности, окружения ПЛИС и выполняемых процессов
- Используйте команду power\_opt для снижения энергопотребления
- Максимально используйте каскадирование блоков памяти

Смотрите дополнительно:

[UG949: Implementation and Design Closure](#)

- [Analyzing and Resolving Timing Violations](#)
- [Applying Common Timing Closure Techniques](#)
- [Implementation Design Hub](#)
- [Timing Closure and Design Analysis Design Hub](#)