**1. Поколения интерфейсов ввода-вывода. Типы интерфейсов. Физическая реализация шин интерфейсов. Примеры.**

**Поколения интерфейсов ввода-вывода:** Интерфейсы ввода-вывода (I/O) претерпели несколько поколений развития, от ранних параллельных интерфейсов до современных высокоскоростных серийных интерфейсов. Основные поколения включают:

1. **Параллельные интерфейсы:**
   * **Пример:** PCI (Peripheral Component Interconnect)
   * **Физическая реализация:** Параллельные шины, многожильные кабели.
2. **Серийные интерфейсы:**
   * **Пример:** PCIe (Peripheral Component Interconnect Express)
   * **Физическая реализация:** Серийные шины, пара проводов (дифференциальные пары).
3. **Интерфейсы высокого уровня:**
   * **Пример:** USB, Thunderbolt
   * **Физическая реализация:** Комбинация высокоскоростных серийных интерфейсов и интегрированных контроллеров.
4. **Сетевые интерфейсы:**
   * **Пример:** Ethernet, Wi-Fi
   * **Физическая реализация:** Оптические волокна, коаксиальные кабели, радиочастотные модули.

**Типы интерфейсов:**

* **Параллельные:** многожильные соединения, которые передают данные одновременно по нескольким линиям (например, старый интерфейс IDE для жестких дисков).
* **Серийные:** передают данные последовательно по одной или нескольким линиям (например, SATA, USB, PCIe).
* **Сетевые:** используют протоколы передачи данных по сетям (например, Ethernet, Wi-Fi).

**Физическая реализация шин интерфейсов:**

* **Проводные:** медные кабели, оптические волокна.
* **Беспроводные:** радиочастотные модули, инфракрасные модули.

**Примеры:**

* **PCIe:** физически реализован через многослойные печатные платы с дифференциальными парами.
* **USB:** использует медные кабели с экранированием.
* **Ethernet:** может использовать как медные витые пары, так и оптические волокна.

**2. Этапы выполнения ОКР, использование и оформление интеллектуальной собственности на IP блоки в составе СБИС**

**Этапы выполнения ОКР (опытно-конструкторские работы):**

1. **Исследовательская фаза:**
   * Анализ требований.
   * Исследование существующих решений.
   * Формирование технического задания.
2. **Проектирование:**
   * Разработка архитектуры.
   * Схемотехническое проектирование.
   * Создание макета.
3. **Разработка и тестирование прототипа:**
   * Производство прототипов.
   * Верификация и тестирование.
   * Корректировка проекта.
4. **Подготовка к серийному производству:**
   * Оптимизация конструкции.
   * Подготовка производственной документации.
   * Пилотное производство.
5. **Серийное производство:**
   * Наладка серийного производства.
   * Контроль качества.
   * Постпродажное обслуживание.

**Использование и оформление интеллектуальной собственности на IP-блоки:**

* **Создание IP-блока:**
  + Разработка и тестирование функциональности.
  + Верификация соответствия спецификациям.
* **Защита IP-блока:**
  + Подача заявок на патенты.
  + Регистрация авторских прав и торговых марок.
* **Лицензирование IP-блока:**
  + Лицензирование использования другим разработчикам.
  + Составление соглашений о неразглашении и условиях использования.

**3. Технологические ограничения на примерах многослойных печатных плат. Тестирование и изготовление проекта.**

**Технологические ограничения многослойных печатных плат:**

* **Толщина и количество слоев:** ограничивается возможностями производства и требованиями проекта.
* **Материалы:** выбор материала для внутренних и внешних слоев влияет на электрические характеристики и стоимость.
* **Разводка сигналов:** плотность сигналов и минимальная ширина дорожек ограничиваются технологическими возможностями фрезеровки и травления.
* **Тепловыделение:** необходимость учета теплового режима работы компонентов и платы в целом.

**Тестирование и изготовление проекта:**

1. **Проектирование:**
   * Использование CAD-систем для разработки схем и топологии платы.
   * Верификация электрической схемы и трассировки.
2. **Изготовление:**
   * Подготовка гербер-файлов и производственной документации.
   * Производство платы (фрезеровка, травление, нанесение маски, пайка).
3. **Тестирование:**
   * Электрическое тестирование на целостность и короткие замыкания.
   * Функциональное тестирование собранной платы.

**4. Основные этапы проектирования СБИС, включая подготовку к производству и тестирование.**

**Этапы проектирования СБИС:**

1. **Техническое задание:**
   * Определение требований.
   * Разработка спецификаций.
2. **Разработка архитектуры:**
   * Создание архитектуры системы.
   * Моделирование и верификация архитектуры.
3. **Схемотехническое проектирование:**
   * Разработка логических схем.
   * Симуляция и верификация логики.
4. **Физическое проектирование:**
   * Размещение и трассировка.
   * Верификация размещения и трассировки.
5. **Подготовка к производству:**
   * Подготовка масок для фотолитографии.
   * Создание производственной документации.
6. **Производство:**
   * Фабрикация чипов.
   * Начальное тестирование на производстве.
7. **Тестирование:**
   * Функциональное тестирование.
   * Оптимизация и отладка.

**5. Обзор языков описания аппаратуры. Комбинационная логика. Логические оптимизации. Автоматическая генерация тестов.**

**Языки описания аппаратуры:**

* **VHDL (VHSIC Hardware Description Language):** используется для описания цифровых систем на уровне логических схем.
* **Verilog:** более лаконичный по сравнению с VHDL, широко используется в промышленности.
* **SystemVerilog:** расширение Verilog с поддержкой объектно-ориентированного программирования и улучшенными средствами верификации.

**Комбинационная логика:**

* Логические схемы, где выходные значения определяются только текущими входными значениями, без учета состояния.

**Логические оптимизации:**

* **Минимизация логики:** использование методов минимизации булевых функций (например, карт Карно).
* **Распараллеливание:** уменьшение задержек за счет параллельной обработки данных.

**Автоматическая генерация тестов:**

* Использование методов верификации, таких как симуляция Монте-Карло и формальные методы.
* Генерация тестовых векторов для проверки корректности работы схем.

**6. Физический синтез СБИС. Технологическое отображение и учет потребляемой мощности.**

**Физический синтез СБИС:**

* **Размещение и трассировка:** автоматическое размещение компонентов на кристалле и прокладка соединений между ними.
* **Учет физических ограничений:** соблюдение технологических норм и ограничений.
* **Оптимизация:** минимизация задержек, площади и потребляемой мощности.

**Технологическое отображение:**

* Преобразование логической схемы в физическую структуру, учитывая технологические ограничения производства.

**Учет потребляемой мощности:**

* Анализ потребления энергии на уровне логики и физического синтеза.
* Оптимизация схемы для снижения энергопотребления, например, через использование тактового понижения частоты и выключения неиспользуемых блоков.

**7. Планирование топологии проекта, включая критерии оптимизации и типичные алгоритмы реализации.**

**Планирование топологии:**

* **Цели:** минимизация площади, задержек, энергопотребления.
* **Алгоритмы:** методы размещения и трассировки, такие как алгоритм Стайнера для минимизации длины соединений.
* **Критерии оптимизации:** плотность размещения, длина соединений, балансировка нагрузки.

**8. Размещение компонентов с учетом технологических ограничений, включая критерии оптимизации и типичные алгоритмы реализации.**

**Размещение компонентов:**

* **Цели:** оптимальное расположение компонентов для минимизации задержек и площади.
* **Технологические ограничения:** минимальные расстояния между компонентами, ограничения по тепловыделению.
* **Алгоритмы:** алгоритмы размещения, такие как Force-Directed Placement, алгоритм Фиддлера для минимизации интерконнектов.

**9. Глобальная трассировка соединений проекта с учетом технологических ограничений, включая критерии оптимизации и типичные алгоритмы реализации.**

**Глобальная трассировка:**

* **Цели:** обеспечение правильного соединения всех компонентов.
* **Технологические ограничения:** минимальные и максимальные длины трасс, ограничения по электромагнитной совместимости.
* **Алгоритмы:** алгоритмы поиска кратчайших путей, такие как алгоритм A\*, алгоритмы минимального остовного дерева.

**10. Детальная трассировка соединений проекта, включая критерии оптимизации и типичные алгоритмы реализации.**

**Детальная трассировка:**

* **Цели:** точное определение маршрутов соединений между компонентами.
* **Критерии оптимизации:** минимизация перекрестных помех, оптимизация сигналов по задержке.
* **Алгоритмы:** алгоритмы детализации трасс, такие как алгоритм Ли, алгоритмы на основе сетевого анализа.

**11. Методы тестирования СБИС. Тесты на этапах проектирования.**

**Методы тестирования СБИС:**

* **Симуляция:** моделирование работы схемы на разных уровнях абстракции.
* **Формальная верификация:** доказательство корректности схемы математическими методами.
* **Тестирование на уровне схемы:** проверка логики и функциональности.

**Тесты на этапах проектирования:**

* **Логические тесты:** проверка правильности работы логических блоков.
* **Функциональные тесты:** проверка выполнения всех заданных функций.
* **Тесты на задержки:** измерение времени выполнения операций.

**12. Методы анализа партий микросхем, применение стендов для тестирования и разбраковки.**

**Методы анализа партий микросхем:**

* **Статистический контроль качества:** выборка и анализ партий микросхем.
* **Испытания на надёжность:** стресс-тесты, термоциклирование.

**Применение стендов для тестирования и разбраковки:**

* **Автоматические тестовые системы:** использование программируемых стендов для массового тестирования.
* **Разбраковка:** сортировка микросхем на годные и дефектные.

**13. Корпусирование СБИС с учетом ограничений, включая тенденции развития конструкторско-технологических решений.**

**Корпусирование СБИС:**

* **Цели:** защита кристалла, обеспечение теплового режима, электрическое соединение с печатной платой.
* **Ограничения:** размеры, тепловыделение, электрические параметры.

**Тенденции развития:**

* **Трехмерные упаковки (3D IC):** многослойные интегральные схемы.
* **Системы-на-кристалле (SoC):** интеграция различных функций на одном кристалле.

**14. Форматы для передачи топологической информации.**

**Форматы для передачи топологической информации:**

* **GDSII (Graphic Data System II):** стандартный формат для передачи данных о топологии интегральных схем.
* **OASIS (Open Artwork System Interchange Standard):** более компактный и эффективный формат по сравнению с GDSII.
* **LEF/DEF (Library Exchange Format/Design Exchange Format):** форматы для обмена данными о библиотеке ячеек и топологии проекта на уровне размещения и трассировки.

Промежуточный контроль проводится в виде дифференциального зачета. Перечень типовых вопросов:

**1. Целевой импеданс распределенной сети питания процессора. Влияния конденсаторов на стабильность напряжения питания.**

**Целевой импеданс распределенной сети питания процессора:**

Целевой импеданс распределенной сети питания (PDN, Power Distribution Network) процессора является критическим параметром, который определяет, насколько стабильно и эффективно будет передаваться питание к различным компонентам процессора. Низкий импеданс в сети питания необходим для минимизации колебаний напряжения, которые могут вызвать сбои в работе процессора. Обычно целевой импеданс определяется исходя из допустимых колебаний напряжения (обычно менее 5% от номинального) и величины потребляемого тока.

**Влияние конденсаторов на стабильность напряжения питания:**

Конденсаторы играют ключевую роль в поддержании стабильности напряжения питания. Они сглаживают колебания и пульсации напряжения, вызванные быстрыми изменениями нагрузки. Конденсаторы разделяются на несколько типов:

1. **Керамические конденсаторы:** имеют низкий эквивалентный последовательный резистор (ESR) и высокую частоту резонанса, что делает их эффективными в подавлении высокочастотных помех.
2. **Танталовые конденсаторы:** имеют высокую емкость на единицу объема и хорошую стабильность, но обладают более высоким ESR по сравнению с керамическими.
3. **Электролитические конденсаторы:** обладают высокой емкостью, но низкой частотой резонанса, что делает их полезными для сглаживания низкочастотных пульсаций.

При проектировании PDN важно правильно подобрать комбинацию различных типов конденсаторов для обеспечения эффективного подавления помех в широком диапазоне частот. Расположение конденсаторов также имеет значение: они должны быть как можно ближе к точкам потребления энергии для минимизации индуктивных потерь.

**2. Нормы проектирования схем с высокой плотностью трасс (HDI) по отношению к нормам проектирования многослойных печатных плат и проводников верхних слоев кристалла СБИС.**

**Нормы проектирования схем с высокой плотностью трасс (HDI):**

HDI (High-Density Interconnect) технологии позволяют проектировать печатные платы с высокой плотностью компонентов и межсоединений. Они включают использование микровиасов, тонких трасс и тонких изоляционных слоев. Основные нормы проектирования HDI включают:

1. **Микровиасы (Microvias):** используются для соединения между слоями. Диаметр микровиасов обычно составляет менее 150 мкм.
2. **Тонкие трассы и зазоры:** минимальная ширина трассы и зазора может быть менее 50 мкм.
3. **Слои материалов:** использование материалов с высокой диэлектрической постоянной для уменьшения перекрестных помех.

**Нормы проектирования многослойных печатных плат:**

В многослойных печатных платах используется большое количество слоев для распределения питания, земли и сигналов. Основные нормы включают:

1. **Минимальные размеры трасс и зазоров:** обычно не менее 100 мкм.
2. **Контроль импеданса:** критически важен для высокоскоростных сигналов.
3. **Размещение слоев питания и земли:** близость слоев питания и земли для создания встроенных конденсаторов, уменьшающих шумы.

**Нормы проектирования проводников верхних слоев кристалла СБИС:**

В СБИС (Системы на Больших Интегральных Схемах) используются специальные нормы для проектирования верхних слоев, где проводники имеют более низкое сопротивление и индуктивность. Основные нормы включают:

1. **Толщина и ширина проводников:** зависит от технологического процесса, обычно проводники верхних слоев шире и толще для уменьшения сопротивления.
2. **Контроль расстояний между проводниками:** критически важно для предотвращения коротких замыканий и перекрестных помех.
3. **Термо- и электромеханические свойства:** учитываются для обеспечения надежности работы кристалла.

**3. Типы корпусов микросхем и их выводов.**

**Типы корпусов микросхем:**

1. **DIP (Dual In-line Package):** традиционный корпус с двумя рядами выводов, используемый в простых и недорогих устройствах.
2. **SOIC (Small Outline IC):** уменьшенный корпус с двумя рядами выводов, предназначенный для поверхностного монтажа.
3. **QFP (Quad Flat Package):** корпус с выводами по периметру, используемый для сложных микросхем с большим количеством выводов.
4. **BGA (Ball Grid Array):** корпус с выводами в виде шариков на нижней поверхности, позволяющий размещать большое количество выводов на небольшой площади.
5. **CSP (Chip Scale Package):** корпус, размеры которого сравнимы с размерами самого кристалла, обеспечивающий минимальные размеры и высокий уровень интеграции.

**Типы выводов:**

1. **Выводы-ножки (Leads):** традиционные выводы в DIP и SOIC корпусах.
2. **Плоские выводы (Gull-wing Leads):** используются в корпусах типа QFP, обеспечивая легкий доступ для пайки.
3. **Шариковые выводы (Balls):** используются в корпусах типа BGA, обеспечивая плотное размещение выводов и улучшенные электрические характеристики.
4. **Площадочные выводы (Pads):** используются в CSP корпусах, обеспечивая минимальные размеры и улучшенные тепловые характеристики.

**4. Методы отжига при решении задач в САПР.**

**Методы отжига:**

Методы отжига, такие как симулированный отжиг (Simulated Annealing), применяются для решения задач оптимизации в системах автоматизированного проектирования (САПР). Этот метод основан на аналогии с физическим процессом отжига, при котором материал нагревается до высокой температуры, а затем медленно охлаждается, чтобы достигнуть состояния с минимальной энергией.

**Основные этапы симулированного отжига:**

1. **Начальная конфигурация:** определяется начальная конфигурация системы и начальная "температура".
2. **Переходы:** случайные изменения конфигурации с оценкой изменения "энергии" системы.
3. **Принятие решений:** изменения принимаются или отклоняются на основе вероятностной функции, зависящей от текущей температуры и изменения энергии.
4. **Охлаждение:** температура постепенно снижается, что уменьшает вероятность принятия менее оптимальных решений.

**Применение:**

Методы отжига используются для решения задач, связанных с размещением компонентов, трассировкой соединений и оптимизацией логики в САПР. Они эффективны для поиска глобальных минимумов в задачах с большим количеством локальных минимумов.

**5. Методы эволюционного программирования при решении задач САПР.**

**Методы эволюционного программирования:**

Эволюционные алгоритмы, такие как генетические алгоритмы, имитируют процессы естественного отбора и генетики для поиска оптимальных решений. Эти методы широко используются в САПР для решения сложных задач оптимизации.

**Основные этапы эволюционного программирования:**

1. **Инициализация:** создание начальной популяции решений.
2. **Оценка:** оценка качества каждого решения с использованием функции приспособленности.
3. **Селекция:** отбор лучших решений для создания новой популяции.
4. **Кроссинговер:** комбинирование частей решений для создания новых решений.
5. **Мутация:** случайные изменения в решениях для поддержания разнообразия популяции.
6. **Завершение:** процесс повторяется до достижения заданного критерия завершения (например, достижения оптимального решения или заданного числа итераций).

**Применение:**

Эволюционные алгоритмы применяются для задач, таких как оптимизация размещения компонентов, трассировка соединений и синтез логики. Они особенно эффективны для сложных задач с большим числом переменных и нелинейными зависимостями.

**6. Конфигурируемые логические блоки ПЛИС и соединения между ними.**

**Конфигурируемые логические блоки (CLBs):**

ПЛИС (FPGA, Field-Programmable Gate Array) состоят из конфигурируемых логических блоков (CLBs), которые являются основными элементами для построения логики. Каждый CLB включает в себя:

1. **Логические элементы (LEs):** базовые элементы, способные выполнять различные логические функции.
2. **Таблицы поиска (LUTs):** используются для реализации произвольных логических функций.
3. **Триггеры (Flip-flops):** позволяют реализовывать синхронные схемы и хранить состояние.

**Соединения между CLBs:**

Соединения между CLBs осуществляются через программируемую межсоединительную матрицу (Interconnect Matrix), которая включает:

1. **Прямые соединения:** короткие соединения между соседними CLBs.
2. **Длинные соединения:** соединения, охватывающие несколько CLBs, позволяющие передавать сигналы на большие расстояния.
3. **Коммутаторы (Switches):** программируемые коммутаторы, позволяющие настраивать маршруты сигналов в зависимости от конфигурации.

**Программируемые межсоединения:**

1. **Прямые линии (Direct Lines):** обеспечивают прямое соединение между соседними блоками.
2. **Локальные линии (Local Lines):** соединяют несколько соседних блоков внутри одной локальной области.
3. **Глобальные линии (Global Lines):** распространяют сигналы по всей ПЛИС для обеспечения синхронизации и передачи данных между отдаленными блоками.

**Применение:**

Конфигурируемые логические блоки и программируемые соединения позволяют ПЛИС реализовывать сложные цифровые схемы, такие как процессоры, контроллеры и интерфейсы, обеспечивая гибкость и возможность перепрограммирования для различных приложений.