АЛГОРИТМИЧЕСКИЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ КОНСТРУИРОВАНИЯ ПЕЧАТНЫХ ПЛАТ И МИКРОСХЕМ

к.т.н. Никаноров А.В.

Алгоритмические методы

1) Глобальные и локальные **критерии оптимизации в задачах конструирования**.

2) **Формализация представления** электрических принципиальных схем. Задание электрических схем графами, мульти-графами. Геометрическое, аналитическое, матричное представление графов.

3) **Методы решения задач компоновки**:

Классификация, сравнительная оценка, математическая постановка задач компоновки; Методы и средства описания монтажного пространства

4) **Алгоритмические методы размещения и трассировки**:

– решения задачи размещения модулей в монтажном пространстве

– основы методов решения задачи трассировки; способы и особенности трассировки проводных соединений; алгоритмы трассировки;

– Особенности трассировки соединений в двусторонних и многослойных печатных платах.

Порядок разработки ПП и СБИС

1. **Архитектура** – определение требований к функциональности.
2. **Модель** – высокоуровневое описание реализации функциональности
3. **Partitioning** – **Декомпозиция** (разбиение)
4. **Floor-planning** – **Поуровневое планирование**
   Разработка общей топологии. Один из этапов проектирования многослойной печатной платы или микросхемы - определение примерного размещения отдельных компонентов и/или групп компонентов по слоям или на площади кристалла.

3. Placement – **Размещение, расположение**

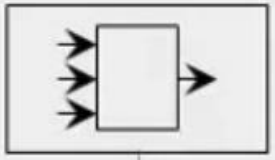
4. Routing – **Трассировка, маршрутизация проводников**

Физический дизайн

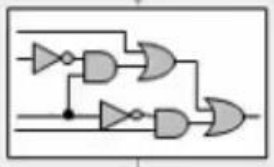
DFT – Design for testability. Тест работоспособности чипа, а не его функциональности.

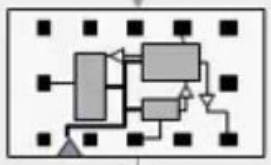
Разделы 1 и 2 выходят за рамки рассмотрения. Остальные разделы относятся к физическому дизайну и определяет свое множество алгоритмических методов, помогающих достигнуть цели.

Порядок разработки ПП и СБИС

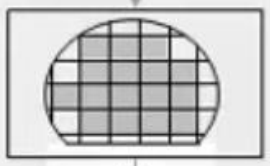


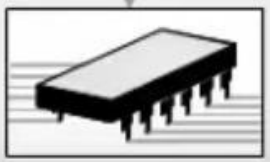
ENTITY test is
port a: in bit;
end ENTITY test;





DRC
LVS
ERC





System Specification

Architectural Design

Functional Design
and Logic Design

Circuit Design

Physical Design

Physical Verification
and Signoff

Fabrication

Packaging and Testing

Chip

Partitioning

Floor Planning
Pin Assignment
Power Planning

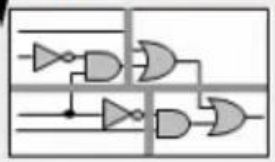
Placement

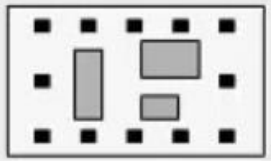
Clock Tree Synthesis

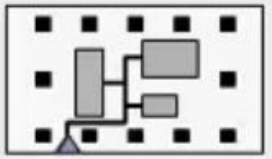
Signal Routing

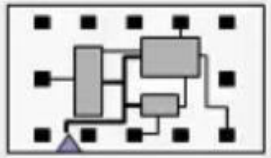
Timing Closure

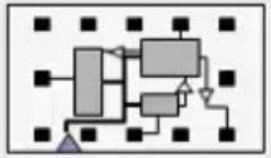
Chip Planning





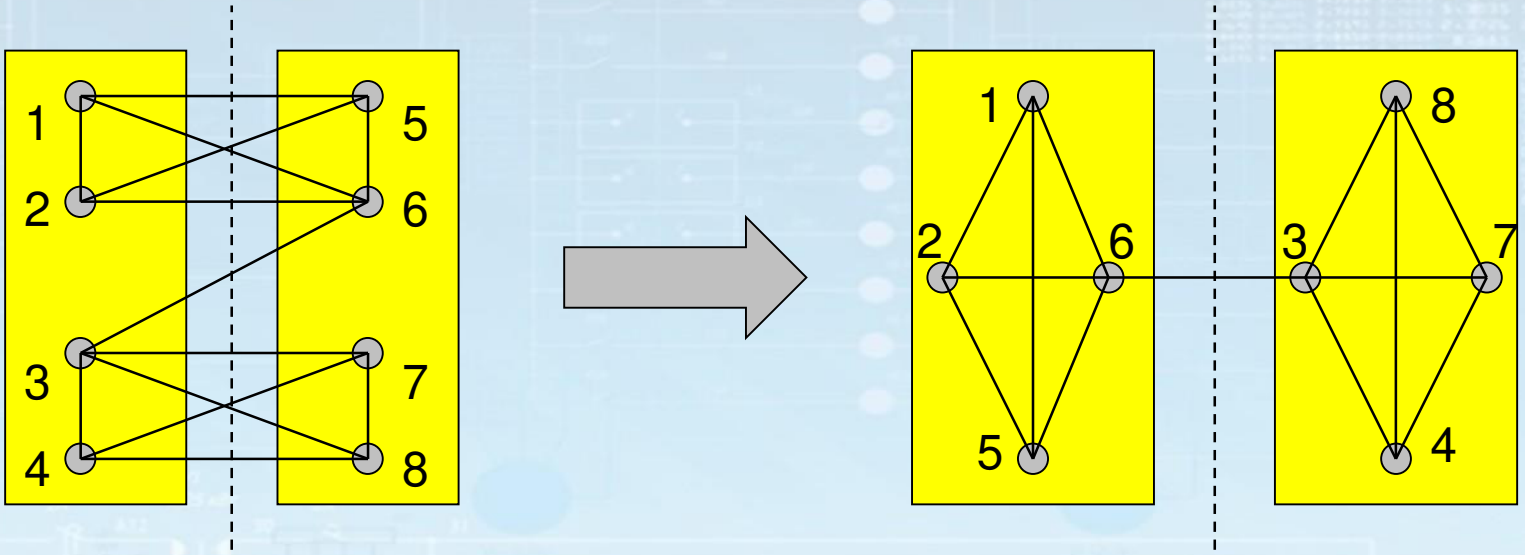






Partitioning - Декомпозиция

Пример алгоритм деления графа пополам



Начальное состояние
Количество связей = 9

Конечное состояние
Количество связей = 1

Алгоритм Кернигана/Лин (**Kernighan-Lin Algorithm**).

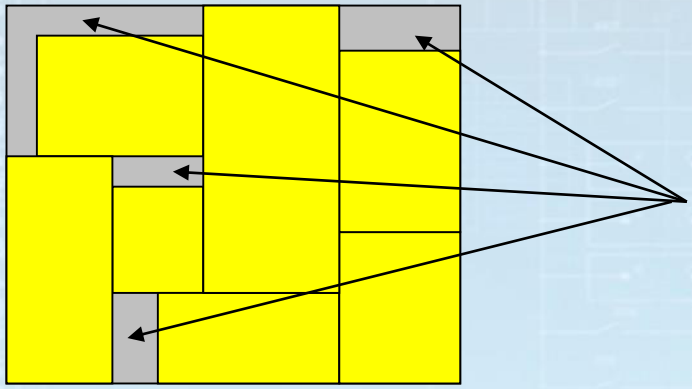
Это один из известных и интересных методов декомпозиции, но это не единственный алгоритм, есть и множество других хороших.

Результаты этапа Декомпозиции

- Площадь, занимаемая каждым блоком может быть оценена.
- Возможные формы блоков могут быть установлены.
- Количество элементов, входящий в каждый блок известно.
- Netlist (список электрических соединений) с указанием связей между блоками доступен.

Floor-Planning – поуровневое планирование

Область

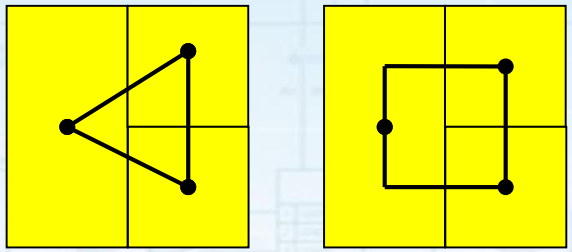


Алгоритмы сведения
к минимуму площади =
Минимум мертвой площади

Мертвое пространство
(deadspace)

Оценка длины проводника

- Точная длина проводника не известна пока не выполнена трассировка
- Положение вывода неизвестно.



Как оценить?

- Оценивается от центра до центра

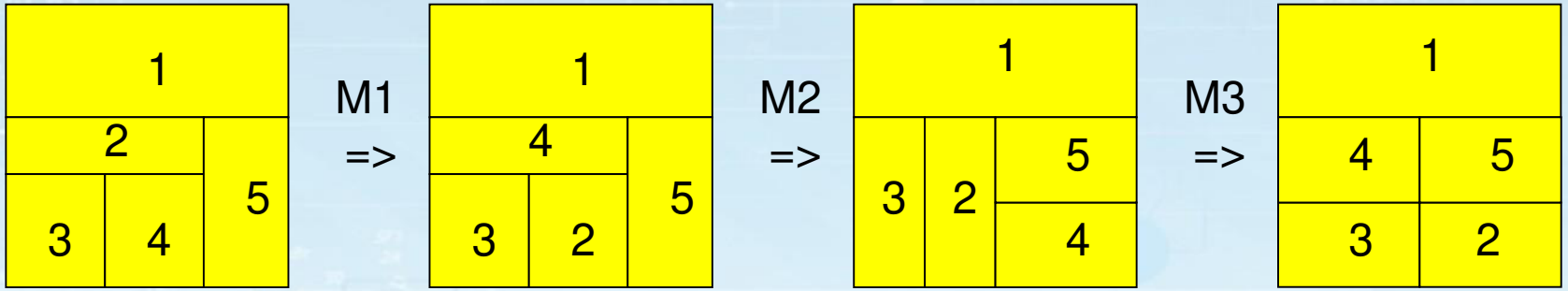
Floor-planning. Простое перемещение

Попытка оптимизировать издержки за счет перемещения блоков вокруг.

M1 : Поменять местами два смежных (соседних) операнда.

M2 : Дополнить соседние блоки по горизонтали и вертикали.

M3 : Поменять смежные два операнда или оператора.



3**4**V**2**H5V1H 32V4**H5**V1H 32V45**HV**1H 32V45VH1H

Цел достигнута, когда все логические ячейки устроены в пределах гибких блоков. Это поможет маршрутизатору завершить операцию соединений.

- Обеспечить минимум всех критических чистых задержек.
- Обеспечить максимальную плотность чипа.
- Обеспечить минимум рассеиваемой мощности.
  Обеспечить минимум перекрестных помех между сигналами.

Placement - Размещение

- Задачи этапа - создание маршрутизируемого дизайна с оптимизацией:
  - временных задержек;
  - занимаемой площади;
  - потребляемой мощности.
- Для вычислительной техники больше 25% цепей критичны к задержкам в линии. Необходимо группировать линии и оптимизировать задержки для обеспечения синхронной работы.

Routing - Маршрутизация

Маршрутизация - Routing

- Соединяет выводы и проводники
- Решение NP-задачи (**MILP->NP**)

Ограничения - Constraints

- Минимизировать общую длину проводника
- Минимизировать перегибы в пути
- Бюджет по времени

Дерево Штейнера -Steiner Tree

- Путь маршрутизации является оптимальным для Steiner Tree (кратчайшей сети, соединяющей заданный конечный набор точек плоскости)
- Нахождение Steiner Tree как NP-завершенную задачу

Типы

- Маршрутизация Мазе – Lee Algorithm
- Маршрутизация канала - Channel routing
- Маршрутизация переключаемых блоков Switchbox routing
- Маршрутизация потока (для однослойного уровня)

Глобальная маршрутизация

Не делает никаких соединений, только планирует.

- Решает, какие каналы будут использоваться для маршрутизации
- Помечает каналы с проводником для плотности размещения

Детальная маршрутизация

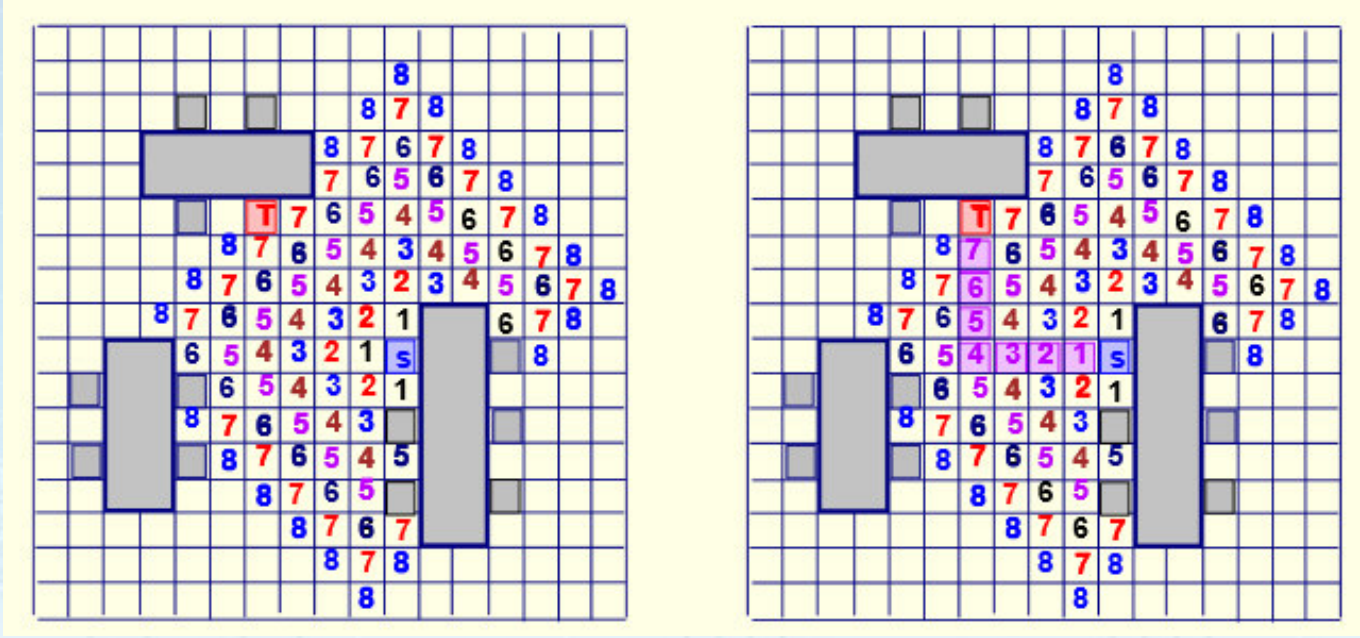
- Создает выводы и переходные отверстия.

Специальная маршрутизация

- Создает дерево тактового сигнала
- Питание и землю ведет раздельными сетями

Алгоритм волновой трассировки (алгоритм Ли)

Найти путь от S до T при помощи ”распространения волны”



Вычислительная сложность алгоритма на сетке = o(M*N) - огромна

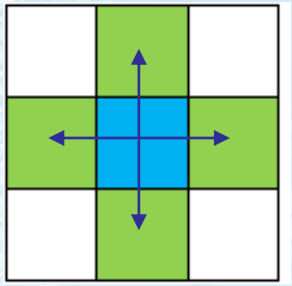
Описание алгоритма Ли

Алгоритм Ли (1961)—алгоритм поиска кратчайшего пути на планарном графе.

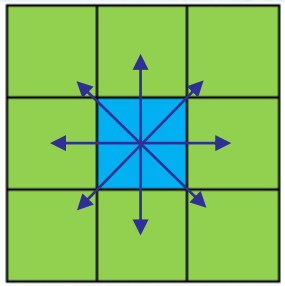
В основном используется при компьютерной трассировке (разводке) печатных плат, соединительных проводников микросхем.

Соседи ячейки в случае:

- *ортогонального (**окрестности фон Неймана**)*



- *ортогонально-диагонального путей (**окрестности Мура**)*



9	10		10	9	8	9	10	11	12	13	14
8	9		9	8	7	8	9	10	11	12	13
7	8	9	8	7	6	7	8	9	10	11	12
6	7	8	7	6	5	6	7			10	11
5					4	5	6	7	8	9	10
4	3	2	1	2	3	4	5	6			11
3	2	1	0	1	2	3	4	5			10
4	3	2	1	2	3	4	5	6	7	8	9

Результат работы волнового алгоритма (ортогональный путь)

7	7		6	6	6	6	6	6	6	7	8
6	6		5	5	5	5	5	5	6	7	8
5	5	5	4	4	4	4	4	5	6	7	8
4	4	5	4	3	3	3	4			7	8
3					2	3	4	5	6	7	8
3	2	1	1	1	2	3	4	5			8
3	2	1	0	1	2	3	4	5			8
3	2	1	1	1	2	3	4	5	6	7	8

Результат работы волнового алгоритма (ортогонально-диагональный путь)

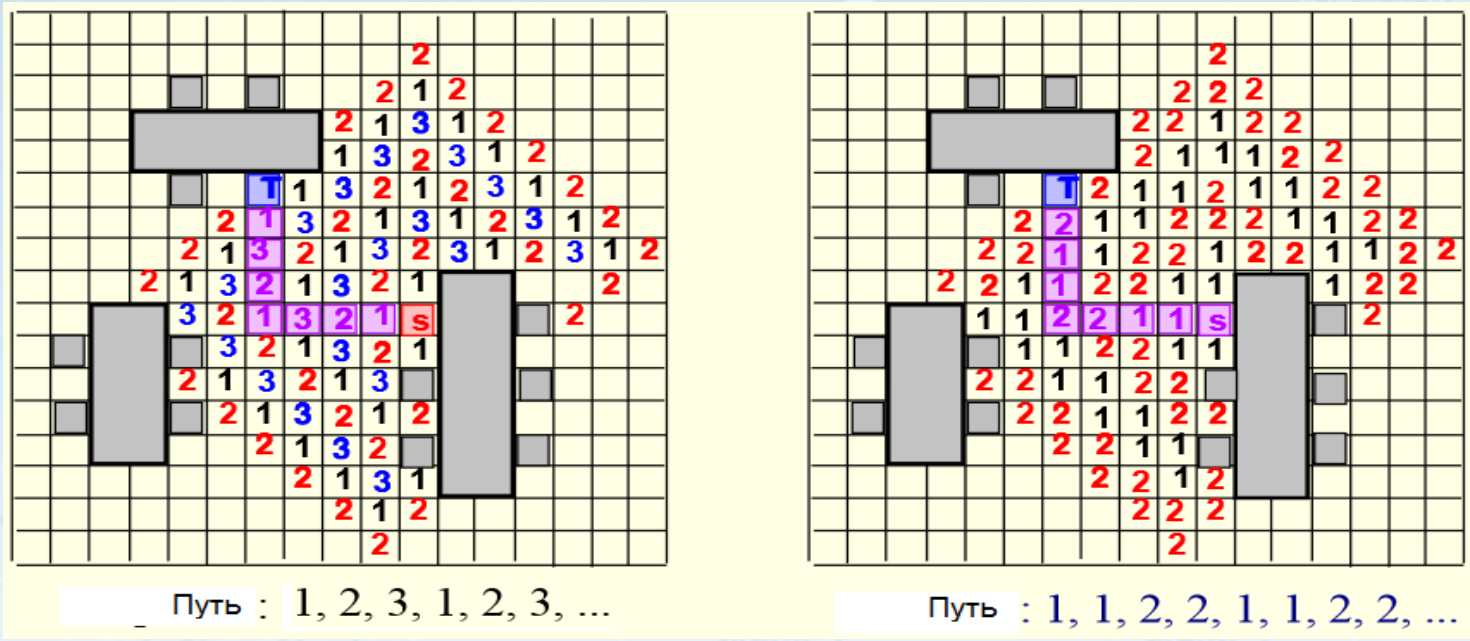
Модификация алгоритма Ли. Метод встречной волны

- Источниками являются обе ячейки, подлежащие электрическому соединению.
- Процесс продолжается до тех пор, пока фронт первой и второй волны не пересекутся.
- Проведение пути от данной ячейки до исходной – по правилам алгоритма Ли.

Преимущество – сокращение вдвое просматриваемой площади, а, соответственно, и времени трассировки.

Алгоритм волновой трассировки алгоритм Эйкерс – 1967

- Метки рассполагаются рядом от k до k+1.
- Схема волны маркируется таким образом, чтобы каждая текущая метка отличалась бы от ее последующей метки



Способ 1. Кодирование последовательности от S: 1,2,3, 1,2,3, пустой блок T, далее заблокировано. Под номер требуется **3 бита.**

Способ 2. Кодирование последовательности от S: 1,1,2,2,1,1,2,2, пустой блок T, далее заблокировано. Под номер требуется **2 бита.**

ОСНОВНАЯ ЦЕЛЬ: сокращение разрядности

Методология разработки СБИС

иерархии

Проектирование на алгоритмическом, функциональном и логическом уровне
Front-end Design

Разработка топологии
Back-end Design

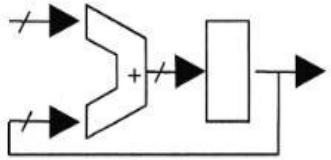
4 — Алгоритмический уровень
система в целом

Целевая комплексная система

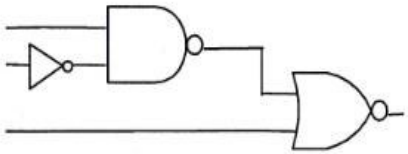
```
main (…)
{
  . . .
  d = a + b * c;
  . . .
  y= cos(x);
  . . .
}
```

3 — Функциональный уровень

Уровень функциональных блоком и структур

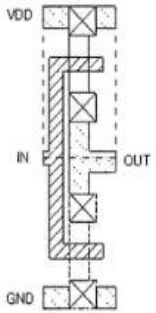


2 — Логический уровень



Построение алгоритмов

1 — Уровень схемы физической топологии



Структурный макет топологии

Структурная логическая схема

(Logic Gate, CELL)

Построение структур из функциональных блоков

(RTL модель)

Поведенческая программа C/C++ (MatLab LabView или др.)

1 2 3 4

Уровень абстракции

Задачи групп разработки СБИС

Front-end Design

Задачи команды логического проектирования:

- Логическое проектирование

  Logic design (**RTL** - Register Transfer Level)

- Синтез логики (**Synthesis**)

- Программа тестовых испытаний (**Test Bench**)

- Планирование примерного размещения (**floor-planning**)

Back-end Design

Задачи команды физического проектирования:

- Декомпозиция (**Placement**)

- Трассировка (**Routing**)

- Физическая проверка (**Physical verification**)

- **GDS II**

GDS II (GDSII, **GDS**, *Graphic Database System*) — формат файлов баз данных, являющийся де-факто промышленным стандартом для обмена данными по интегральным схемам и их топологиям. Данный формат описывает плоские геометрические формы.

GDS II

- **GDS II** (*Graphic Database System*) — формат файлов баз данных, де-факто, промышленный стандарт для обмена данными по интегральным схемам и их топологиям. Данный формат описывает плоские геометрические формы, текстовые метки и иную информацию в иерархической форме. Данные могут использоваться для обмена между различными САПР или для создания фотошаблонов.

- 3D-модель ячейки в формате GDSII с тремя слоями металла, без диэлектрика.

- Желтые структуры — межсоединения из металла.
  Красные — затворы транзисторов.
  В нижней части — подложка.