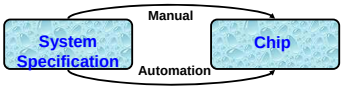
21. Circuit Optimizations

- Previous Unit:
  - Packaging
  - I/O
- This Unit:
  - Circuit optimization overview
  - Various techniques
    - Partitioning, placement, routing
    - Gate sizing, wire sizing, buffer insertion

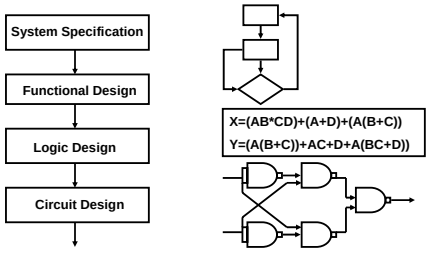
VLSI Design Cycle



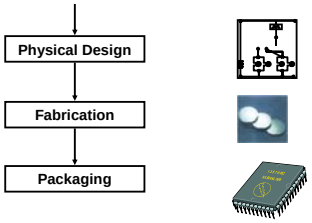
- **Large number of devices**
- **Time-to-market competition**
- **Power (and other) constraints**

Optimizations are everywhere

VLSI Design Cycle



VLSI Design Cycle (cont.)



Physical Design

Physical design converts a circuit description into a geometric description. This description is used to manufacture a chip. The physical design cycle consists of

1 Partitioning

2 Floorplanning and Placement

3 Routing

Traditional View

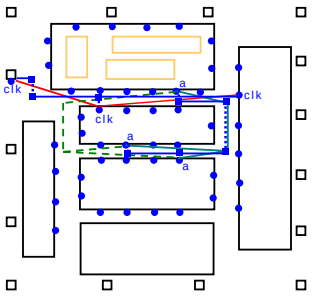
Physical Design Process

Design Steps:

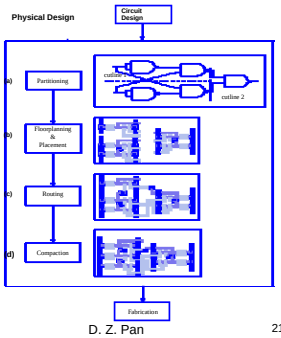
Partition & Clustering
Floorplan & Placement
Pin Assignment
Global Routing
Detailed Routing

Methodology:

Divide-and-Conquer



Physical Design Cycle

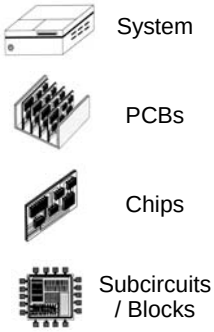


Levels of Partitioning

System Level Partitioning

Board Level Partitioning

Chip Level Partitioning



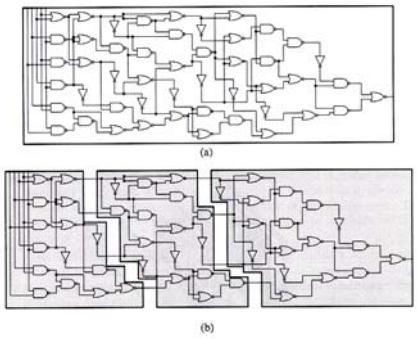
System

PCBs

Chips

Subcircuits / Blocks

Partitioning of a Circuit



Importance of Circuit Partitioning

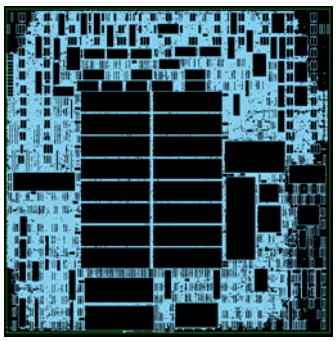
- **Divide-and-conquer methodology**
  - **The most effective way to solve problems of high complexity**
  - **E.g.: min-cut based placement, partitioning-based test generation,…**
- **System-level partitioning for multi-chip designs**
  - **inter-chip interconnection delay dominates system performance.**
- **Circuit emulation/parallel simulation**
  - **partition large circuit into multiple FPGAs (e.g. Quickturn), or multiple special-purpose processors (e.g. Zycad).**
- **Parallel CAD development**
  - **Task decomposition and load balancing**
- **In deep-submicron designs, partitioning defines local and global interconnect, and has significant impact on circuit performance**

…… ……

Placement

- Input:
  - Blocks (standard cells and macros) B_1, … , B_n
  - Shapes and Pin Positions for each block B_i
  - Nets N_1, … , N_m
- Output:
  - Coordinates (x_i , y_i) for block B_i.
  - No overlaps between blocks
  - The total wire length is minimized
  - The area of the resulting block is minimized or given a fixed die
- Other consideration: timing, routability, clock, buffering and interaction with physical synthesis

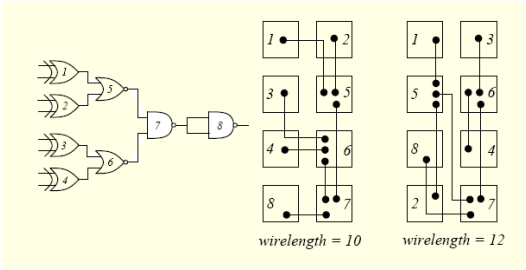
Modern Mixed-mode Placement



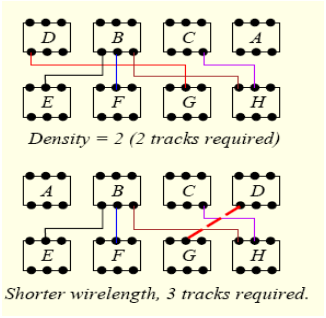
- **Many macros**
- **data paths + dust logic**
- **I/O constraint (area I/O or wirebond)**

(source: IBM)

Different Wire Length



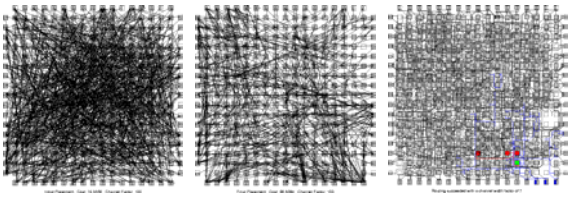
Different Routability/Chip Area



Placement can Make a Difference

- MCNC Benchmark circuit e64 (contains 230 4-LUT). Placed to a FPGA.

Random Initial Placement	Final Placement	After Detailed Routing



Importance of Placement

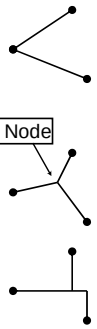
- Fundamental problem for physical design
- Glue of the physical synthesis
- Becomes very active again in recent years:
  - ISPD'05 and 06 Placement Contests
- Reasons:
  - Serious interconnect issues (delay, routability, noise) in deep-submicron design
  - Placement problem becomes significantly larger
  - Cong et al. [ASPDAC-03, ISPD-03, ICCAD-03] point out that existing placers are far from optimal, not scalable, and not stable

Interconnect Topology Optimization

- Problem: given a source and a set of sinks, build the best interconnect topology to minimize different design objectives:
  - Wire length: traditional (lower capacitance load, and overall congestion)
  - Performance: DSM
  - New interest: speed and other non-traditional routing architecture
- In most cases, topology means tree
  - Because tree is the most compact structure to connect everything without redundancy
  - Delay analysis is easy (cf. mesh)

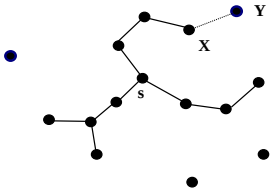
Terminology

- For multi-terminal net, we can easily construct a tree (spanning tree) to connect the terminals together.
- However, the wire length may be unnecessarily large.
- Better use Steiner Tree:
  - A tree connecting all terminals as well as other added nodes (Steiner nodes).
- Rectilinear Steiner Tree:
  - Steiner tree such that edges can only run horizontally and vertically.
  - Manhattan planes



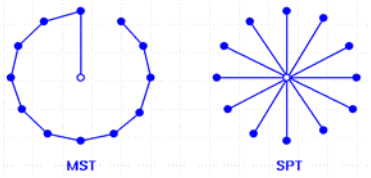
Prim's Algorithm for Minimum Spanning Tree

- Grow a connected subtree from the source, one node at a time.
- At each step, choose the closest un-connected node and add it to the subtree.

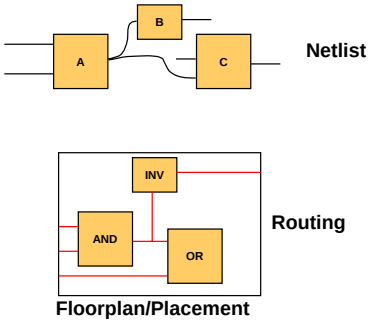


Interconnect Topology Optimization Under Linear Delay Model

- Conventional Routing Algorithms Are Not Good Enough
  - *Minimum spanning tree* may have very long source-sink path.
  - *Shortest path tree* may have very large routing cost.
- Want to minimize path lengths and routing cost at the same time.



Routing in design flow

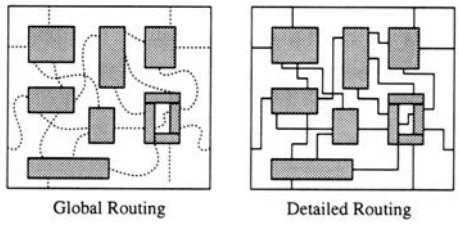


Routing Problem is Very Hard

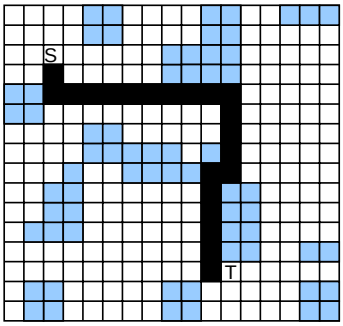
- Minimum Steiner Tree Problem:
  - Given a net, find the Steiner tree with the minimum length.
  - This problem is NP-Complete!
- May need to route tens of thousands of nets simultaneously without overlapping.
- Obstacles may exist in the routing region.

General Routing Paradigm

Two phases:



Maze Routing



An Illustration

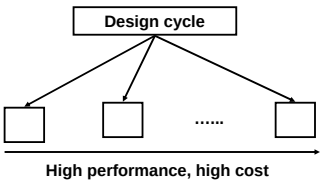
S 0	1	2	3
1	2	3	
	3	4	5
5	4	5	T 6

Extraction and Timing Analysis

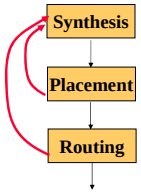
- After global routing and detailed routing, information of the nets can be extracted and delays can be analyzed.
- If some nets fail to meet their timing budget, detailed routing and/or global routing needs to be repeated.

Complexities of Physical Design

- **More than 100 million transistors**
- **Performance driven designs**
- **Power-constrained designs**
- **Time-to-Market**



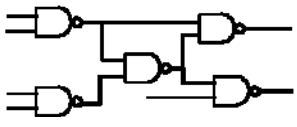
Grand Challenge: Design Closure



BIG problem: non-converging or too many iterations

Transistor/Gate Sizing

- Given: Logic network with or without cell library
  Find: Optimal size for each transistor/gate to minimize delay, or area or power under delay constraint

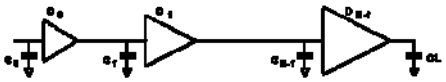


- Transistor sizing versus gate sizing (=> device sizing)

Device Sizing

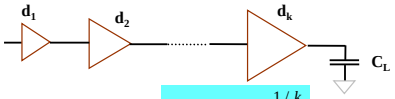
- Device sizing is one of the key techniques for circuit optimization
  - For standard cell type of designs, gate sizing
    - For example of inverters, INV-A, INV-B, INV-C, …, INV-N, …, each having a different driving capabilities.
  - For microprocessor or custom designs, more fine-grained control, transistor sizing
    - For each transistor
- Main techniques
  - Analytical formula: e.g., driver sizing [Lin-Linholm, JSSC'75]
  - Greedy algorithm: [Cong et al, 1996], [Chen et al, ICCAD 1998]
  - Mathematical programming

Driver Sizing



- Given:
  - A chain of cascaded drivers driving a load
  - Ignore the interconnect between drivers (i.e. assume driver and load CL is closer enough)
- Obtain:
  - Optimize the driver sizes to minimize delay, or minimize total area while meeting target delay
- [Lin-Linholm, JSSC'75]: a classic result without wiring consideration
- Delay and area/power tradeoff

An Early Work on Driver Sizing
[Lin-Linholm, JSSC'75]

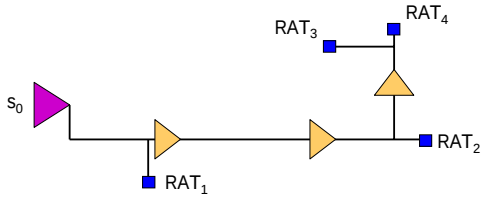


- Constant stage ratio, $\frac{d_{i+1}}{d_i} = \left(\frac{C_L}{C_g}\right)^{1/k}$
- if the number of drivers is not fixed, $\frac{d_{i+1}}{d_i} = e$
- Interconnect is modeled as a lumped capacitor

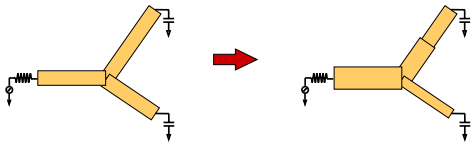
Buffer Insertion

Find: Buffer locations and a routing tree such that slack at the source is minimized

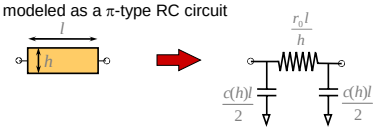
$$q(s_0) = \min_{1 \le i \le 4}\{RAT(s_i) - delay(s_0, s_i)\}$$



Wire Sizing Problem



- We can modeled as a π-type RC circuit



- Both resistance and capacitance depend on wire width.
- Hence delay can be optimized by changing wire width.