CS612
Algorithms for Electronic Design Automation

Course Overview

Mustafa Ozdal

What is EDA?

- Stands for *Electronic Design Automation*
  - a.k.a VLSI CAD
- Software tools to support engineers in the creation of new IC designs.
- EDA tools significantly reduce the cost and time-to-market of new projects.
  - A CPU can easily contain > 1B transistors in a single chip
  - Manual design is prohibitive

What is EDA?

- Solves a wide-range of problems

  high-level system design to fabrication (and everything in between)

- Strong software skills required

- This course will cover the physical design problems
  - Abstract and algorithmic problems
  - No EE knowledge needed

Mustafa Ozdal
Computer Engineering Department, Bilkent University

Why Study EDA Algorithms?

- You may consider a career in the EDA field
  - EDA companies: Synopsys, Cadence, Mentor Graphics, …
  - Design companies: Intel, IBM, Apple, AMD, Nvidia, TI, Qualcomm, ARM, TSMC, …
- You may be working in a related field
  - e.g. Computer architecture: What is the hardware cost, energy cost, etc. of a new feature?
  - e.g. Scientific computing: Common algorithms used in both domains (e.g. graph partitioning, clustering, etc.)
- You may want to improve your algorithmic skills
  - We will study algorithms for abstract problems that also occur in other domains. e.g. routing, rectangle packing, partitioning, etc.

Course Overview

- Schedule:

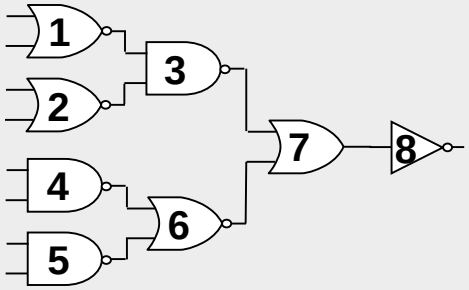
Lecture:	Tue. 13:40-14:30	EA502
Lecture:	Thu. 15:40-17:30	EA502
Spare Hour:	Tue. 14:40:15:30	

- Course project: More in-depth study of a topic
  - Literature survey + implementation + experiments
  - Presentation (survey + plans)
  - Final report (implementation + experiments + conclusions)

Sneak Preview of the Problems

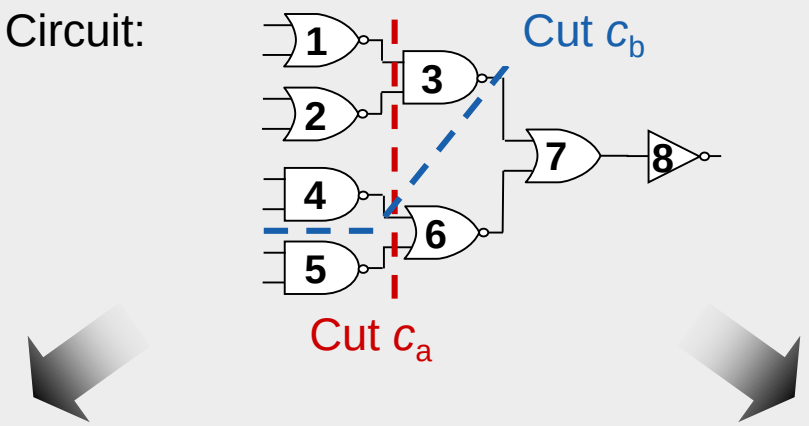
Partitioning Problem

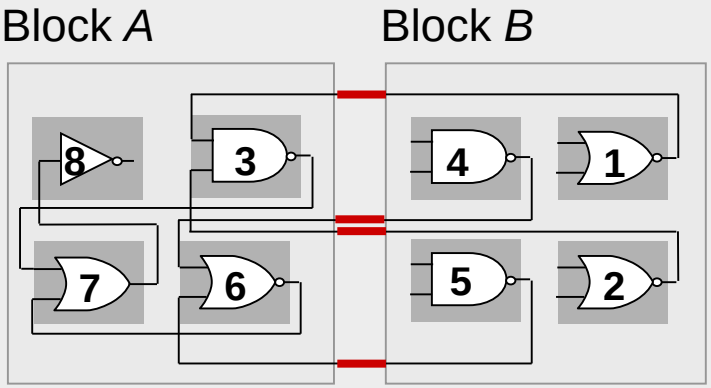
Circuit:



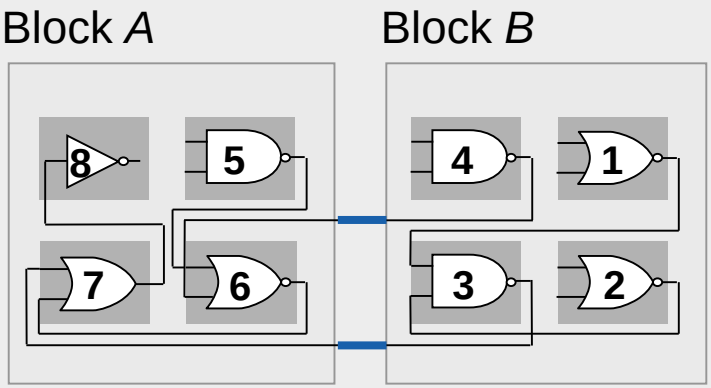
Partition the netlist into 2 equal parts (i.e. each part must have 4 gates) such that the # of edges between two partitions is minimized.

Partitioning Problem





Cut c_a: four external connections



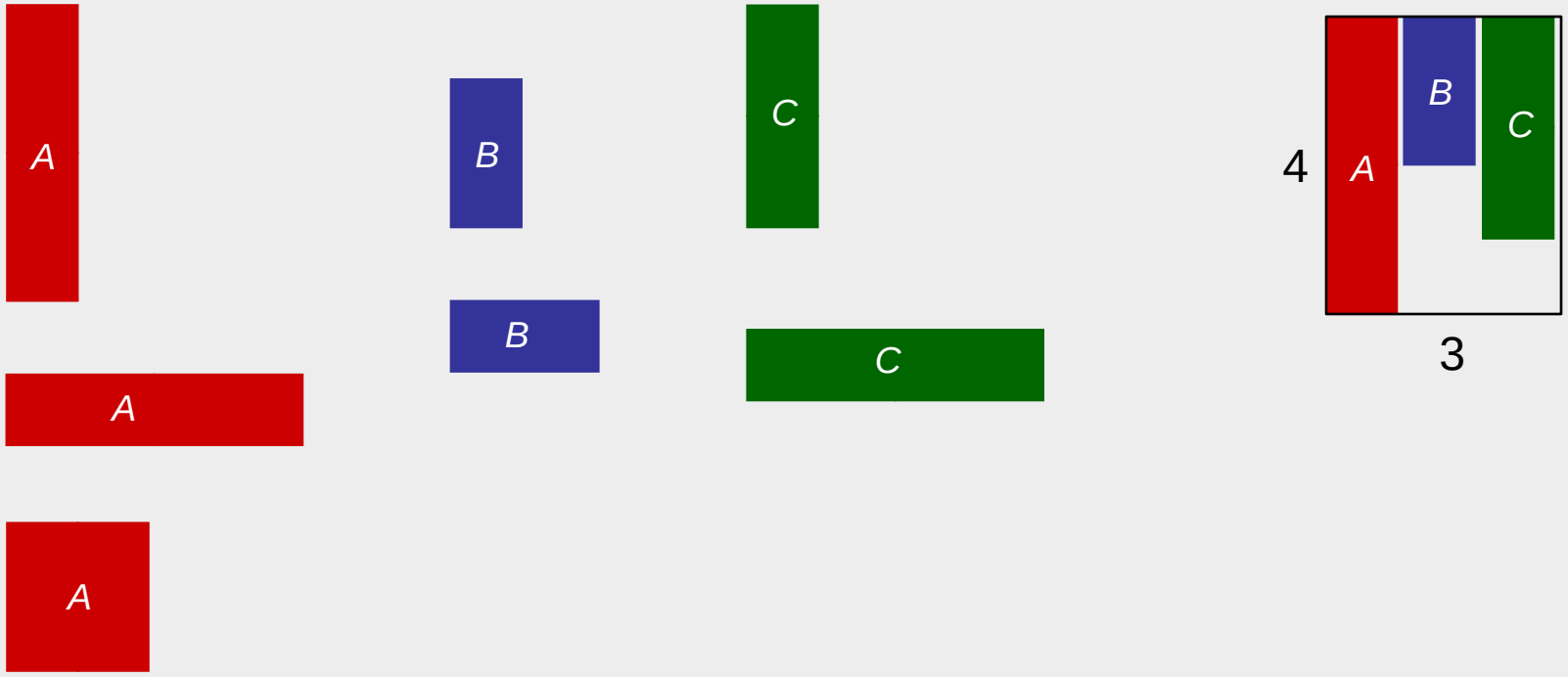
Cut c_b: two external connections

How to do this for > 1M gates?

Floorplanning Problem

Example
Given: Three blocks with the following potential widths and heights
Block *A*: $w = 1, h = 4$ or $w = 4, h = 1$ or $w = 2, h = 2$
Block *B*: $w = 1, h = 2$ or $w = 2, h = 1$
Block *C*: $w = 1, h = 3$ or $w = 3, h = 1$

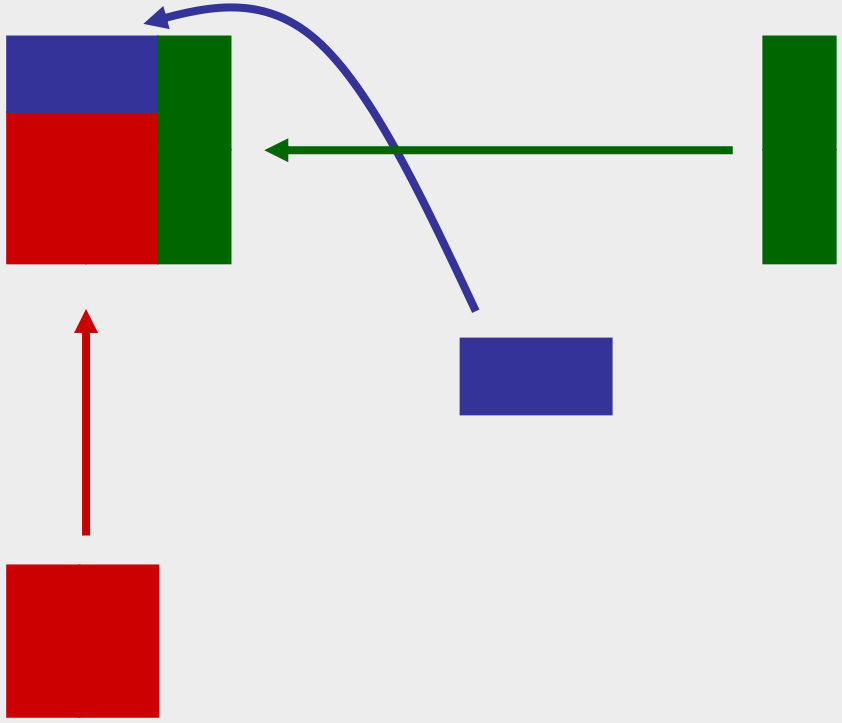
Task: Floorplan with minimum total area enclosed



Floorplanning Problem

Example
Given: Three blocks with the following potential widths and heights
Block *A*: $w = 1, h = 4$ or $w = 4, h = 1$ or $w = 2, h = 2$
Block *B*: $w = 1, h = 2$ or $w = 2, h = 1$
Block *C*: $w = 1, h = 3$ or $w = 3, h = 1$

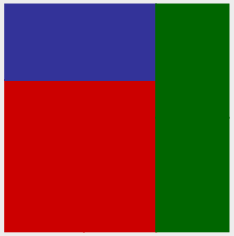
Task: Floorplan with minimum total area enclosed



Floorplanning Problem

Example
Given: Three blocks with the following potential widths and heights
Block *A*: $w = 1, h = 4$ or $w = 4, h = 1$ or $w = 2, h = 2$
Block *B*: $w = 1, h = 2$ or $w = 2, h = 1$
Block *C*: $w = 1, h = 3$ or $w = 3, h = 1$

Task: Floorplan with minimum total area enclosed

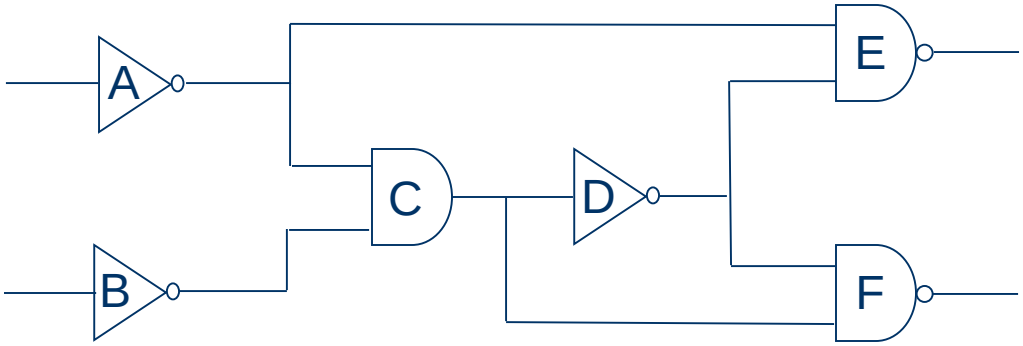


Solution:
Aspect ratios
Block *A* with $w = 2, h = 2$; Block *B* with $w = 2, h = 1$; Block *C* with $w = 1, h = 3$

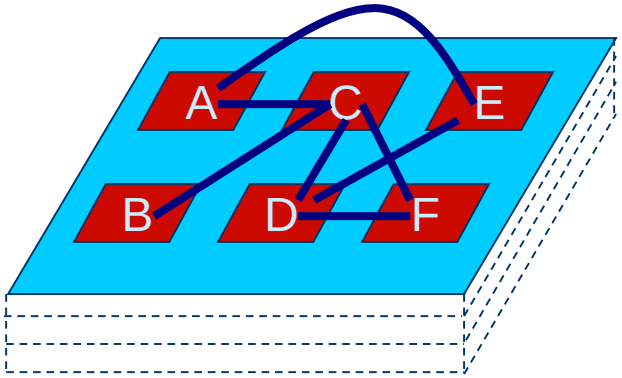
This floorplan has a global bounding box with minimum possible area (9 square units).

How to do this for 1000s of blocks?

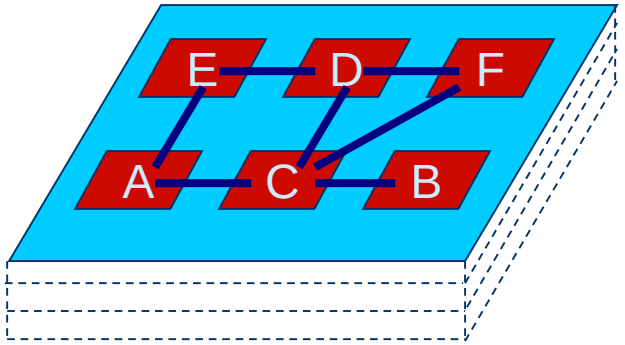
Placement



Which is better?



More wirelength
Harder routing



Less wirelength
Easier routing

Placement as an Optimization Problem

- Place all cells in the netlist such that:
  - Minimize chip area
  - Minimize wire length
  - Make routing easy
  - Satisfy timing constraints
    - Keep cells on critical paths closer
  - Satisfy various other design constraints
- A typical design can have > 1M cells
- NP-complete problem

Routing Problem

Netlist:

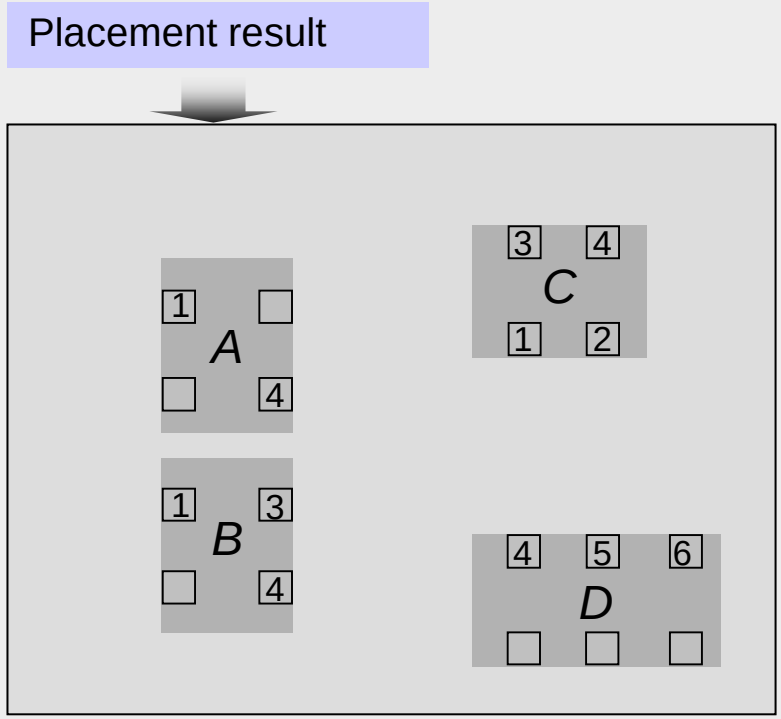
$N_1 = \{C_4, D_6, B_3\}$

$N_2 = \{D_4, B_4, C_1, A_4\}$

$N_3 = \{C_2, D_5\}$

$N_4 = \{B_1, A_1, C_3\}$

Technology Information (Design Rules)



Routing Problem

Netlist:

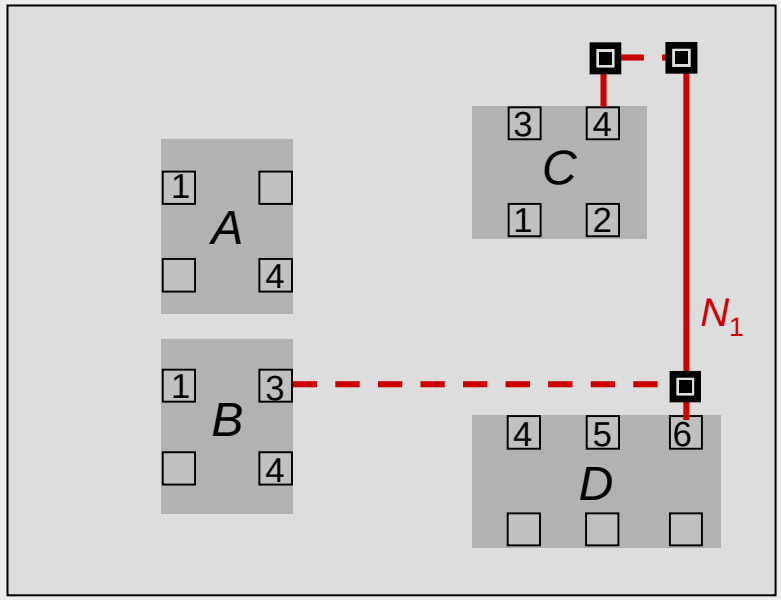
$N_1 = \{C_4, D_6, B_3\}$

$N_2 = \{D_4, B_4, C_1, A_4\}$

$N_3 = \{C_2, D_5\}$

$N_4 = \{B_1, A_1, C_3\}$

Technology Information (Design Rules)



Routing Problem

Netlist:

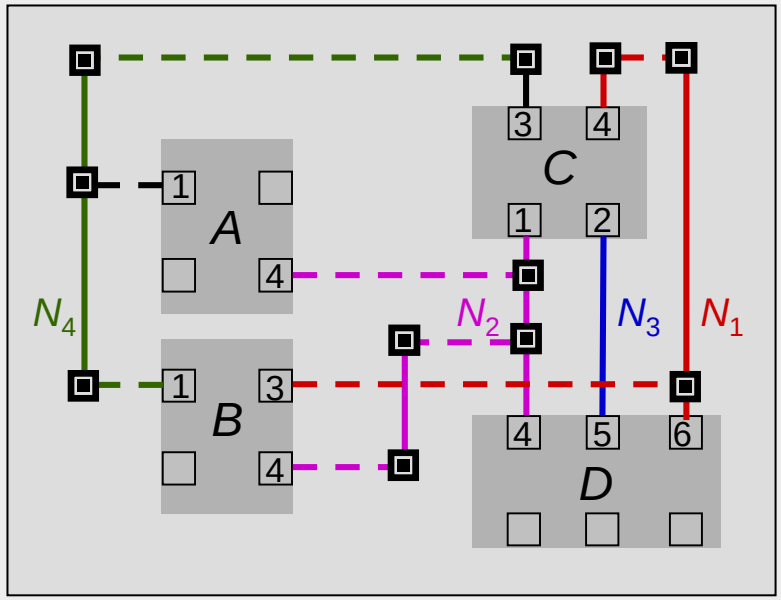
$N_1 = \{C_4, D_6, B_3\}$

$N_2 = \{D_4, B_4, C_1, A_4\}$

$N_3 = \{C_2, D_5\}$

$N_4 = \{B_1, A_1, C_3\}$

Technology Information
(Design Rules)



How to do this for > 1M nets?