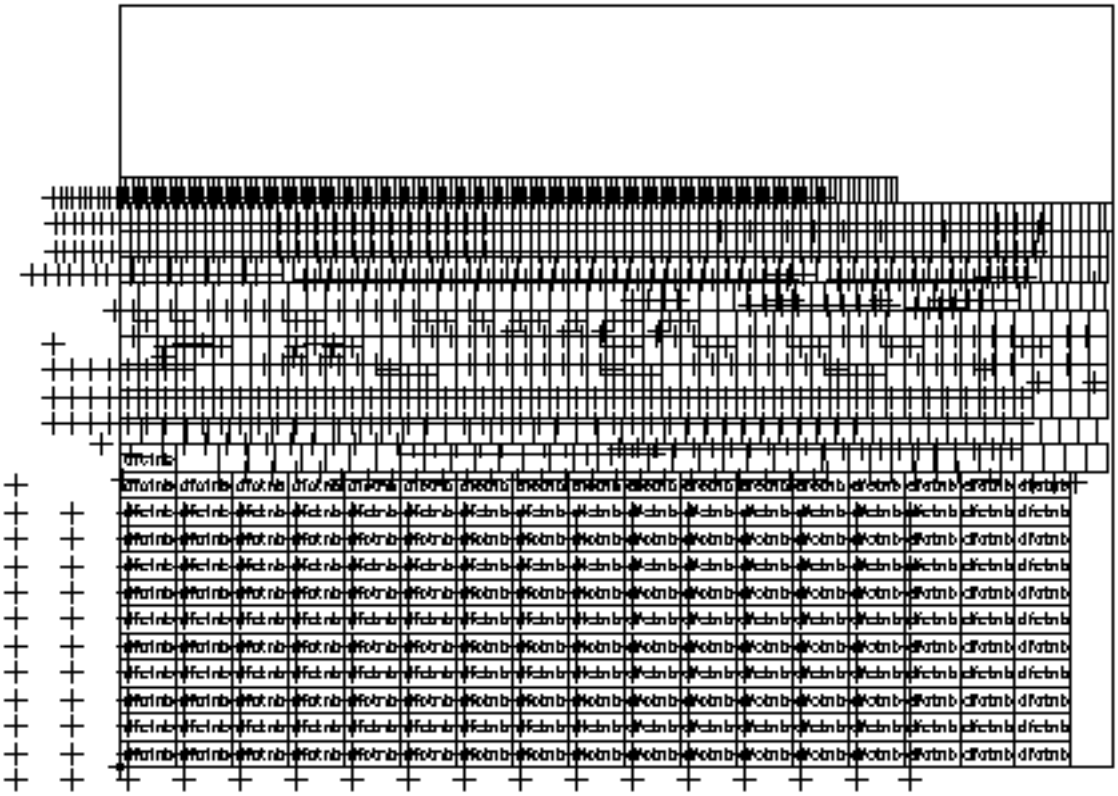
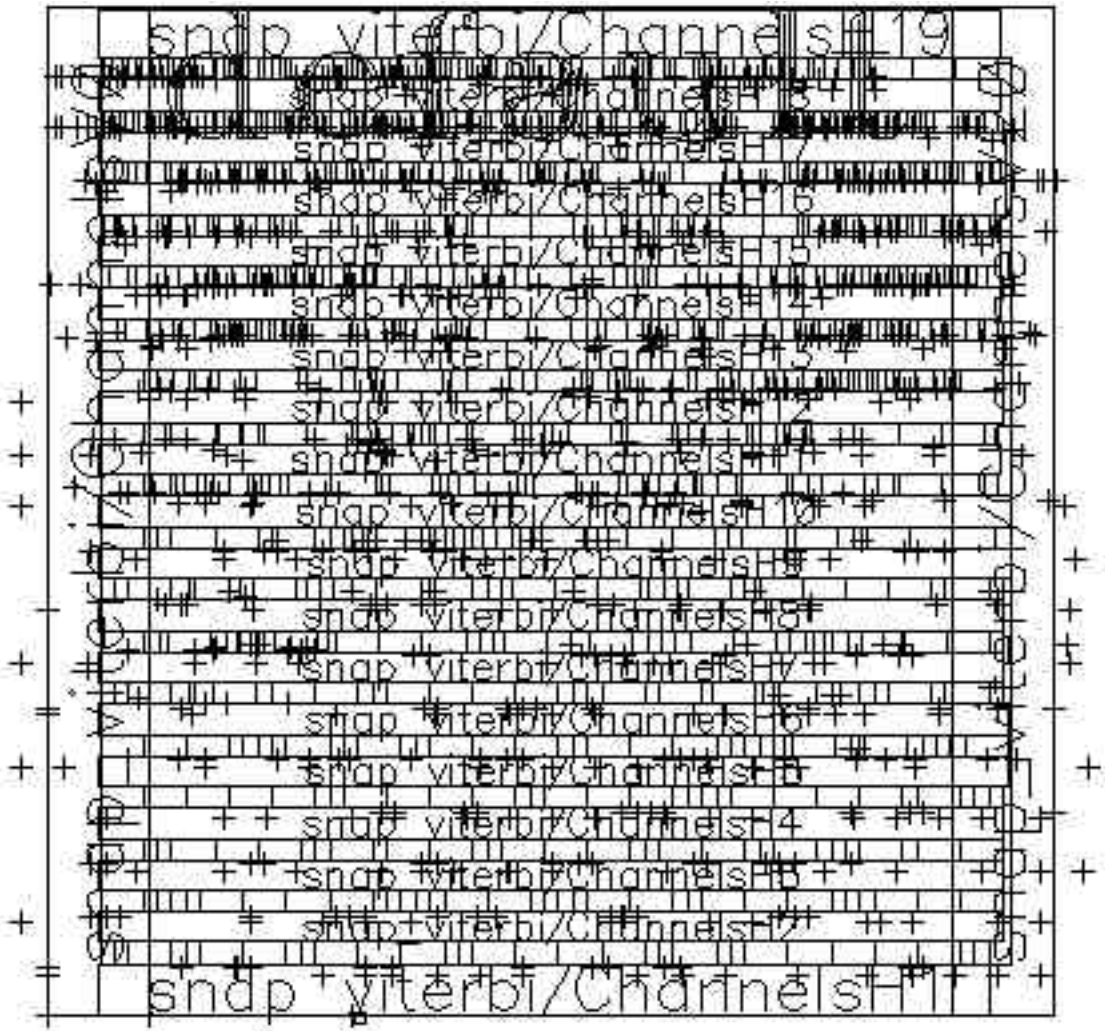
FLOORPLANNING AND PLACEMENT

16

Key terms and concepts: The input to floorplanning is the output of system partitioning and design entry—a netlist. The output of the placement step is a set of directions for the routing tools.



The starting point for floorplanning and placement for the Viterbi decoder (standard cells).



The Viterbi decoder after floorplanning and placement.

16.1 Floorplanning

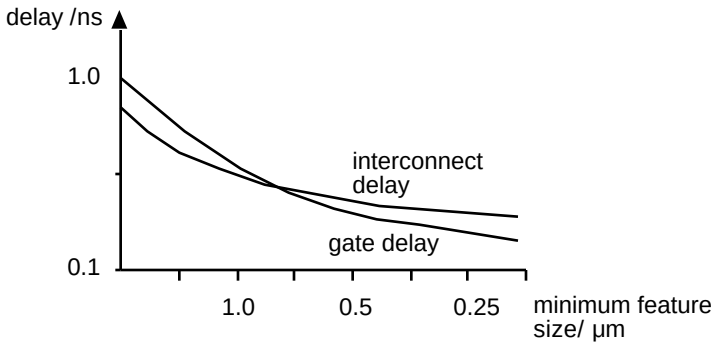
Key terms and concepts: Interconnect and gate delay both decrease with feature size—but at different rates • Interconnect capacitance bottoms out at 2pFcm^{-1} for a minimum-width wire, but gate delay continues to decrease • Floorplanning predicts interconnect delay by estimating interconnect length

Interconnect and gate delays.

As feature sizes decrease, both average interconnect delay and average gate delay decrease—but at different rates.

This is because interconnect capacitance tends to a limit that is independent of scaling.

Interconnect delay now dominates gate delay.



16.1.1 Floorplanning Goals and Objectives

Key terms and concepts: Floorplanning is a mapping between the **logical description** (the **netlist**) and the **physical description** (the **floorplan**).

Goals of floorplanning:

- arrange the blocks on a chip,
- decide the location of the I/O pads,
- decide the location and number of the power pads,
- decide the type of power distribution, and
- decide the location and type of clock distribution.

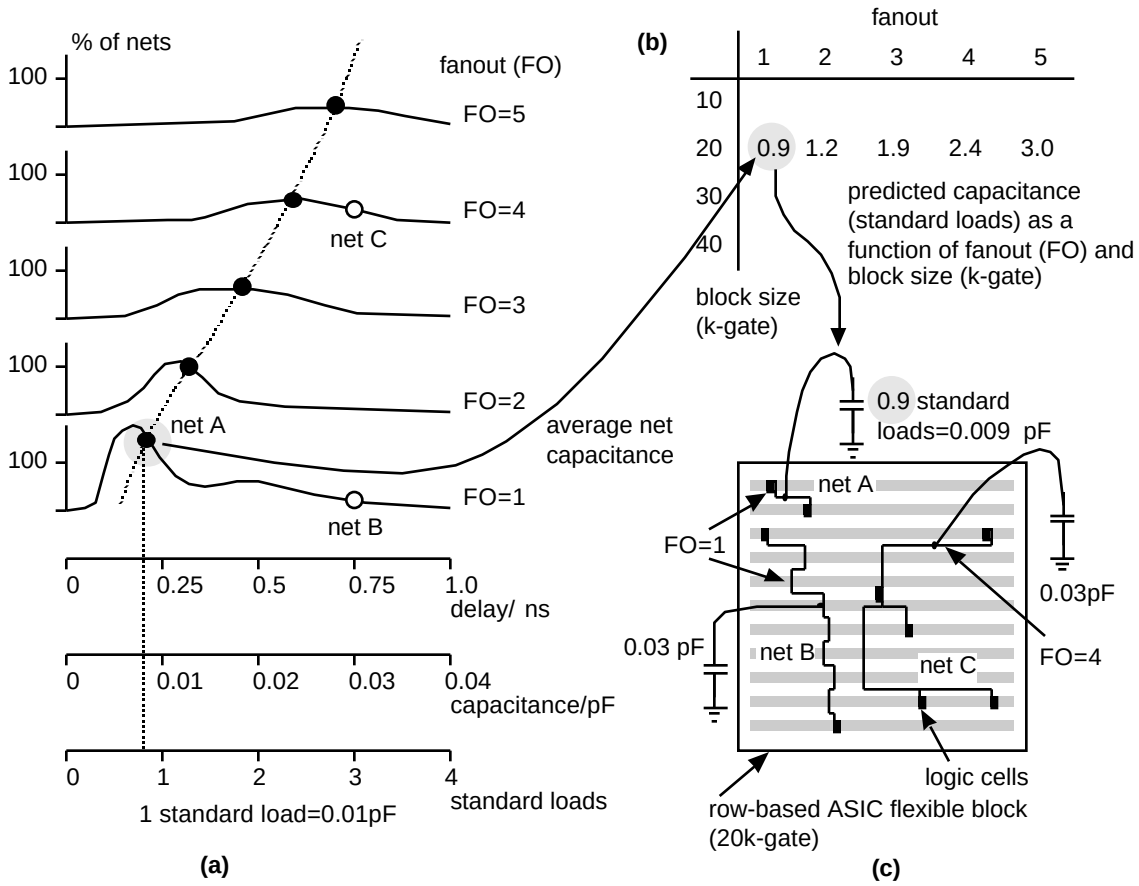
Objectives of floorplanning are:

- to minimize the chip area, and
- minimize delay.

16.1.2 Measurement of Delay in Floorplanning

Key terms and concepts: To predict performance before we complete routing we need to answer "How long does it takes to get from Russia to China?" • In floorplanning we may even move Russia and China • We don't yet know the **parasitics** of the **interconnect capacitance** • We

know only the **fanout** (**FO**) of a net and the size of the block • We estimate interconnect length from **predicted-capacitance tables** (**wire-load tables**)



Predicted capacitance.

(a) Interconnect lengths as a function of fanout (FO) and circuit-block size.

(b) Wire-load table.

There is only one capacitance value for each fanout (typically the average value).

(c) The wire-load table predicts the capacitance and delay of a net (with a considerable error).

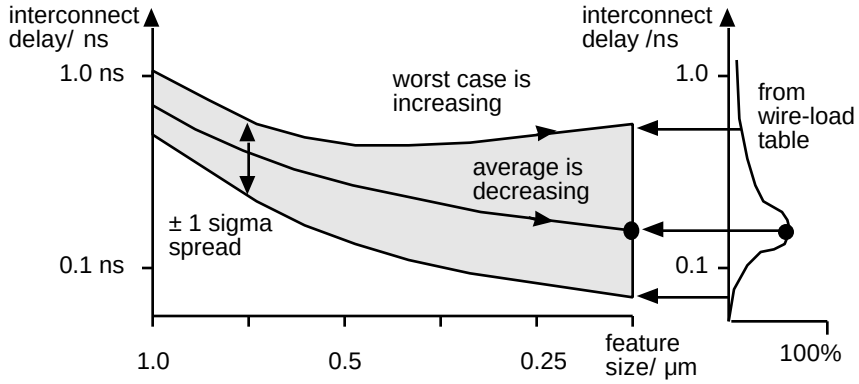
Net A and net B both have a fanout of 1, both have the same predicted net delay, but net B in fact has a much greater delay than net A in the actual layout (of course we shall not know what the actual layout is until much later in the design process).

A wire-load table showing average interconnect lengths (mm).

Array (available gates)	Chip size (mm)	Fanout 1	Fanout 2	Fanout 4
3k	3.45	0.56	0.85	1.46
11k	5.11	0.84	1.34	2.25
105k	12.50	1.75	2.70	4.92

Worst-case interconnect delay.

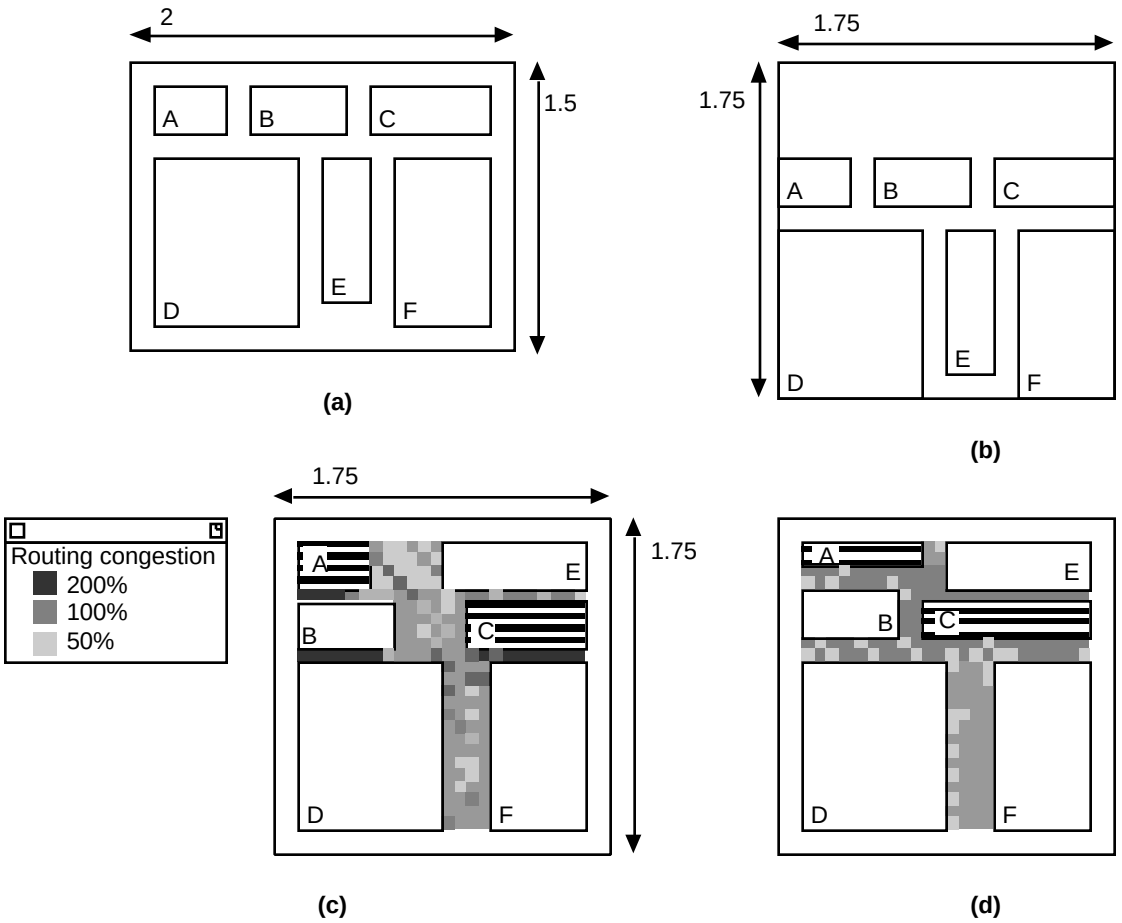
As we scale circuits, but avoid scaling the chip size, the worst-case interconnect delay increases.



16.1.3 Floorplanning Tools

Key terms and concepts: we start with a **random floorplan** generated by a floorplanning tool • **flexible blocks** and **fixed blocks** • **seeding** • **seed cells** • **wildcard symbol** • **hard seed** • **soft seed** • **seed connectors** • **rat's nest** • **bundles** • **flight lines** • **congestion** • **aspect ratio** • **die**

cavity • congestion map • routability • interconnect channels • channel capacity • channel density



Congestion analysis.

(a) The initial floorplan with a 2:1.5 die aspect ratio.

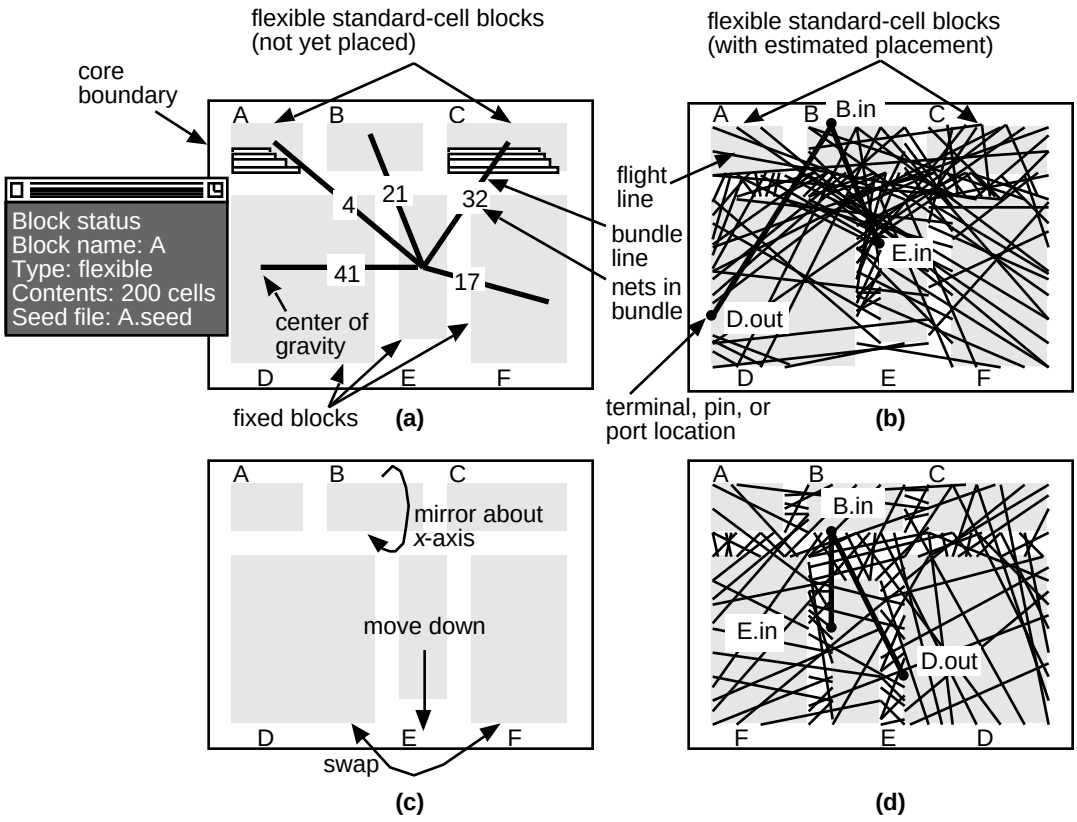
(b) Altering the floorplan to give a 1:1 chip aspect ratio.

(c) A trial floorplan with a congestion map.

Blocks A and C have been placed so that we know the terminal positions in the channels. Shading indicates the ratio of channel density to the channel capacity.

Dark areas show regions that cannot be routed because the channel congestion exceeds the estimated capacity.

(d) Resizing flexible blocks A and C alleviates congestion.



Floorplanning a cell-based ASIC.

(a) Initial floorplan generated by the floorplanning tool.

Two of the blocks are flexible (A and C) and contain rows of standard cells (unplaced).

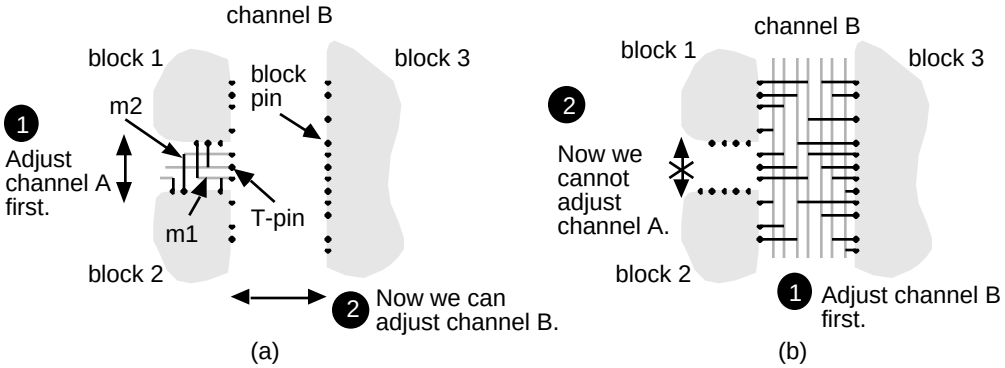
A pop-up window shows the status of block A.

(b) An estimated placement for flexible blocks A and C.

The connector positions are known and a rat's nest display shows the heavy congestion below block B.

(c) Moving blocks to improve the floorplan.

(d) The updated display shows the reduced congestion after the changes.



Routing a T-junction between two channels in two-level metal.

The dots represent logic cell pins.

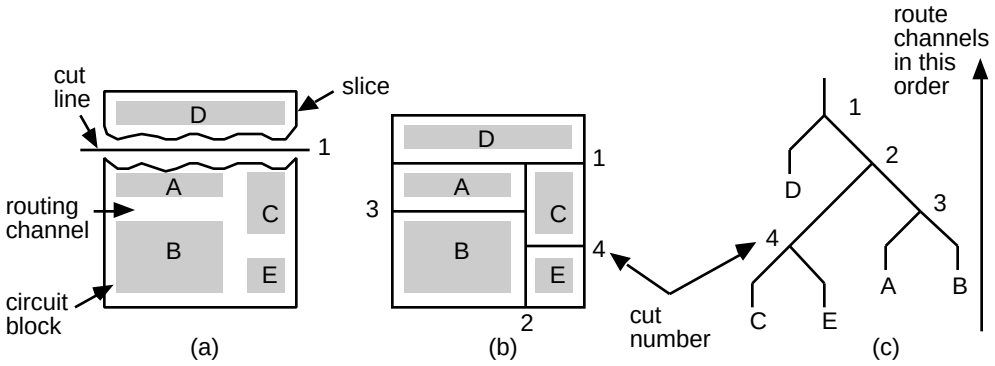
(a) Routing channel A (the stem of the T) first allows us to adjust the width of channel B.

(b) If we route channel B first (the top of the T), this fixes the width of channel A.

We have to route the stem of a T-junction before we route the top.

16.1.4 Channel Definition

Key terms and concepts: **channel definition** or **channel allocation • channel ordering • slicing floorplan • cyclic constraint • switch box • merge • selective flattening • routing order**



Defining the channel routing order for a slicing floorplan using a slicing tree.

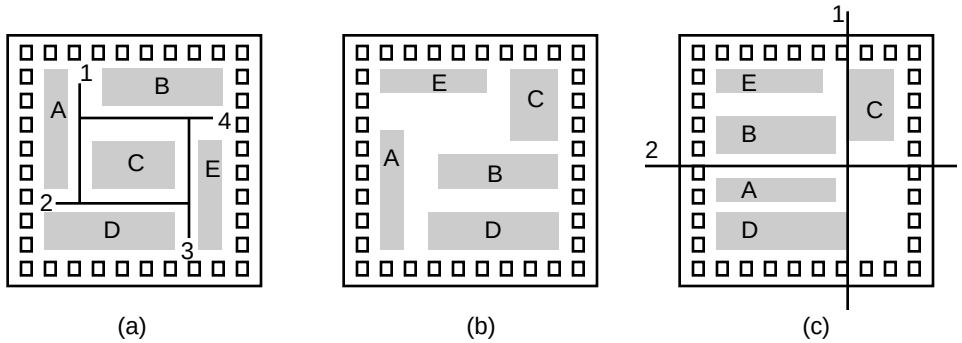
(a) Make a cut all the way across the chip between circuit blocks.

Continue slicing until each piece contains just one circuit block.

Each cut divides a piece into two without cutting through a circuit block.

(b) A sequence of cuts: 1, 2, 3, and 4 that successively slices the chip until only circuit blocks are left.

(c) The slicing tree corresponding to the sequence of cuts gives the order in which to route the channels: 4, 3, 2, and finally 1.

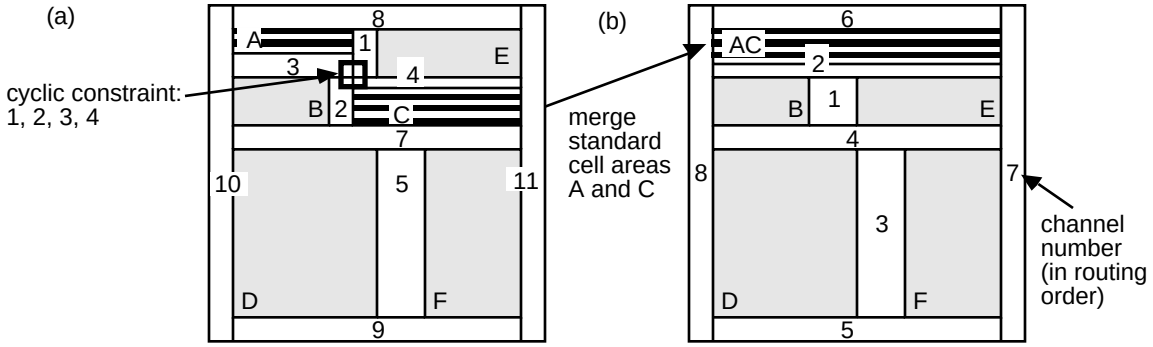


Cyclic constraints.

(a) A nonslicing floorplan with a cyclic constraint that prevents channel routing.

(b) In this case it is difficult to find a slicing floorplan without increasing the chip area.

(c) This floorplan may be sliced (with initial cuts 1 or 2) and has no cyclic constraints, but it is inefficient in area use and will be very difficult to route.



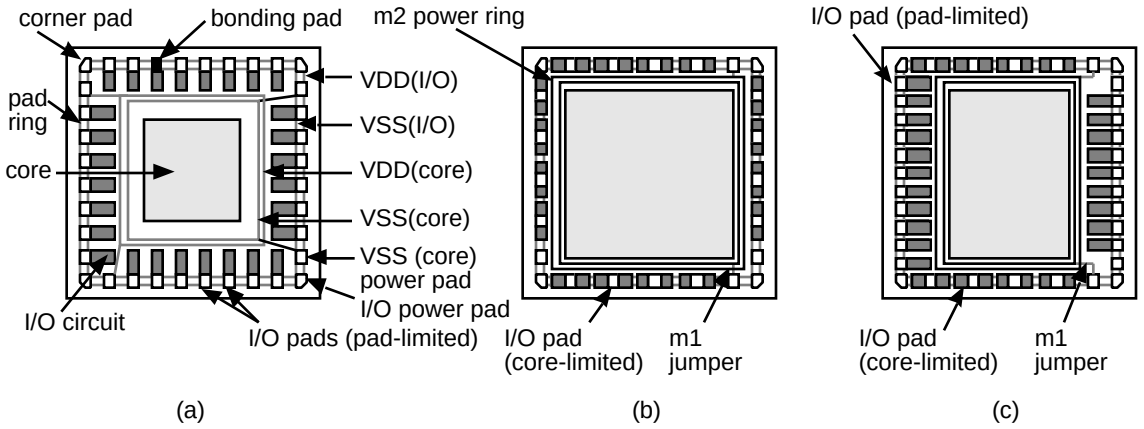
Channel definition and ordering.

(a) We can eliminate the cyclic constraint by merging the blocks A and C.

(b) A slicing structure.

16.1.5 I/O and Power Planning

Key terms and concepts: die • chip carrier • package • bonding • pads • lead frame • package pins • core • pad ring • pad-limited die • core-limited die • pad-limited pads • core-limited pads • power pads • power buses (or power rails) • power ring • dirty power • clean power • electrostatic discharge (ESD) • chip cavity • substrate connection • down bond (or drop bond) • pad seed • double bond • multiple-signal pad • oscillator pad • clock pad. • corner pad • edge pads • two-pad corner cell • bond-wire angle design rules • simultaneously switching outputs (SSOs) • pad mapping • logical pad • physical pad • pad library. • pad-format changer or hybrid corner pad. • global power nets • mixed power supplies • multiple power supplies • stagger-bond • area-bump • ball-grid array (BGA) • pad slot (or pad site) • I/O-cell pitch • pad pitch • channel spine • preferred layer • preferred direction

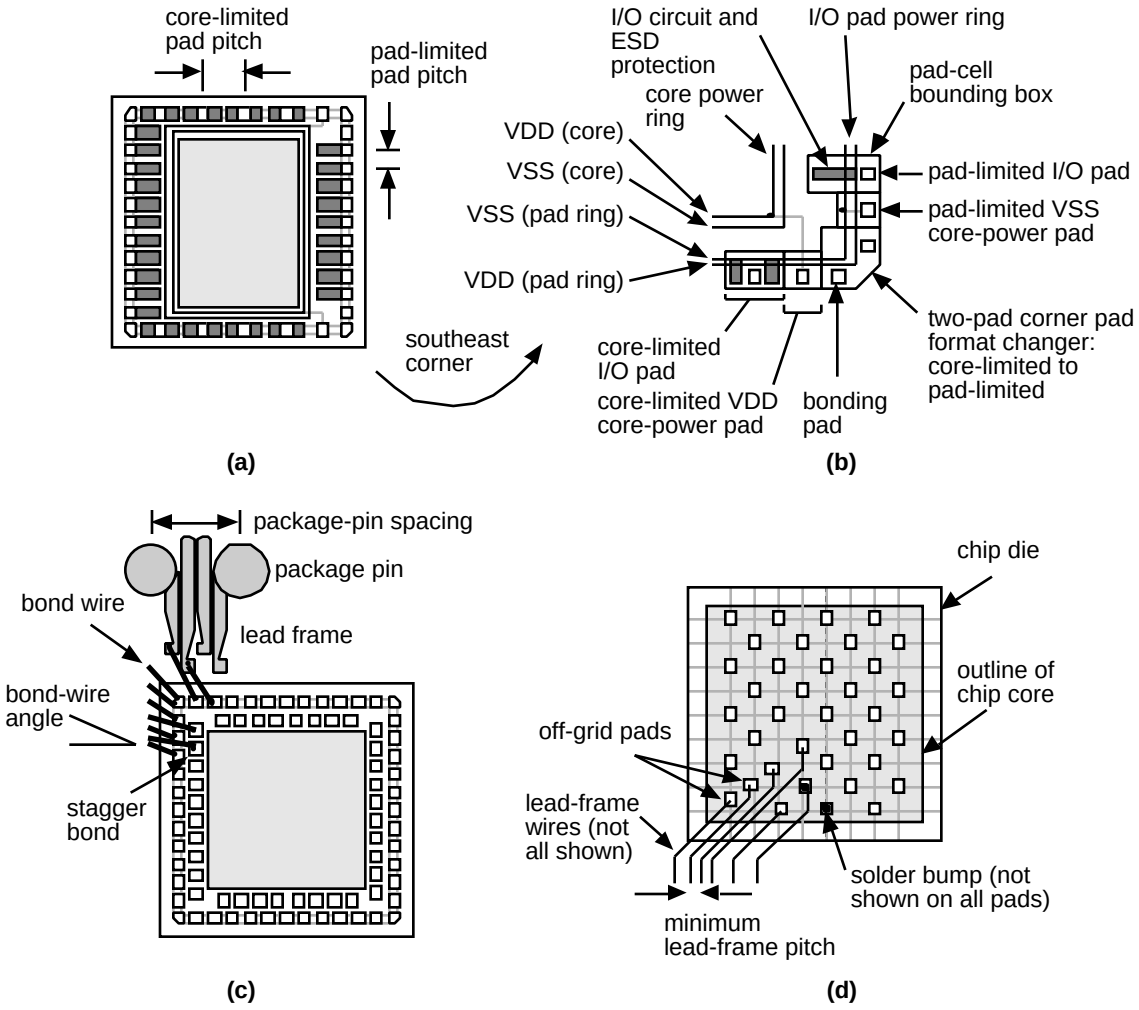


Pad-limited and core-limited die.

(a) A pad-limited die. The number of pads determines the die size.

(b) A core-limited die: The core logic determines the die size.

(c) Using both pad-limited pads and core-limited pads for a square die.



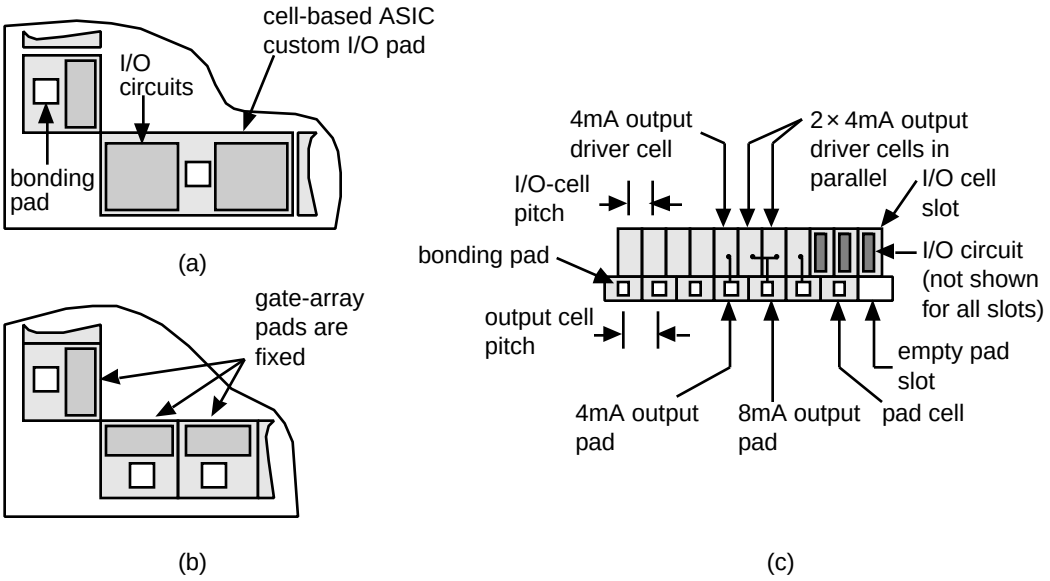
Bonding pads.

(a) This chip uses both pad-limited and core-limited pads.

(b) A hybrid corner pad.

(c) A chip with stagger-bonded pads.

(d) An area-bump bonded chip (or flip-chip). The chip is turned upside down and solder bumps connect the pads to the lead frame.

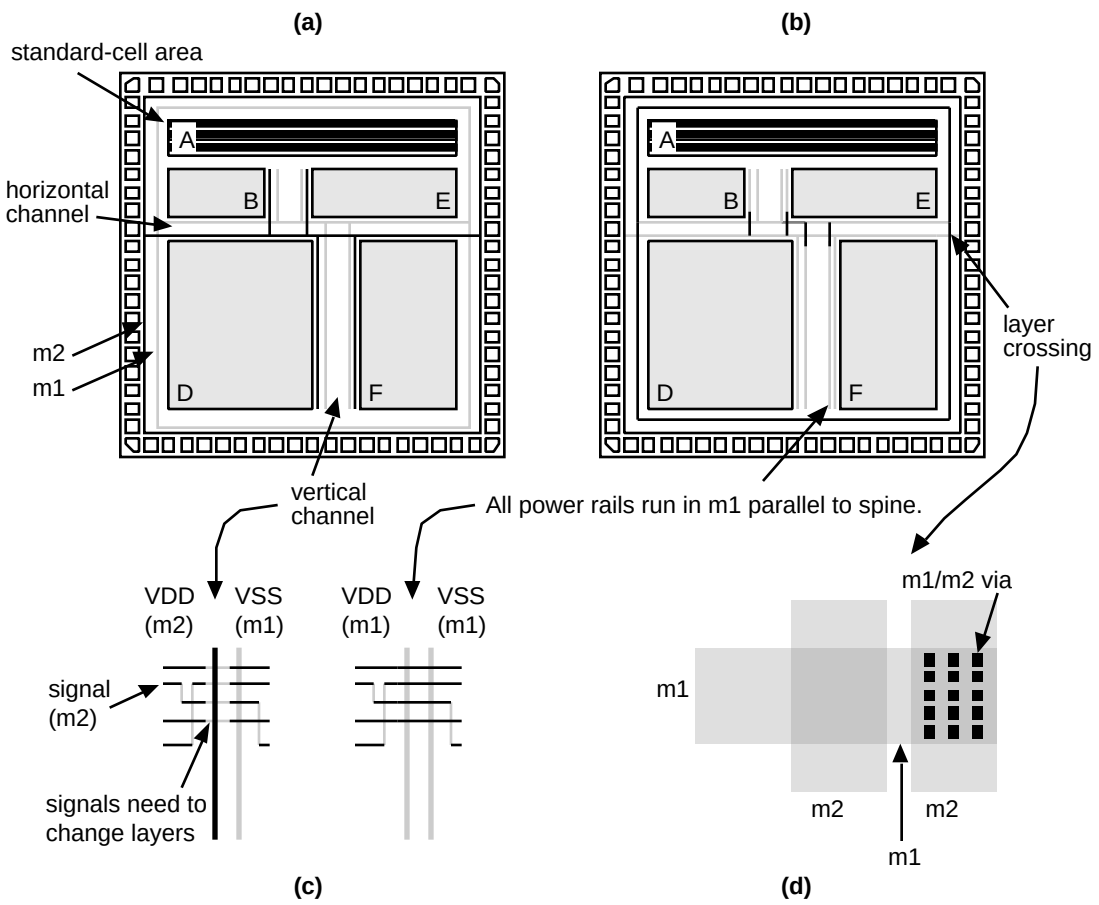


Gate-array I/O pads.

(a) Cell-based ASICs may contain pad cells of different sizes and widths.

(b) A corner of a gate-array base.

(c) A gate-array base with different I/O cell and pad pitches.



Power distribution.

(a) Power distributed using m1 for VSS and m2 for VDD.

This helps minimize the number of vias and layer crossings needed but causes problems in the routing channels.

(b) In this floorplan m1 is run parallel to the longest side of all channels, the channel spine.

This can make automatic routing easier but may increase the number of vias and layer crossings.

(c) An expanded view of part of a channel (interconnect is shown as lines).

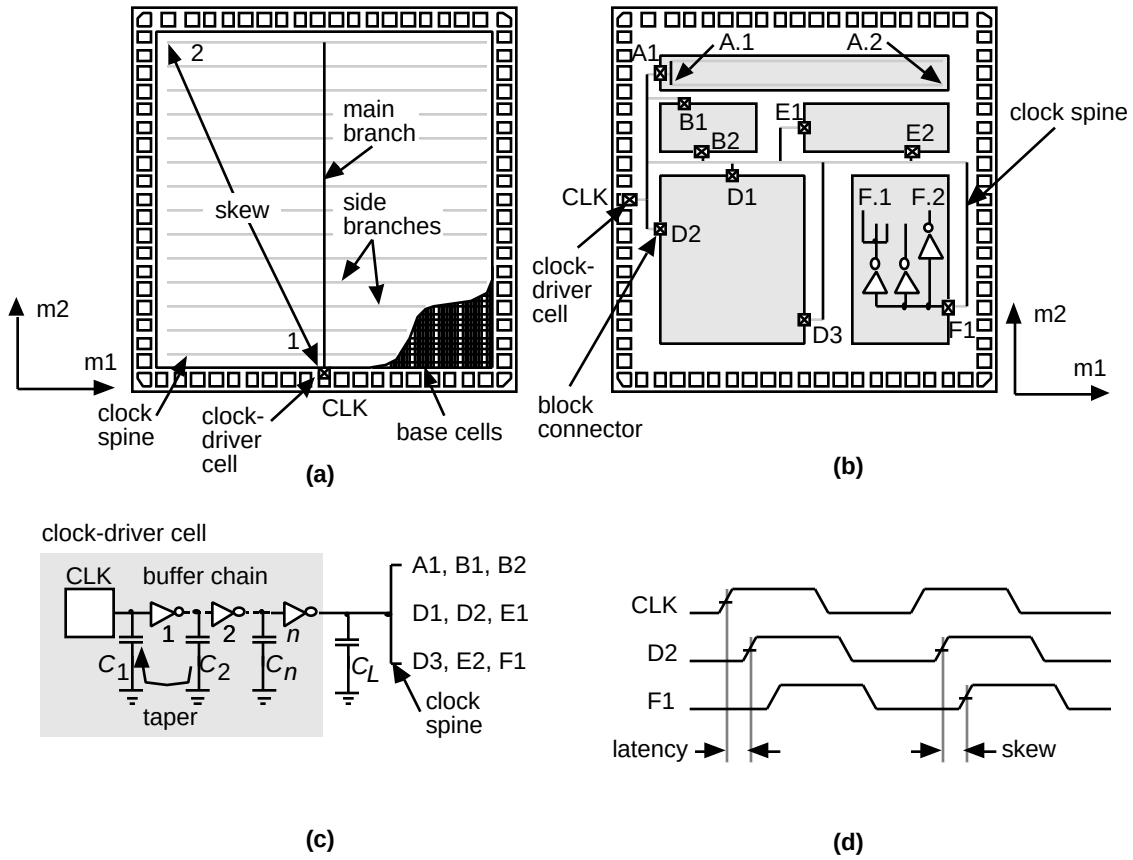
If power runs on different layers along the spine of a channel, this forces signals to change layers.

(d) A closeup of VDD and VSS buses as they cross.

Changing layers requires a large number of via contacts to reduce resistance.

16.1.6 Clock Planning

Key terms and concepts: clock spine • clock skew • clock latency • taper • hot-electron wearout • phase-locked loop (PLL) is an electronic flywheel • jitter



Clock distribution.

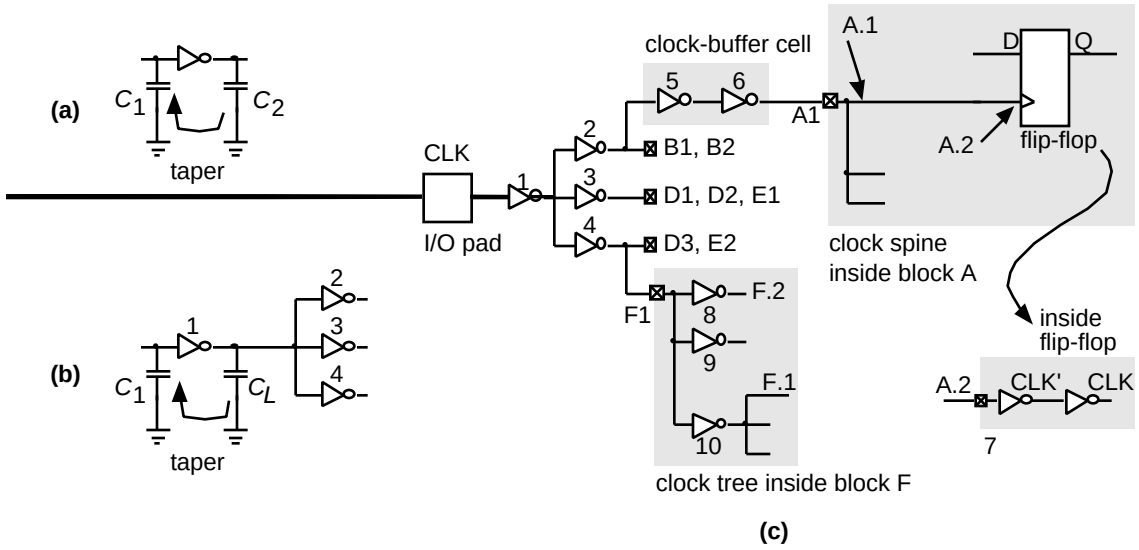
(a) A clock spine for a gate array.

(b) A clock spine for a cell-based ASIC (typical chips have thousands of clock nets).

(c) A clock spine is usually driven from one or more clock-driver cells.

Delay in the driver cell is a function of the number of stages and the ratio of output to input capacitance for each stage (taper).

(d) Clock latency and clock skew. We would like to minimize both latency and skew.



A clock tree.

(a) Minimum delay is achieved when the taper of successive stages is about 3.

(b) Using a fanout of three at successive nodes.

(c) A clock tree for a cell-based ASIC

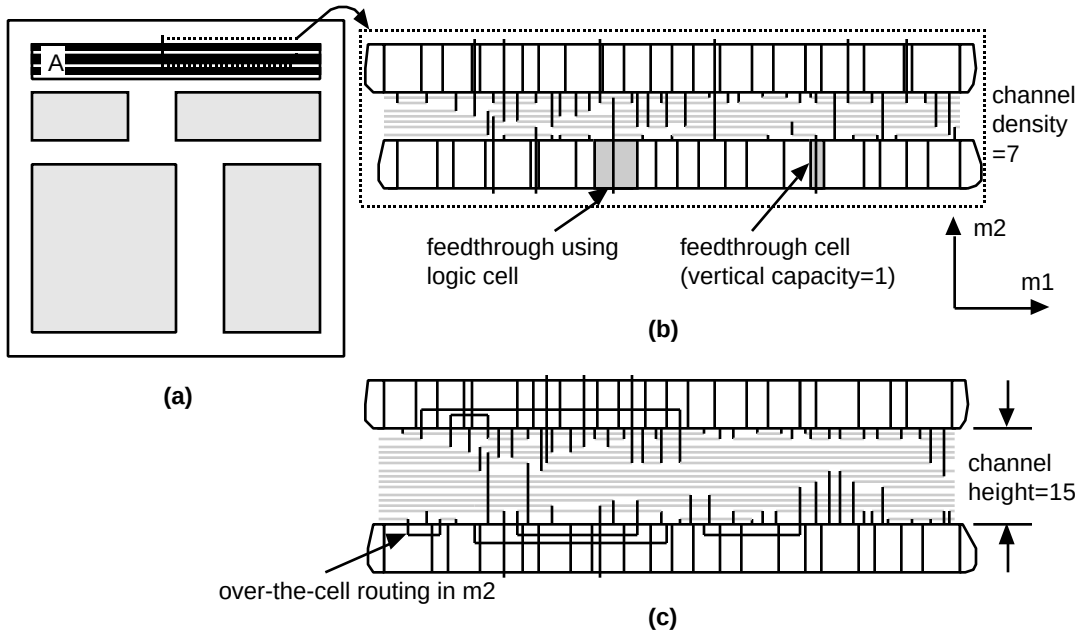
We have to balance the clock arrival times at all of the leaf nodes to minimize clock skew.

16.2 Placement

Key terms and concepts: Placement is more suited to automation than floorplanning. Thus we need measurement techniques and algorithms.

16.2.1 Placement Terms and Definitions

Key terms and concepts: row-based ASICs • over-the-cell routing (OTC routing) • channel capacity • feedthroughs • vertical track (or just track) • uncommitted feedthrough (also built-in feedthrough, implicit feedthrough, or jumper) • double-entry cells • electrically equivalent connectors (or equipotential connectors) • feedthrough cell (or crosser cell) • feedthrough pin or feedthrough terminal • spacer cell • alternative connectors • must-join connectors • logically equivalent connectors • logically equivalent connector groups • fixed-resource ASICs

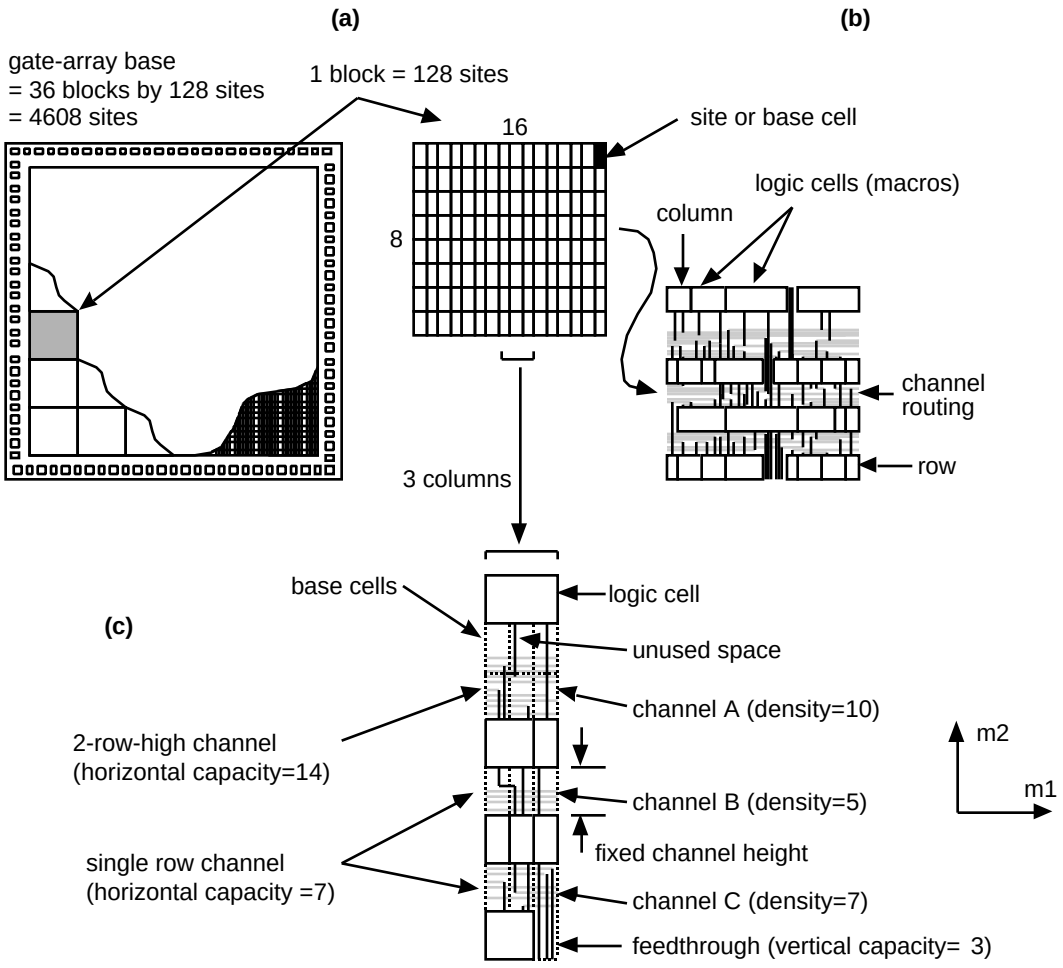


Interconnect structure.

(a) A two-level metal CBIC floorplan.

(b) A channel from the flexible block A. This channel has a channel height equal to the maximum channel density of 7 (there is room for seven interconnects to run horizontally in m1).

(c) A channel that uses OTC (over-the-cell) routing in m2.



Gate-array interconnect.

(a) A small two-level metal gate array (about 4.6k-gate).

(b) Routing in a block.

(c) Channel routing showing channel density and channel capacity.

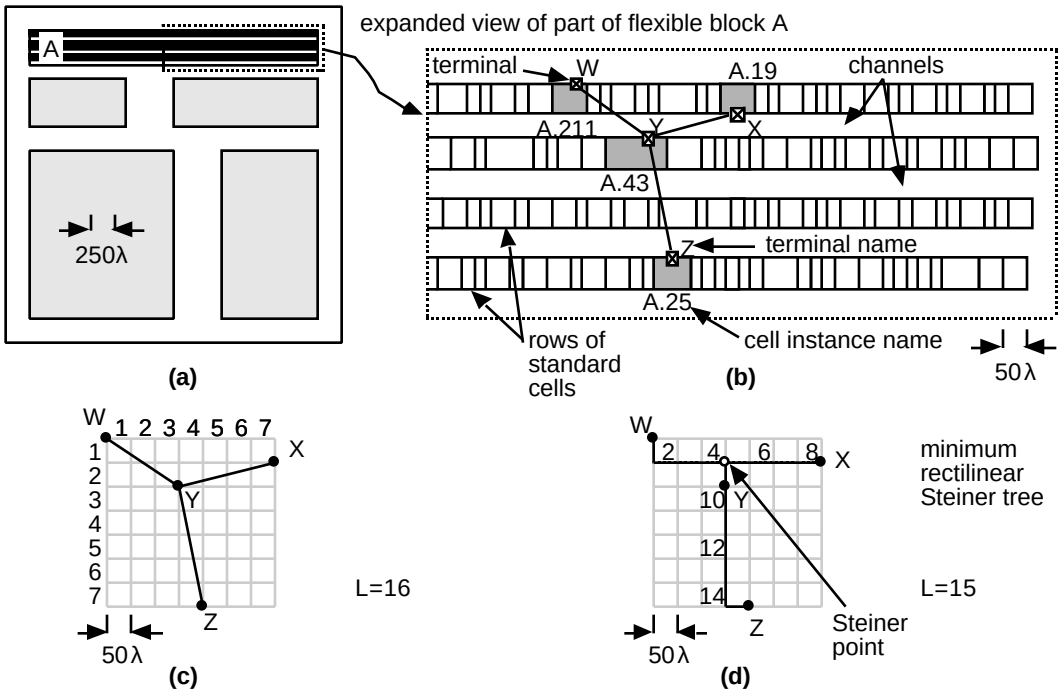
The channel height on a gate array may only be increased in increments of a row. If the interconnect does not use up all of the channel, the rest of the space is wasted. The interconnect in the channel runs in m1 in the horizontal direction with m2 in the vertical direction.

16.2.2 Placement Goals and Objectives

Key terms and concepts: Goals: (1) Guarantee the router can complete the routing step • (2) Minimize all the critical net delays • (3) Make the chip as dense as possible • Objectives: (1) Minimize power dissipation • (2) Minimize crosstalk between signals

16.2.3 Measurement of Placement Goals and Objectives

Key terms and concepts: trees on graphs (or just trees) • Steiner trees • rectilinear routing • Manhattan routing • Euclidean distance • Manhattan distance • minimum rectilinear Steiner tree (MRST) • complete graph • complete-graph measure • bounding box • half-perimeter measure (or bounding-box measure) • meander factor • interconnect congestion • maximum cut line • cut size • timing-driven placement • metal usage



Placement using trees on graphs.

(a) A floorplan.

(b) An expanded view of the flexible block A showing four rows of standard cells for placement (typical blocks may contain thousands or tens of thousands of logic cells).

We want to find the length of the net shown with four terminals, W through Z, given the placement of four logic cells (labeled: A.211, A.19, A.43, A.25).

(c) The problem for net (W, X, Y, Z) drawn as a graph.

The shortest connection is the minimum Steiner tree.

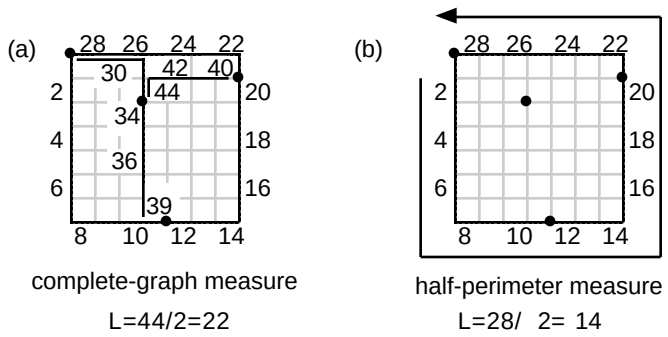
(d) The minimum rectilinear Steiner tree using Manhattan routing.

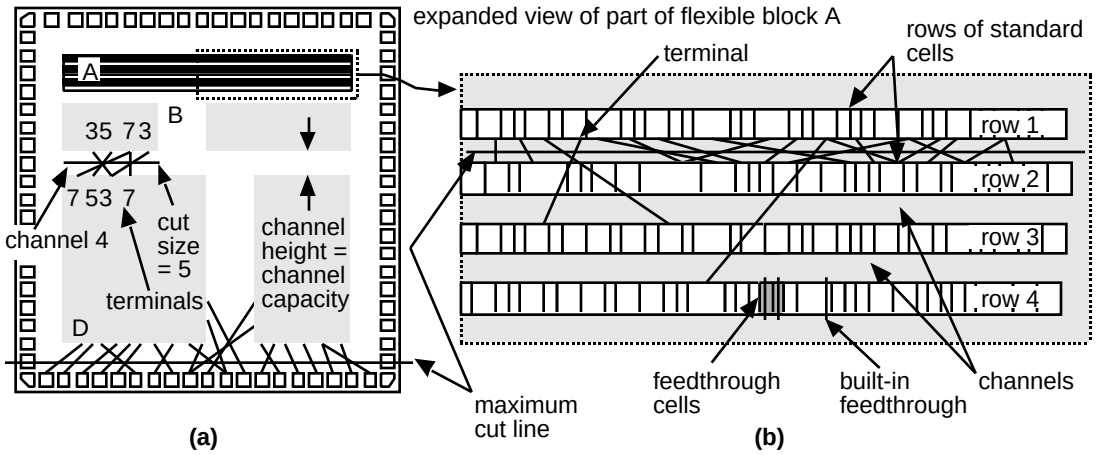
The rectangular (Manhattan) interconnect-length measures are shown for each tree.

Interconnect-length measures.

(a) Complete-graph measure.

(b) Half-perimeter measure.





Interconnect congestion for a cell-based ASIC.

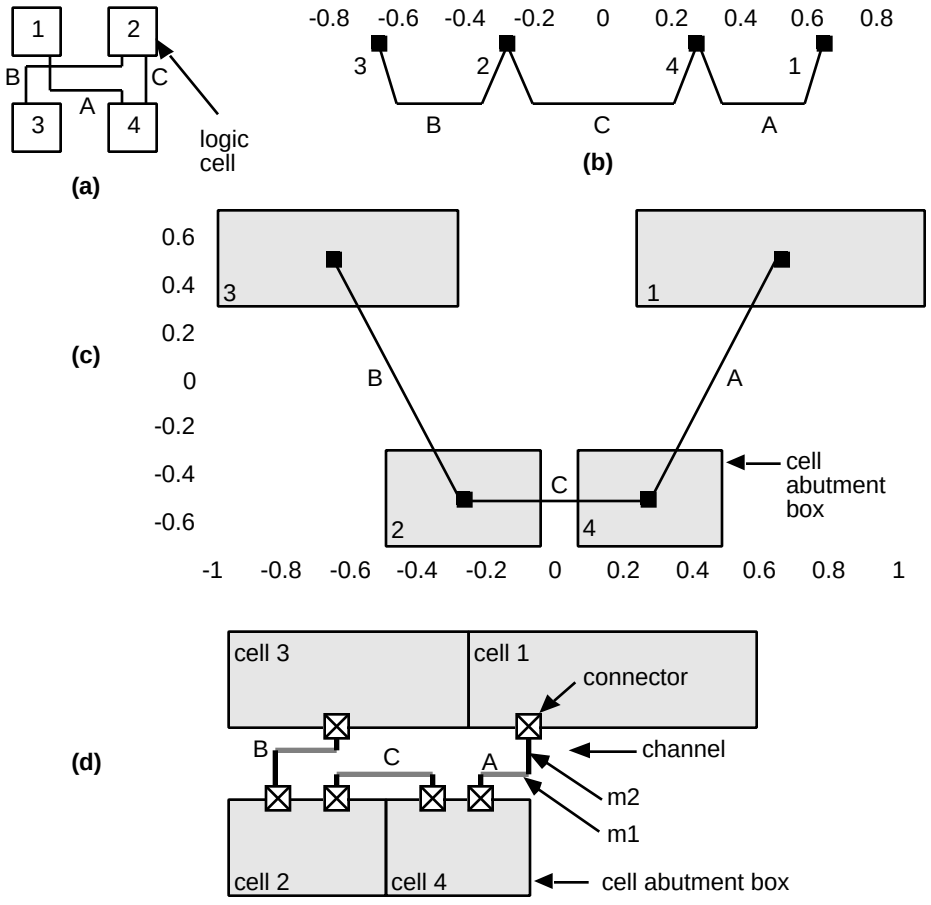
(a) Measurement of congestion.

(b) An expanded view of flexible block A shows a maximum cut line.

16.2.4 Placement Algorithms

Key terms and concepts: constructive placement method • variations on the min-cut algorithm • eigenvalue method • seed placements • min-cut placement • bins • eigenvalue placement algorithm • connectivity matrix (spectral methods) • quadratic placement • disconnection matrix (also called the Laplacian) • characteristic equation • eigenvectors and eigenvalues

16.2.5 Eigenvalue Placement Example



Eigenvalue placement.

(a) An example network.

(b) The one-dimensional placement.

The small black squares represent the centers of the logic cells.

(c) The two-dimensional placement.

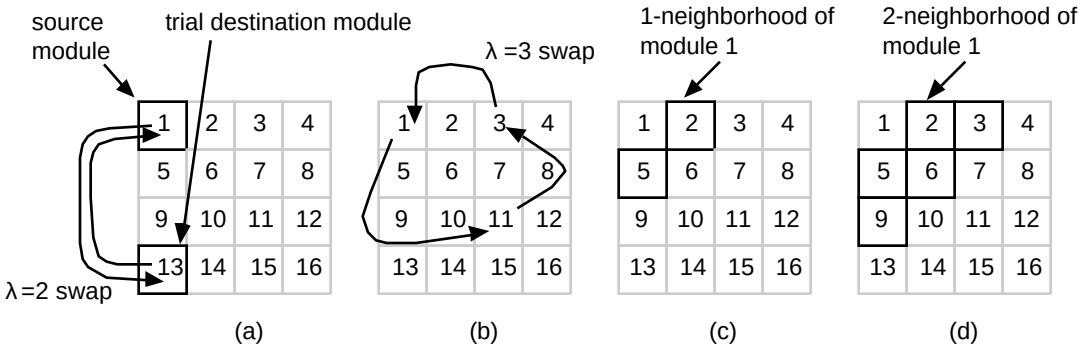
The eigenvalue method takes no account of the logic cell sizes or actual location of logic cell connectors.

(d) A complete layout.

We snap the logic cells to valid locations, leaving room for the routing in the channel.

16.2.6 Iterative Placement Improvement

Key terms and concepts: iterative placement improvement • interchange or iterative exchange • pairwise-interchange algorithm • λ-optimum • neighborhood exchange algorithm • neighborhood • ε-neighborhood • force-directed placement methods • Hooke's law • force-directed interchange • force-directed relaxation • force-directed pairwise relaxation



Interchange.

(a) Swapping the source logic cell with a destination logic cell in pairwise interchange.

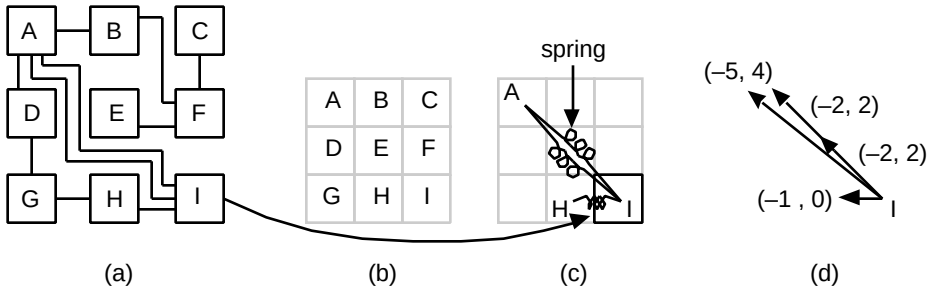
(b) Sometimes we have to swap more than two logic cells at a time to reach an optimum placement, but this is expensive in computation time.

Limiting the search to neighborhoods reduces the search time.

Logic cells within a distance ε of a logic cell form an ε-neighborhood.

(c) A one-neighborhood.

(d) A two-neighborhood.



Force-directed placement.

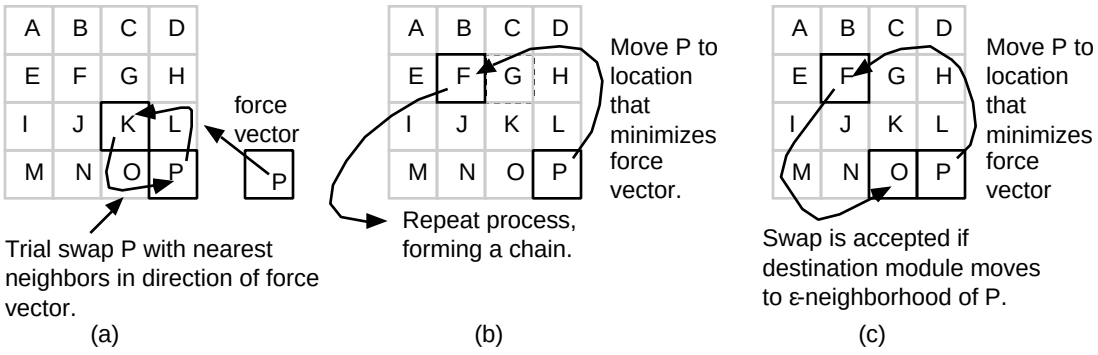
(a) A network with nine logic cells.

(b) We make a grid (one logic cell per bin).

(c) Forces are calculated as if springs were attached to the centers of each logic cell for each connection.

The two nets connecting logic cells A and I correspond to two springs.

(d) The forces are proportional to the spring extensions.



Force-directed iterative placement improvement.

(a) Force-directed interchange.

(b) Force-directed relaxation.

(c) Force-directed pairwise relaxation.

16.2.7 Placement Using Simulated Annealing

Key terms and concepts:

1. Select logic cells for a trial interchange, usually at random.

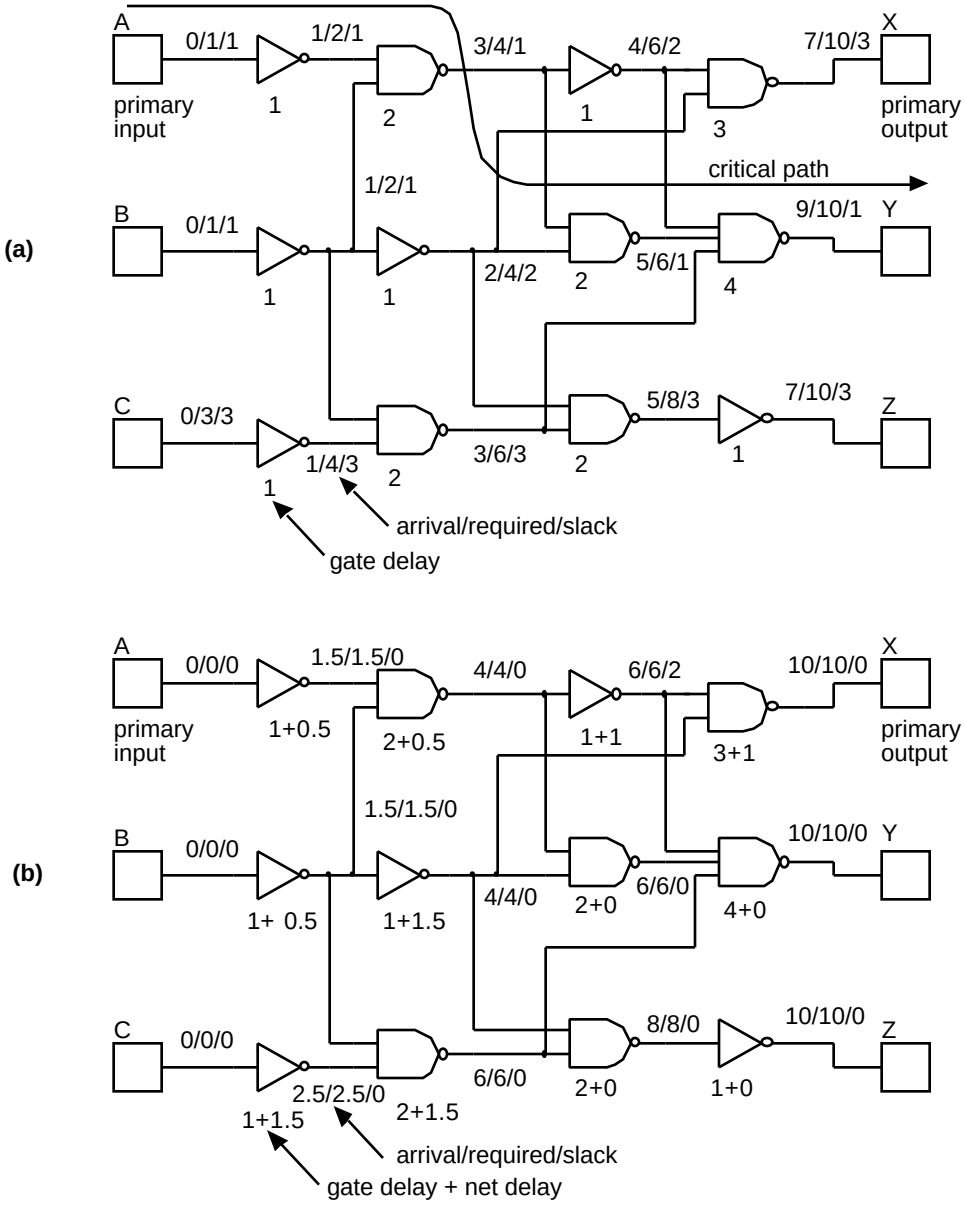
2. Evaluate the objective function E for the new placement.

3. If ΔE is negative or zero, then exchange the logic cells. If ΔE is positive, then exchange the logic cells with a probability of $\exp(-\Delta E/T)$.

4. Go back to step 1 for a fixed number of times, and then lower the temperature T according to a cooling schedule: $T_{n+1}=0.9\,T_n$, for example.

16.2.8 Timing-Driven Placement Methods

Key terms and concepts: zero-slack algorithm primary inputs • arrival times • actual times • required times • primary outputs • slack time

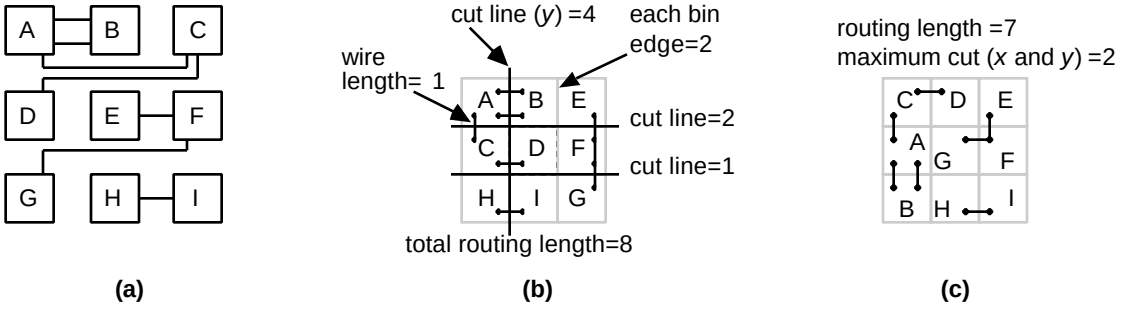


The zero-slack algorithm.

(a) The circuit with no net delays.

(b) The zero-slack algorithm adds net delays (at the outputs of each gate, equivalent to increasing the gate delay) to reduce the slack times to zero.

16.2.9 A Simple Placement Example



(a) **(b)** **(c)**

Placement example.

(a) An example network.

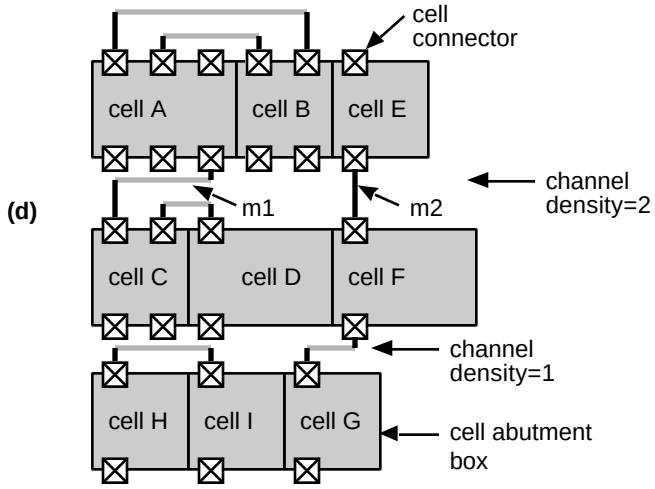
(b) In this placement, the bin size is equal to the logic cell size and all the logic cells are assumed equal size.

(c) An alternative placement with a lower total routing length.

(d) A layout that might result from the placement shown in b.

The channel densities correspond to the cut-line sizes.

Notice that the logic cells are not all the same size (which means there are errors in the interconnect-length estimates we made during placement).



16.3 Physical Design Flow

Key terms and concepts:

Because interconnect delay now dominates gate delay, the trend is to include placement within a floorplanning tool and use a separate router.

1. **Design entry**. The input is a logical description with no physical information.

2. **Initial synthesis**. The initial synthesis contains little or no information on any interconnect loading. The output of the synthesis tool (typically an EDIF netlist) is the input to the floorplanner.

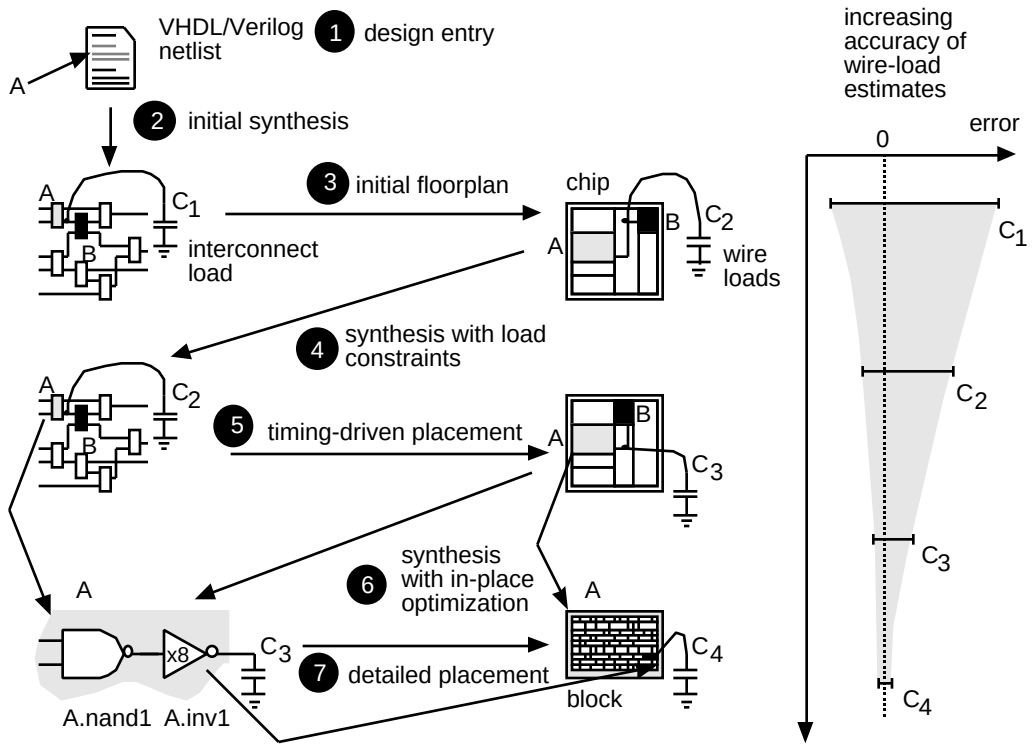
3. **Initial floorplan**. From the initial floorplan interblock capacitances are input to the synthesis tool as load constraints and intrablock capacitances are input as wire-load tables.

4. **Synthesis with load constraints**. At this point the synthesis tool is able to resynthesize the logic based on estimates of the interconnect capacitance each gate is driving. The synthesis tool produces a forward annotation file to constrain path delays in the placement step.

5. **Timing-driven placement**. After placement using constraints from the synthesis tool, the location of every logic cell on the chip is fixed and accurate estimates of interconnect delay can be passed back to the synthesis tool.

6. **Synthesis** with **in-place optimization** (IPO). The synthesis tool changes the drive strength of gates based on the accurate interconnect delay estimates from the floorplanner without altering the netlist structure.

7. **Detailed placement**. The placement information is ready to be input to the routing step.



Timing-driven floorplanning and placement design flow.

16.4 Information Formats

16.4.1 SDF for Floorplanning and Placement

Key terms and concepts: standard delay format (SDF)• back-annotation • forward-annotation • timing constraints

```
  (INSTANCE B) (DELAY (ABSOLUTE
    (INTERCONNECT A.INV8.OUT B.DFF1.Q (:0.6:) (:0.6:))))


  (TIMESCALE 100ps) (INSTANCE B) (DELAY (ABSOLUTE
  (NETDELAY net1 (0.6)))


  (TIMESCALE 100ps) (INSTANCE B.DFF1) (DELAY (ABSOLUTE
    (PORT CLR (16:18:22) (17:20:25))))


(TIMESCALE 100ps) (INSTANCE B) (TIMINGCHECK
  (PATHCONSTRAINT A.AOI22_1.O B.ND02_34.O (0.8) (0.8)))


(TIMESCALE 100ps) (INSTANCE B) (TIMINGCHECK
  (SUM (AOI22_1.O ND02_34.I1) (ND02_34.O ND02_35.I1) (0.8)))


(TIMESCALE 100ps) (INSTANCE B) (TIMINGCHECK
  (DIFF (A.I_1.O B.ND02_1.I1) (A.I_1.O.O B.ND02_2.I1) (0.1)))


(TIMESCALE 100ps) (INSTANCE B) (TIMINGCHECK
  (SKEWCONSTRAINT (posedge clk) (0.1)))
```

16.4.2 PDEF

Key terms and concepts: physical design exchange format (PDEF)

```
(CLUSTERFILE
  (PDEFVERSION "1.0")
  (DESIGN "myDesign")
  (DATE "THU AUG 6 12:00 1995")
```

```
  (VENDOR "ASICS_R_US")
  (PROGRAM "PDEF_GEN")
  (VERSION "V2.2")
  (DIVIDER .)
  (CLUSTER (NAME "ROOT")
    (WIRE_LOAD "10mm x 10mm")
    (UTILIZATION 50.0)
    (MAX_UTILIZATION 60.0)
    (X_BOUNDS 100 1000)
    (Y_BOUNDS 100 1000)
      (CLUSTER (NAME "LEAF_1")
        (WIRE_LOAD "50k gates")
        (UTILIZATION 50.0)
        (MAX_UTILIZATION 60.0)
        (X_BOUNDS 100 500)
        (Y_BOUNDS 100 200)
        (CELL (NAME L1.RAM01)
        (CELL (NAME L1.ALU01)
        )
    )
)
```

16.4.3 LEF and DEF

Key terms and concepts: library exchange format (LEF) • design exchange format (DEF)

16.5 Summary

Key terms and concepts: Interconnect delay now dominates gate delay • Floorplanning is a mapping between logical and physical design • Floorplanning is the center of design operations for all types of ASIC • Timing-driven floorplanning is an essential ASIC design tool • Placement is an automated function