References:

• M. Smith, *Application Specific Integrated Circuits*, Chap. 16
• Cadence Virtuoso User Manual
Top-level IC design process

- Typically done **before** individual circuit block layouts
  - Top-level netlists usually created before any layout

- Create top-level schematic
  - “Components” are functional blocks and I/O pads
  - Blocks include IP and user-created modules

- Create a chip “floor plan” from the schematic
  - Place functional blocks and I/O pads
  - Connections shown as overflows

- Route top-level connections (automatic or interactive)
  - Eliminate overflows, DRC errors, shorts
  - Create layouts of user-designed modules
Chip floorplan

standard-cell area

fixed blocks

I/O pads

0.02 in
500 μm
Modulo-7 counter in pad frame
Floorplanning (Smith text chap. 15, 16)

- **Floorplanning**: arrange major blocks prior to detailed layout to optimize chip area
  - input is a **netlist of circuit blocks** (hierarchical)
    - after system “partitioning” into multiple ICs
  - estimate layout areas, shapes, etc.
    - Flexible blocks – shape can be changed
    - Fixed block – shape/size fixed
- do initial placement of blocks (keep highly-connected blocks close)
- decide location of I/O pads, power, clock
Floorplan a cell-based IC (Fig. 16.6)
- may have to fit into “die cavity” in a package

Initial random floorplan

Blocks moved to improve floorplan

flexible standard-cell blocks (not yet placed)

Flexible standard-cell blocks (with estimated placement)

Heavy congestion below B

Reduced congestion after changes
Congestion analysis (Fig. 16.7)

Initial 2:1.5 die aspect ratio

Altered to 1:1 aspect ratio

A & B resized to reduce congestion

Channel density
Routing a T junction

1. Adjust channel A first.
2. Now we can adjust channel B.

Preferred

Now we cannot adjust channel A.

Constraining
Define channel routing order

- Make “cuts” (slice in two) to separate blocks
- Slicing tree, corresponding to sequence of cuts, determines routing order for channels
  - route in inverse order of cuts
Non-slicing structure

Cyclic constraint prevents channel routing

Cannot find slicing floorplan without increasing chip area

Slicing floorplan possible, but inefficient in use of chip area
Power distribution

Uses special power pads, wires, routing

Option a:
m1 for VSS
m2 for VDD

-----
Potential problems in routing channel

Many layer changes/vias if VDD/VSS on different layers

Option b:
m1 parallel to longest side

-----------
Easier routing but more vias

Array of via contacts for VDD/VSS Buses.
Clock distribution (minimize skew)

Often use “clock tree” structure
MOSIS SCMOS Pad Library

- Includes 6 pad types:
  - Input & output pads with buffers
  - VDD & GND pads with ESD
  - Analog IO pad with ESD
  - Analog reference pad with ESD

- Assemble into a “frame” in which pads butt against each other
  - Allows VDD & GND wires to form a continuous ring
  - Special “spacer” and “corner” pads complete the ring

- ADK tools will generate a pad frame from a schematic
MOSIS
TSMC 0.35um
Hi-ESD
Pad Frame

(l) lambda=0.30um
MOSIS
TSMC 0.35um
Hi-ESD
Pad Frame

Physical layout

Corner pad (passes VDD/GND)

VDD/GND wires form continuous ring through the pad frame

Spacer pad if no signal
MOSIS I/O Pad Schematic

- Bonding Pad
- Inputs to logic ckts
- Output enable
- Outputs from logic ckts
Simplified pad circuit

Enable = 0 (Enable_bar = 1)
- \( Q_2 \) off
- \( Q_4 \) on → pulls \( Q_0 \) to VDD → \( Q_7 \) off
- \( Q_6 \) on → pulls \( Q_6 \) to GND → \( Q_0 \) off

Enable = 1
- \( Q_3 \) on
- \( \overline{Q_3} = Q_0 = \overline{O_{out}} \) → Inverted by \( Q_7/Q_8 \)
MOSIS 1.6 um bidirectional pad

Source: Weste, "CMOS VLSI Design"
ASIC frame + core in Virtuoso

Process:
1. Create "core" block
2. Create pad frame
3. Connect them
Top-level bottom-up design process

- Generate block layouts and for each block:
  - Import the GDSII (or DEF) stream into a Virtuoso library
  - Import the Verilog netlist into the library
  - Perform DRC and LVS on each block until “clean”
  - Create a schematic symbol from the netlist in the library
- Create a block diagram/schematic in Virtuoso “Composer”
  - Create a library for the top-level circuit block and create a schematic view
  - Instantiate schematic symbols from the library
  - Interconnect with nets and add pins
  - Check and save
- Create a layout from the schematic diagram
Top-level block schematic in “Composer”

Figure 12.1: Starting schematic showing the three connected modules
Before module and I/O placement

Blocks initially outside boundary

Figure 12.3: Initial layout before module and I/O placement
After placing modules and pins
Power routing between blocks

Connect power rings
Nets shown as “overflows”
Routed circuit block
Block symbol (to connect to I/O pads)

Figure 12.11: Symbol for the Three Blocks example core
Pad frame with signal wires
Zoomed view of pad frame

Figure 12.13: Pad frame with signal wires (zoomed view)
Schematic: block + pad frame

Figure 12.14: Frame and core components connected together
Placement of frame and core
Power/ground routed manually
Before signal routing
After routing – final layout