ASIC Physical Design
Top-Level Chip Layout

ELEC 5250/6250
Floorplanning (Text chap. 15, 16)

- **Floorplanning**: arrange major blocks prior to detailed layout to minimize chip area
  - input is a netlist of circuit blocks (hierarchical)
    - after system “partitioning”
  - 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
Chip floorplan
Floorplan a cell-based IC (Fig. 16.6)
- must fit into “die cavity” in a package

Initial random floorplan

Blocks moved to improve floorplan

Flexible standard-cell blocks

Heavy congestion below B

Reduced congestion after changes
Congestion analysis (Fig. 16.7)

(a) Congestion map
(b) Trial floorplans

Channel density

Routing congestion:
- 200%
- 100%
- 50%
Routing a T junction

1. Adjust channel A first.

2. Now we can adjust channel B.

1. Adjust channel B first.

2. Now we cannot adjust channel A.
Define channel routing order

- Make “cuts” (slice in two) until individual 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

m1 for VSS
m2 for VDD
Clock distribution

(a) Clock distribution diagram showing main branch, side branches, and base cells. The clock spine and clock driver cell are highlighted.

(b) Detailed view of the clock distribution network with block connector labeled.

(c) Diagram of a clock-driver cell with buffer chain and capacitance values labeled.

(d) Timing diagrams illustrating clock, D2, F1 signals with latency and skew indicated.
Top level layout design flow

- Create a symbol for each core block \( (adk\_daic) \)
- Create a chip-level schematic from core blocks and pads \( (adk\_daic) \)
- Generate design viewpoints \( (adk\_dve) \)
- Create a layout cell for the chip \( (adk\_ic) \)
  - Place core logic blocks from the schematic
  - Generate a pad frame
  - Move/alter core blocks to simplify routing
  - Route pads to core blocks
  - Design rule check & fix problems
- Generate mask data

** Refer to on-line tutorials by Yan/Xu and by Dixit/Poladia
Chip-level schematic (1)

- Generate a symbol for each “core” logic block
  - In DA-IC, open the schematic (eg. \textit{modulo7})
  - Select: \textit{Miscellaneous} $\rightarrow$ \textit{Generate Symbol}
  - Add “\textit{phy\_comp}” property to the symbol
    - Select the \textit{body} of the symbol
    - From the popup menu: \textit{Properties} $\rightarrow$ \textit{Add}
    - Enter property name: \textit{phy\_comp}
    - Enter property value: \textit{mod7b}
      (layout cell name for the block created in IC Station)
  - Check & save

Example on next slide
Symbol with \textit{phy\_comp} property

Layout cell is "mod7b" for logic schematic "modulo7"
Chip-level schematic (2)

- In DA-IC, create a schematic for the chip
  - Instantiate core blocks
    - Menu palette: Add > Instance
    - Select and place generated symbol
  - Add pads from ADK Library > Std. Cells > Pads > tsmc035: In, Out, BiDir, VDD, GND
  - Wire pads to logic blocks and connectors
  - Assign pin numbers, if known
    - Change pad instance name to PINdd (dd = 2-digit pin #)
  - Check & save
- Create design viewpoints with adk_dve

Example on next slide
Assigning PAD pin numbers

Change instance name property on pads to PINxx
xx = 2-digit pin number (01 – 40 for Tiny Chip package)

Place pad on chip pin 01

Default instance names
Top-level schematic for “modulo7” chip
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

- **Inputs to logic circuits**
- **Bonding Pad**
- **Outputs from logic circuits**
Simplified pad circuit

ENABLE = 0  (ENABLE_bar = 1)

- $Q_3$ off
- $Q_4$ on  - pulls $O$ to VDD => $Q_7$ off
- $Q_6$ off  - pulls $O$ to GND => $Q_8$ off

ENABLE = 1

- $Q_3$ on  - $pO = O$ => $\overline{out}$
- $Q_4$ off
- $Q_5$ on
- $Q_6$ off

$\overline{in}$ unbuffed
$in$-bar

$pad$
ADK I/O Pad Schematic

(Configured as output pad)
MOSIS 1.6 μm bidirectional pad

Source: Weste, “CMOS VLSI Design”
Chip layout

• Start IC Station (*adk_ic*) & create a new layout cell
  • enter cell name
  • logic source is “layout” viewpoint of chip schematic
  • same library, process file, rules file, and options as standard cell layout

• Open the schematic
  • ADK Edit menu: *Logic Source > Open*
  • In the schematic, select all core cells (*but not pads*)
  • Place the cells: *Place > Inst*

• Generate the pad frame
  • Top menu bar: *ADK > Generate Padframe > tsmc035*
Chip layout (2)

- Move, rotate, flip core logic cells as desired to make routing easier
  - **BUT - DO NOT EDIT OR MOVE PAD CELLS**

- Autoroute all connections
  - Select autoroute all on P&R menu
  - Click “options” on prompt bar, and unselect “Expand Channels” (prevents pads from being moved)

- Add missing VDD/GND wires, if necessary
  - Autorouter might only route one VDD/GND wire, even if multiple VDD/GND pads are in the schematic
  - Manually add others: *Objects>Add>Path*
    - VDD/GND net width = 50  [Rule of thumb: $\approx 6\text{ma}/1\text{µm width}$]
    - VDD/GND net vias = 6x6 (copy an existing via)
Modulo-7 counter in pad frame
Layout design rule check (DRC)

- Design rules file specified at startup
  
  Ex. `tsmc035.rules`

- From main palette, select ICrules
  
  - Click **Check**
  
  - In prompt box, enter the names of the pad cells in the “Exclude Cell” boxes

    - `PadOut, PadInC, PadGnd, PadVdd, PadNoConnect, Padlesscorner`

Example on next slide
DRC check – exclude Pad cells

Fix any DRC errors – especially with manual wiring