

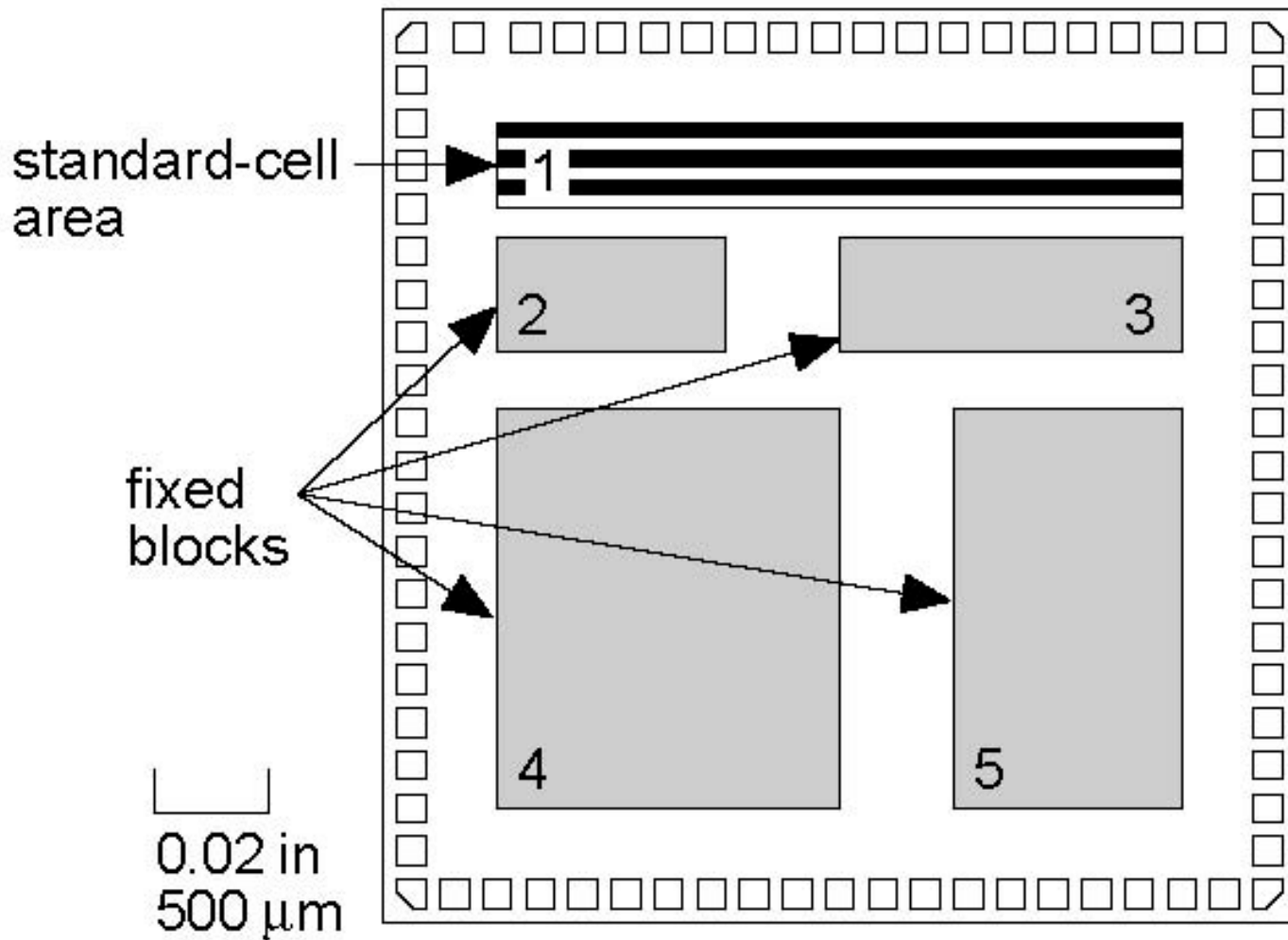
# ASIC Physical Design Top-Level Chip Layout

ELEC 5250/6250

# Floorplanning (Text chap. 15, 16)

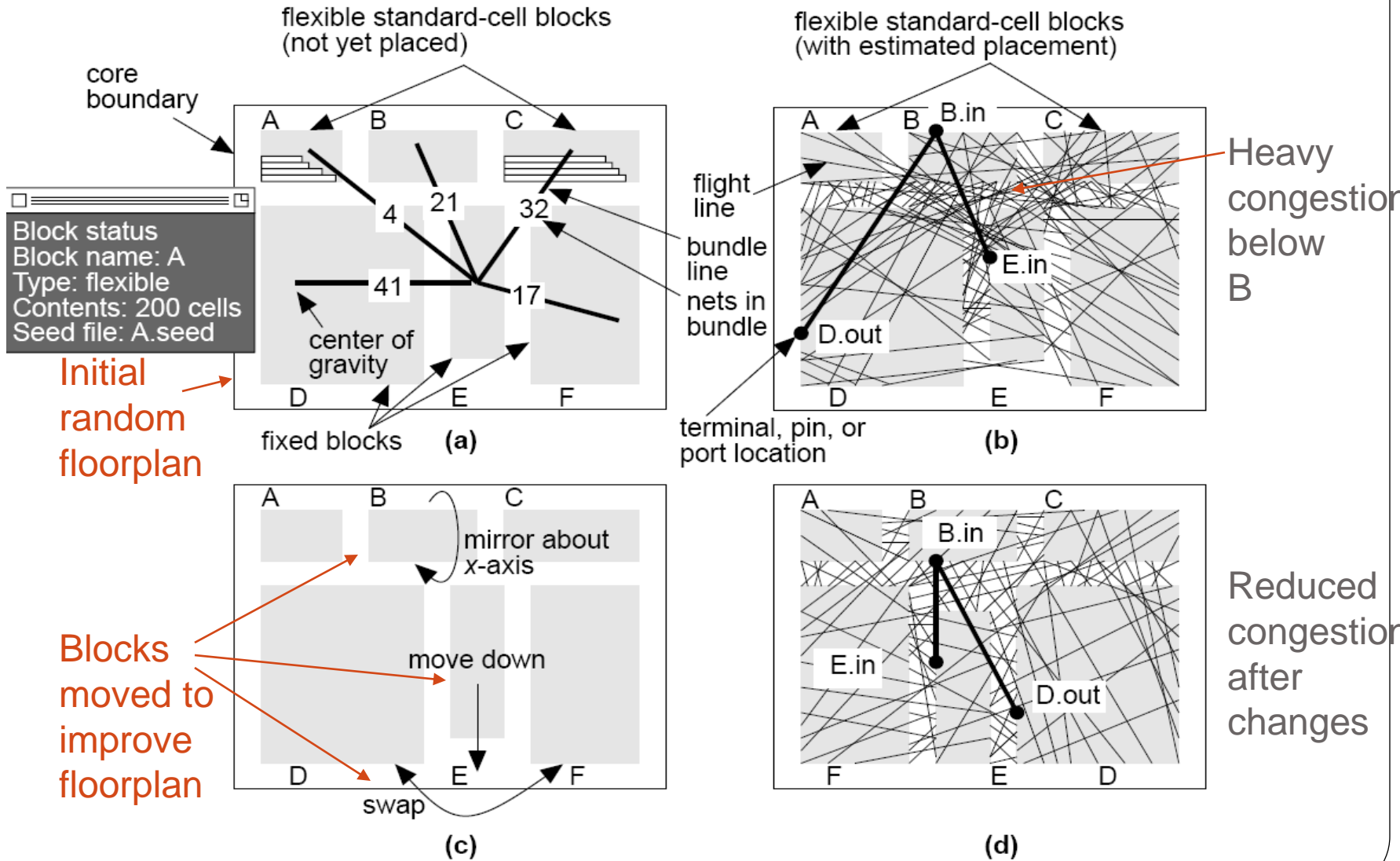
- **Floorplanning:** arrange major blocks prior to detailed layout to minimize chip area
  - input is a netlist of circuit blocks (hierarchical)
    - after system “partitioning”
  - estimate layout areas, shapes, etc.
    - Flexible blocks – shape can be changed
    - Fixed block – shape/size fixed
  - do initial placement of blocks (keep highly-connected blocks close)
  - decide location of I/O pads, power, clock

# Chip floorplan

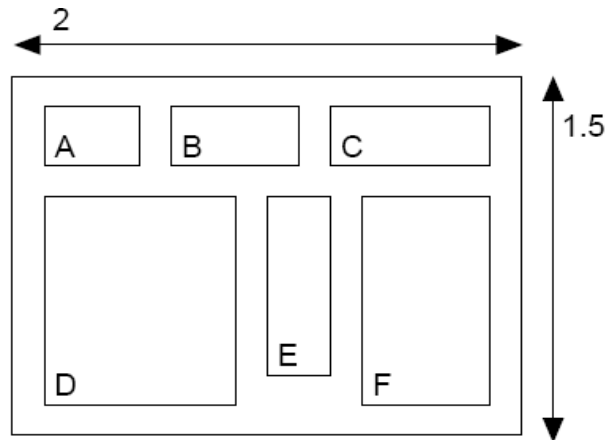


# Floorplan a cell-based IC (Fig. 16.6)

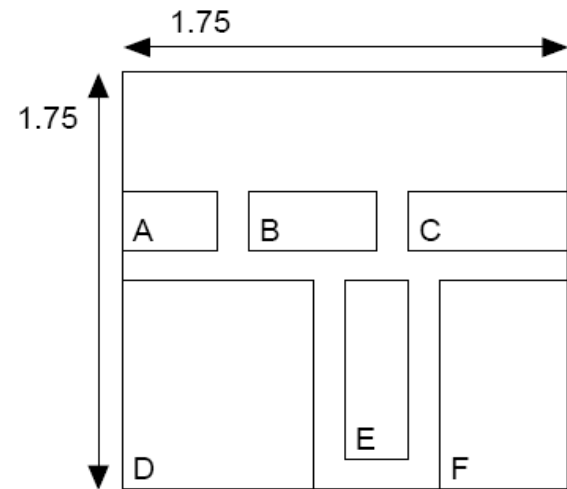
- must fit into "die cavity" in a package



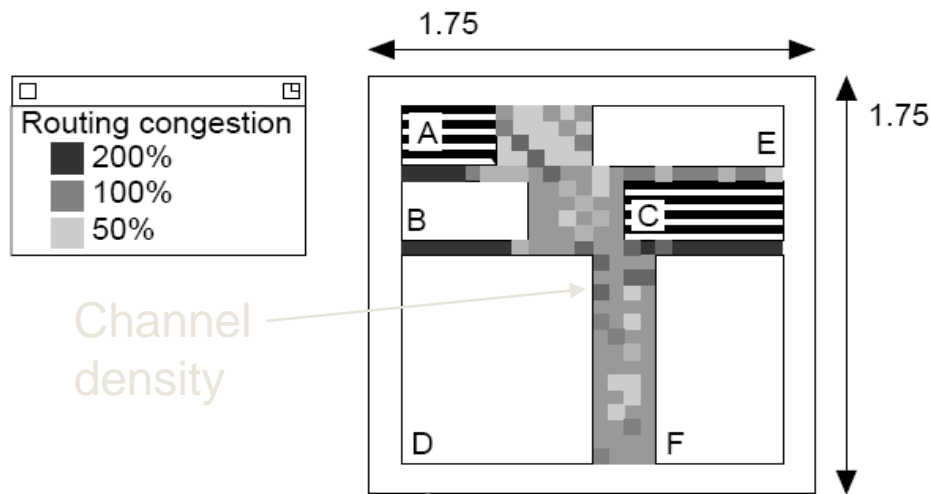
# Congestion analysis (Fig. 16.7)



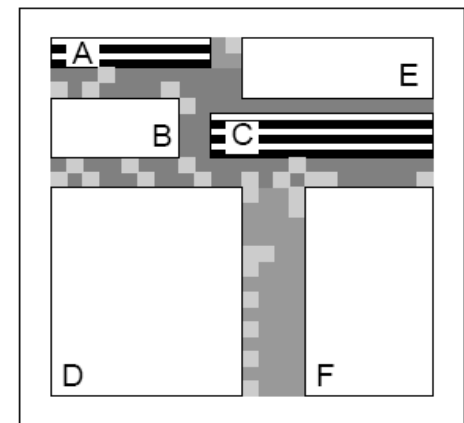
(a)



(b)

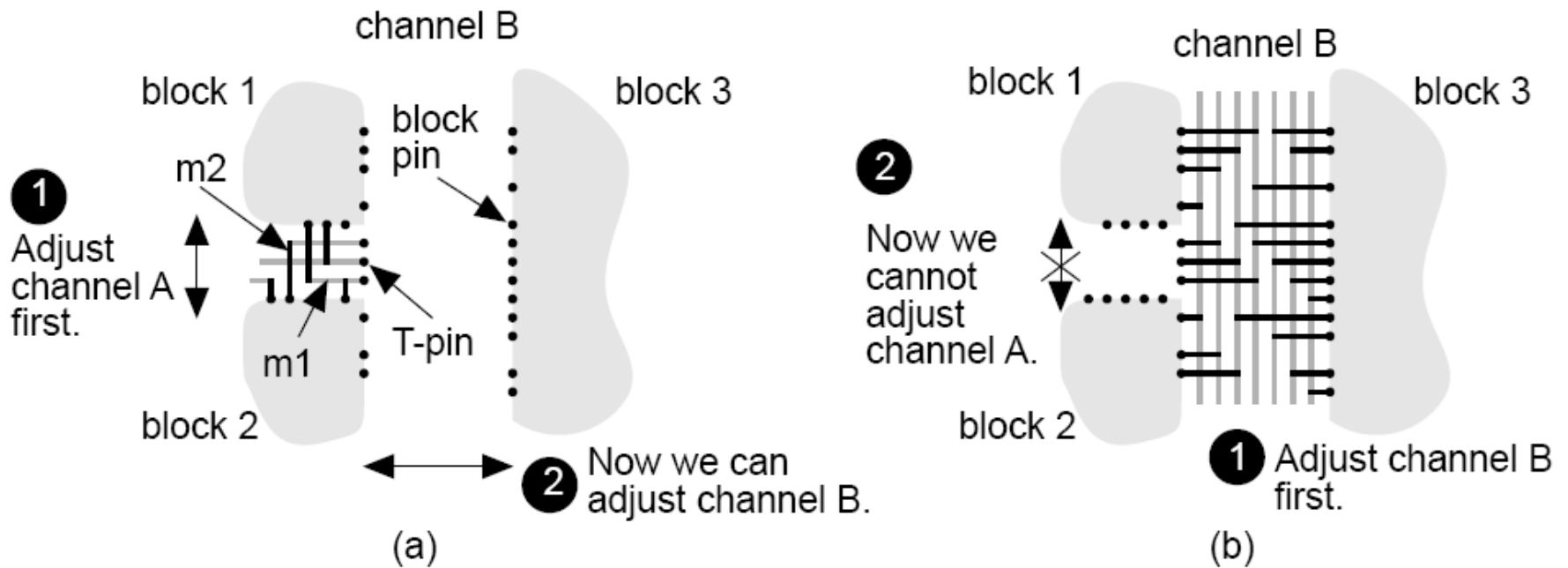


(c)

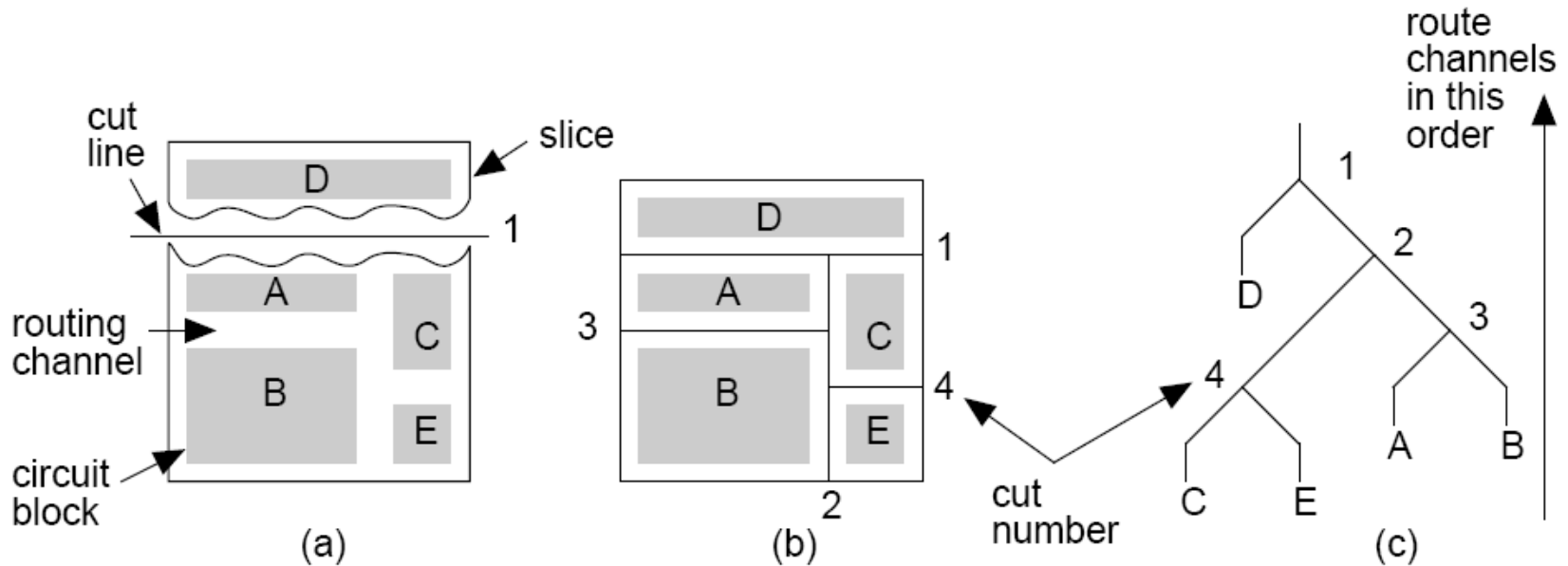


(d)

# Routing a T junction

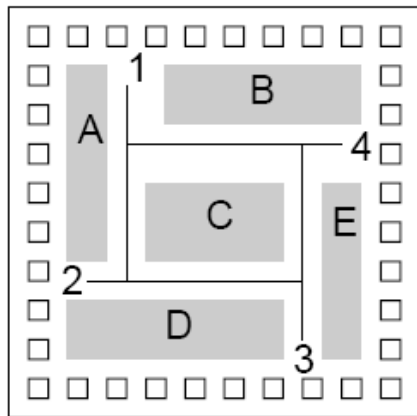


# Define channel routing order



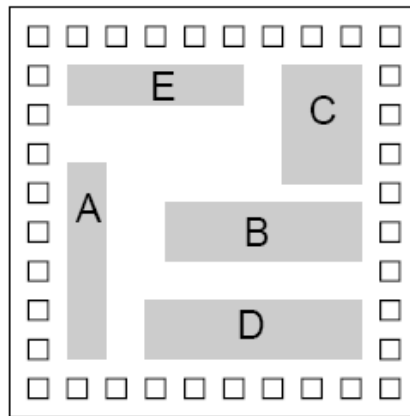
- Make “cuts” (slice in two) until individual blocks
- Slicing tree, corresponding to sequence of cuts determines routing order for channels
  - route in inverse order of cuts

# Non-slicing structure



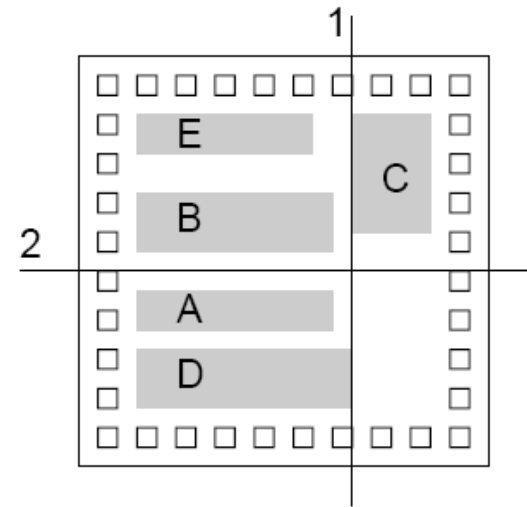
(a)

Cyclic constraint prevents channel routing



(b)

Cannot find slicing floorplan without increasing chip area

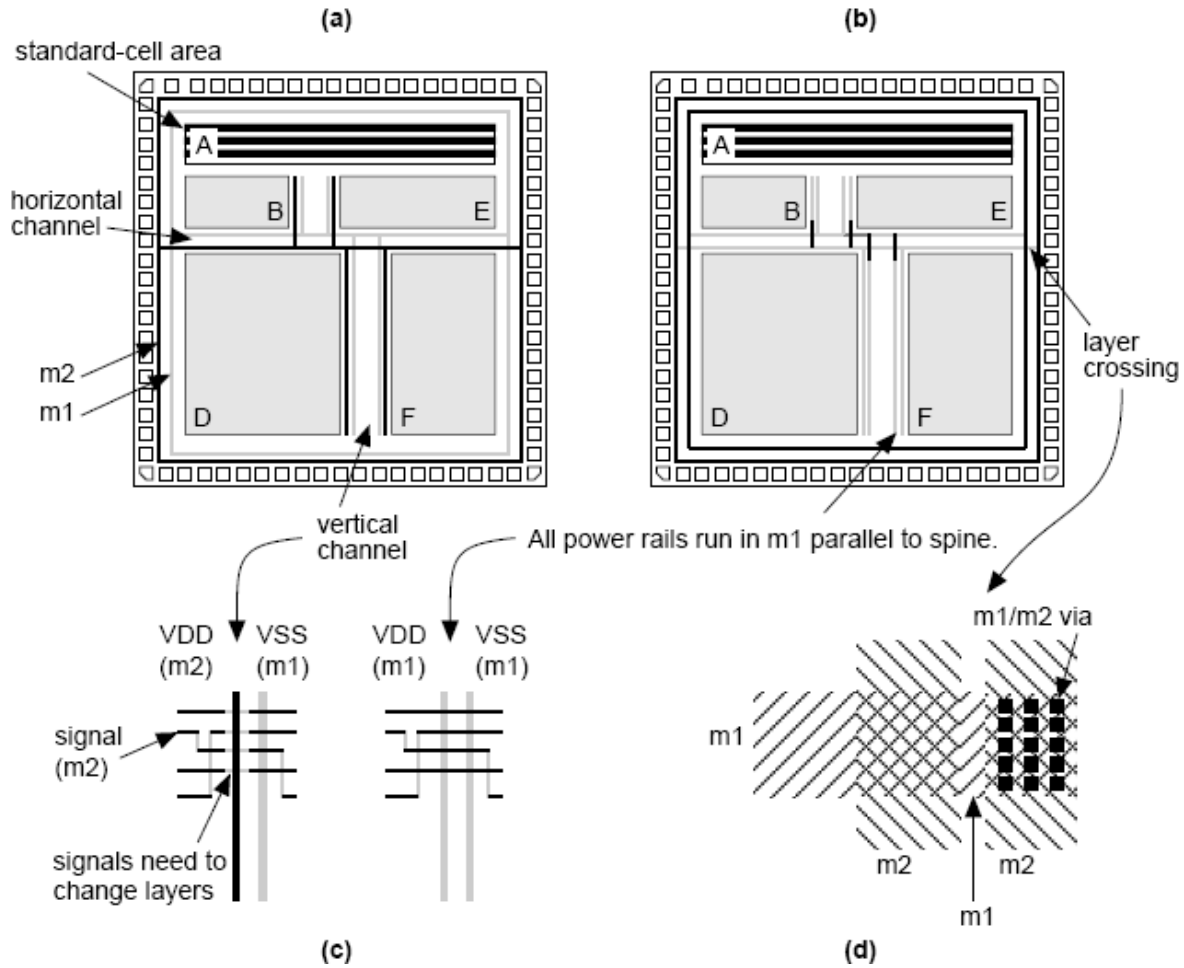


(c)

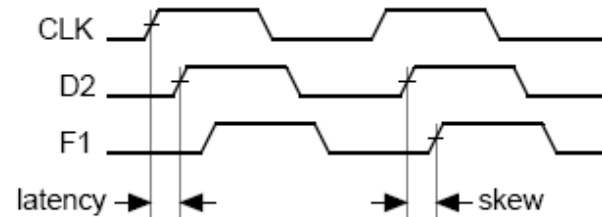
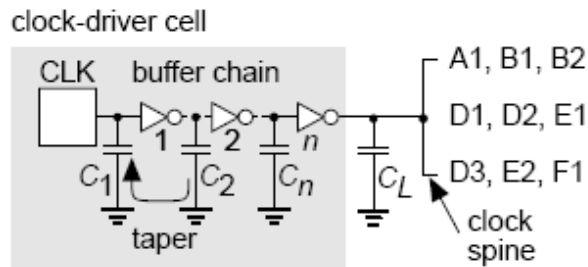
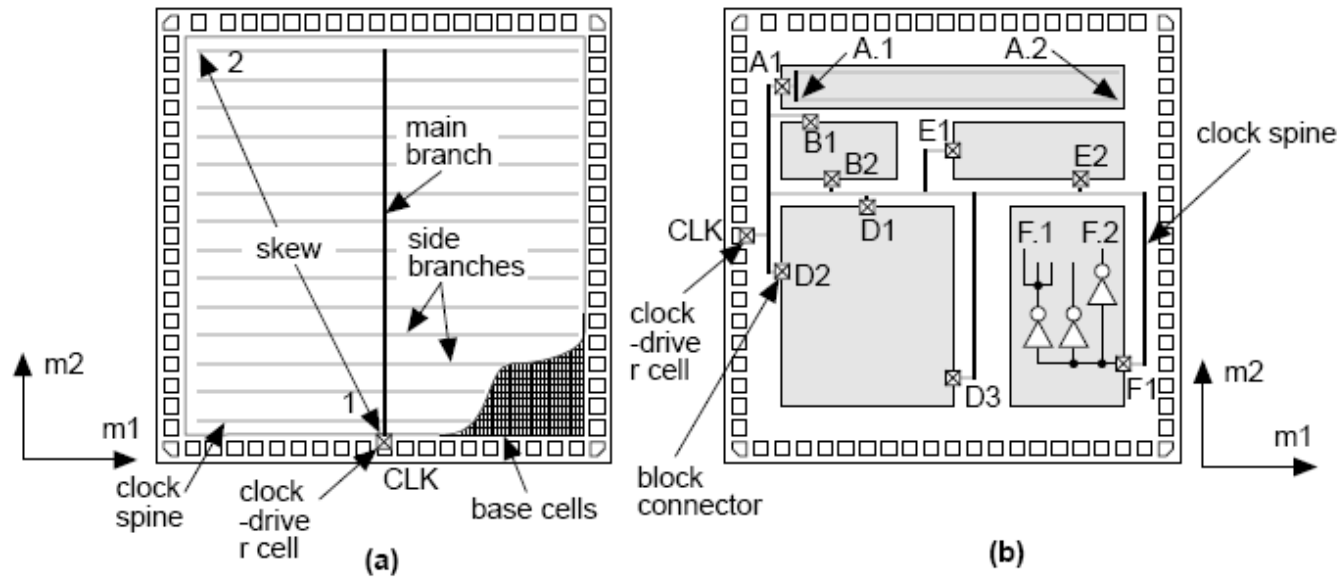
Slicing floorplan possible, but inefficient in use of chip area

# Power distribution

m1 for VSS  
m2 for VDD



# Clock distribution



(c)

(d)

# 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

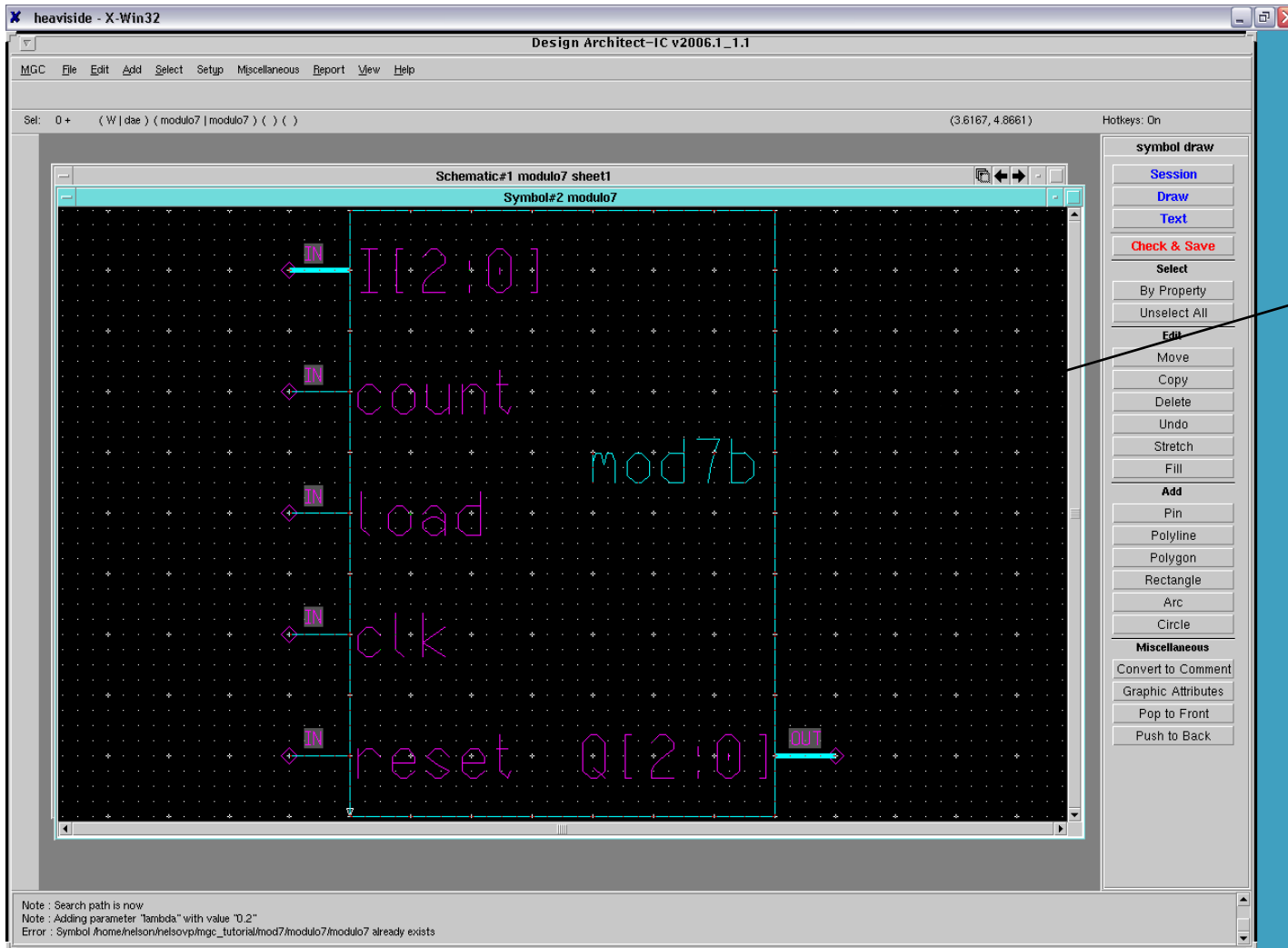
\*\* Refer to on-line tutorials by Yan/Xu and by Dixit/Poladia

# Chip-level schematic (1)

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

Example on next slide

# Symbol with *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 pallete: [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](#)

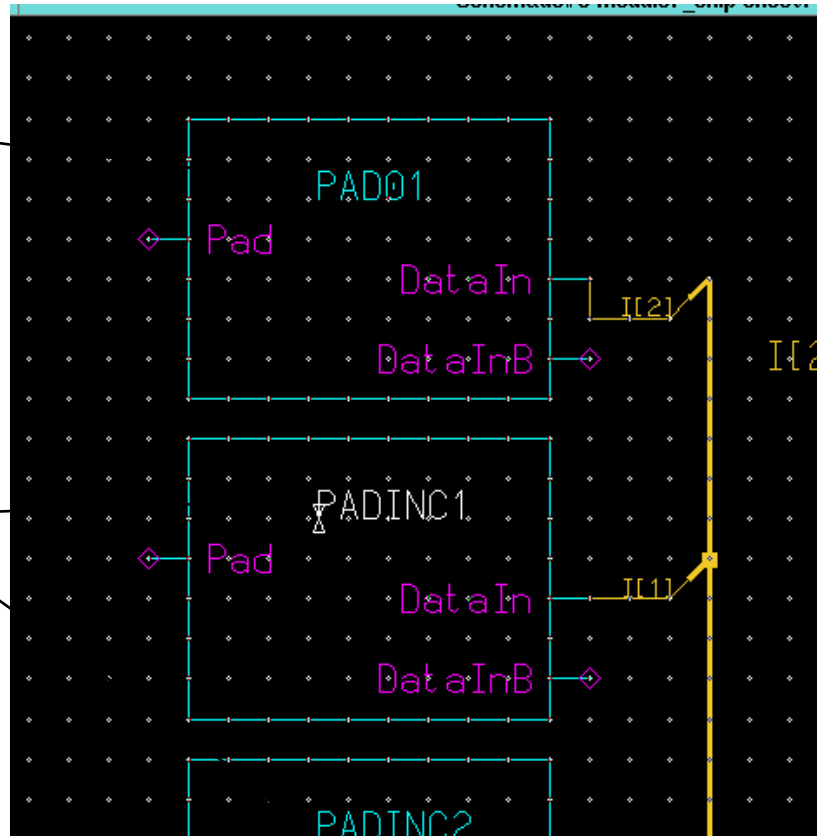
Example on next slide

# Assigning PAD pin numbers

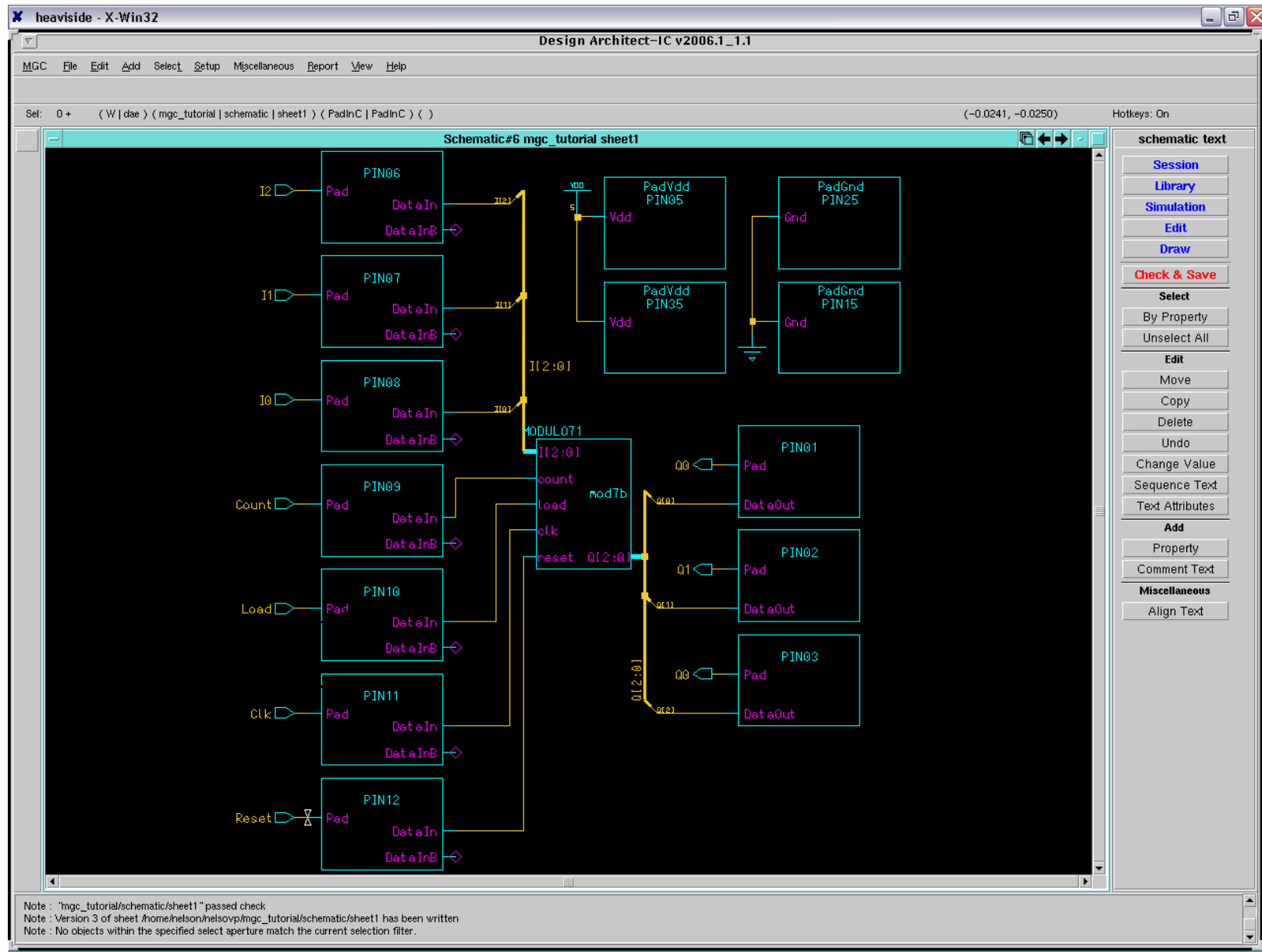
Change instance name property on pads to PINxx  
xx = 2-digit pin number (01 – 40 for Tiny Chip package)

Place pad on  
chip pin 01

Default  
instance  
names



# Top-level schematic for “modulo7” chip

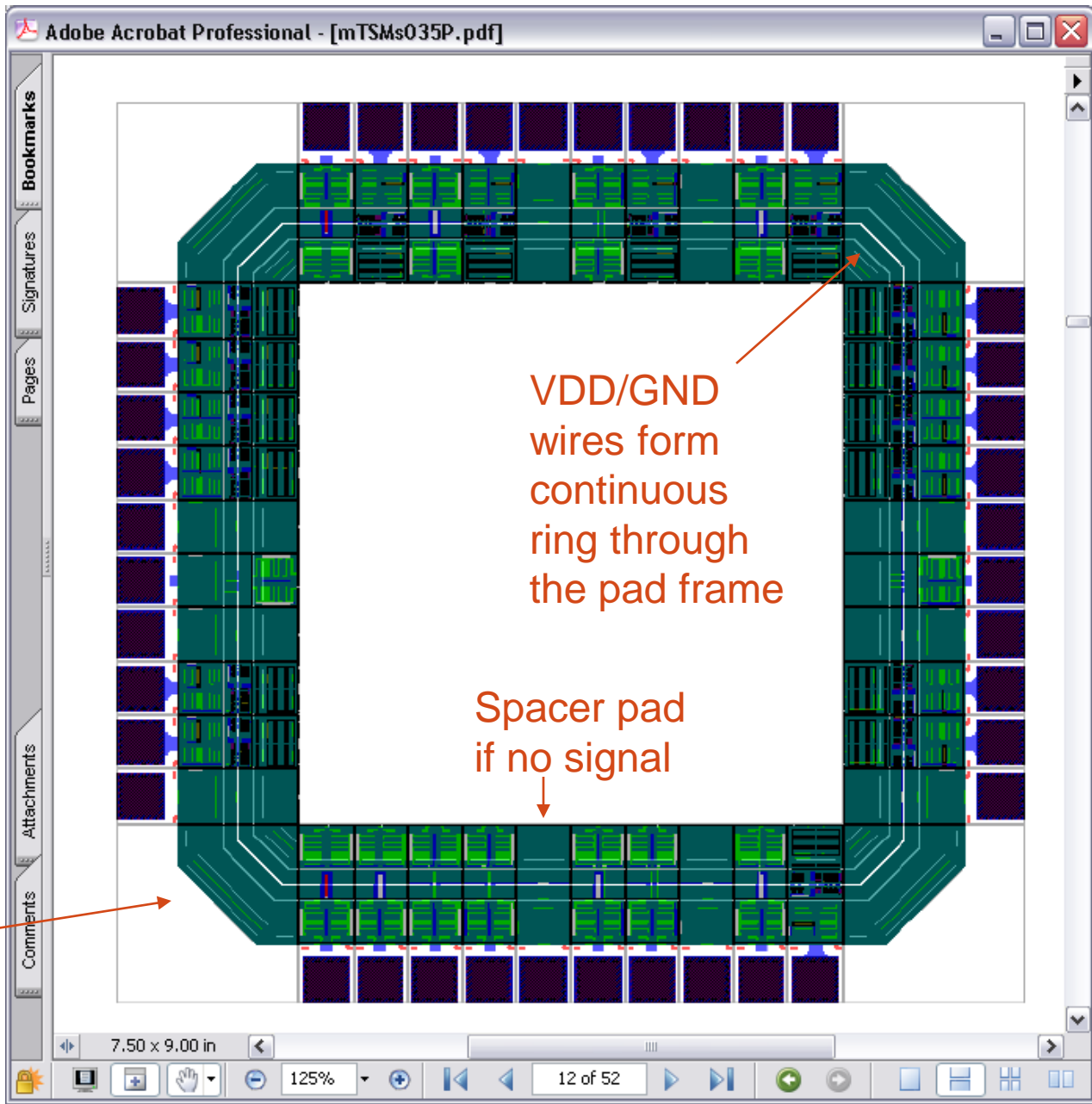


# MOSIS SCMOS Pad Library

<http://www.mosis.com/cell-libraries/scn035-pads-tiny/mTSMs035P.pdf>

- 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



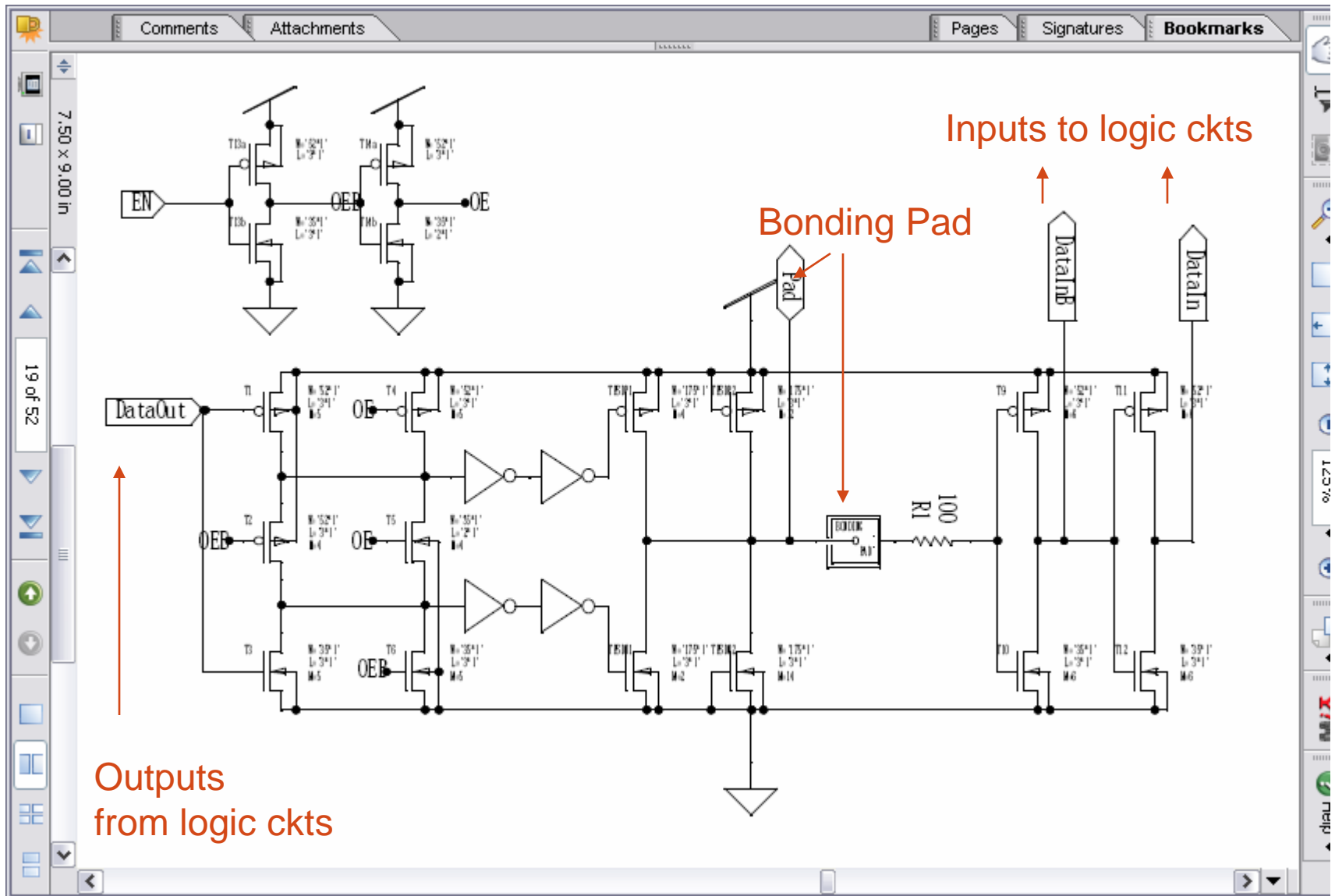


MOSIS  
TSMC 0.35um  
Hi-ESD  
Pad Frame

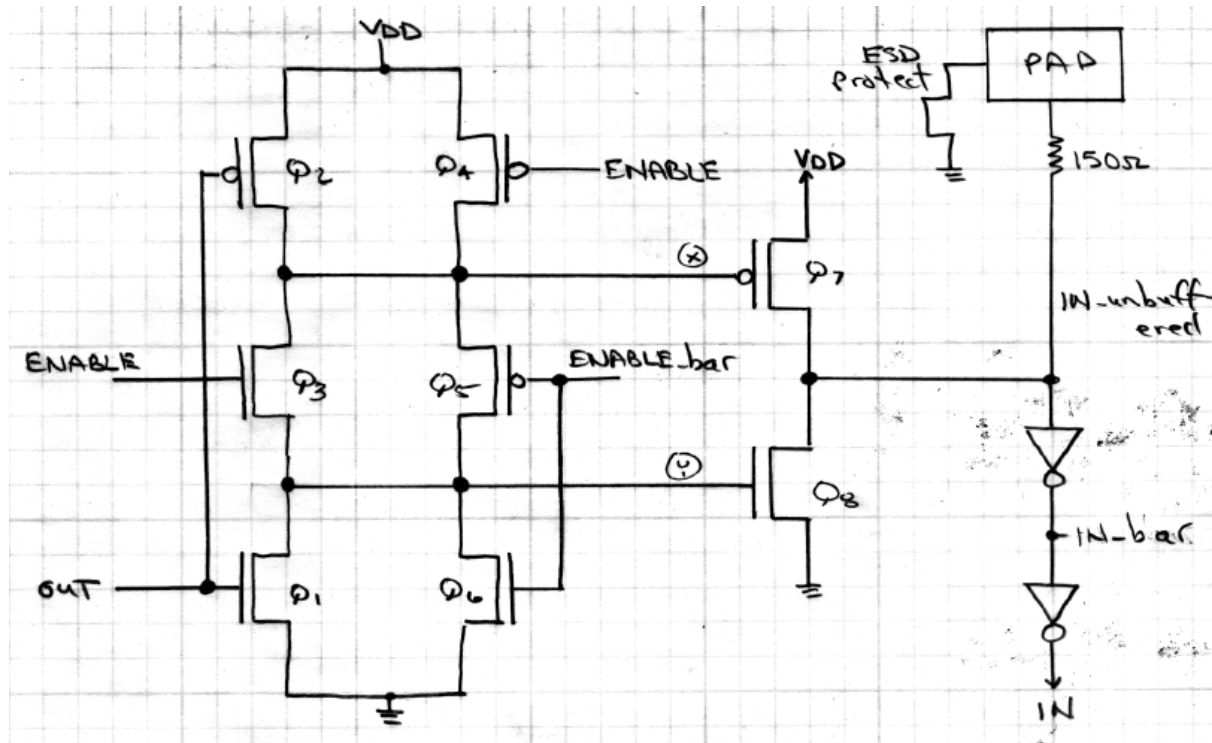
Physical layout

Corner pad  
(passes VDD/GND)

# MOSIS I/O Pad Schematic



# Simplified pad circuit



ENABLE = 0 (ENABLE\_bar = 1)

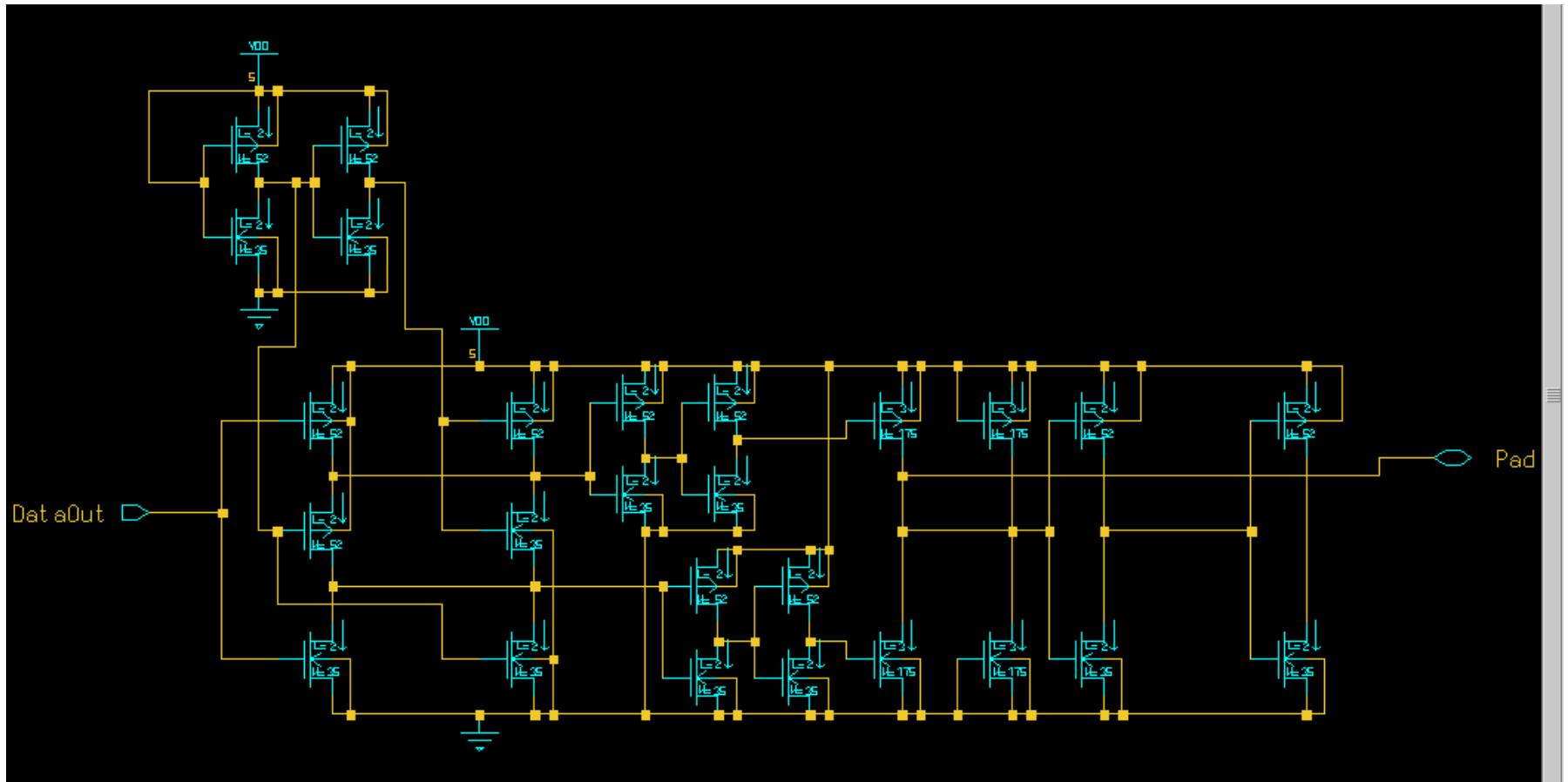
- Q3 off
- Q4 on - pulls X to VDD  $\Rightarrow$  Q7 off
- Q5 off
- Q6 on - pulls Y to GND  $\Rightarrow$  Q8 off

ENABLE = 1

- Q3 on  $\Rightarrow$  X = Y = OUT  $\Rightarrow$  Inverted by Q7/Q8
- Q4 off
- Q5 on
- Q6 off

# ADK I/O Pad Schematic

(Configured as output pad)



# MOSIS 1.6 $\mu\text{m}$ bidirectional pad

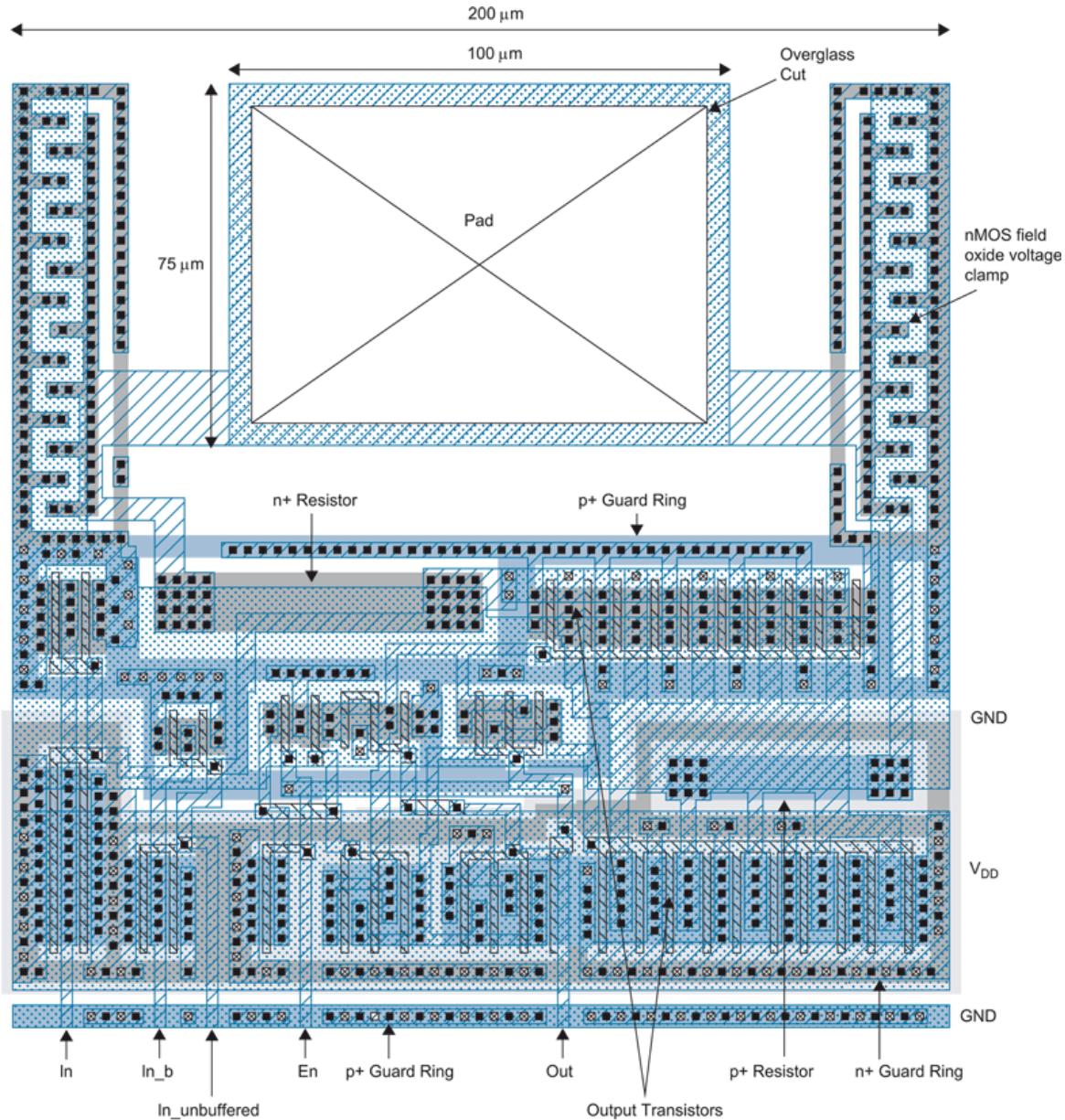


FIG 12.23 MOSIS 1.6  $\mu\text{m}$  bidirectional pad. Color version on inside front cover.

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*

Continued

# 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:  $\approx 6\text{ma}/1\mu\text{m width}$ ]
    - VDD/GND net vias = 6x6 (copy an existing via)



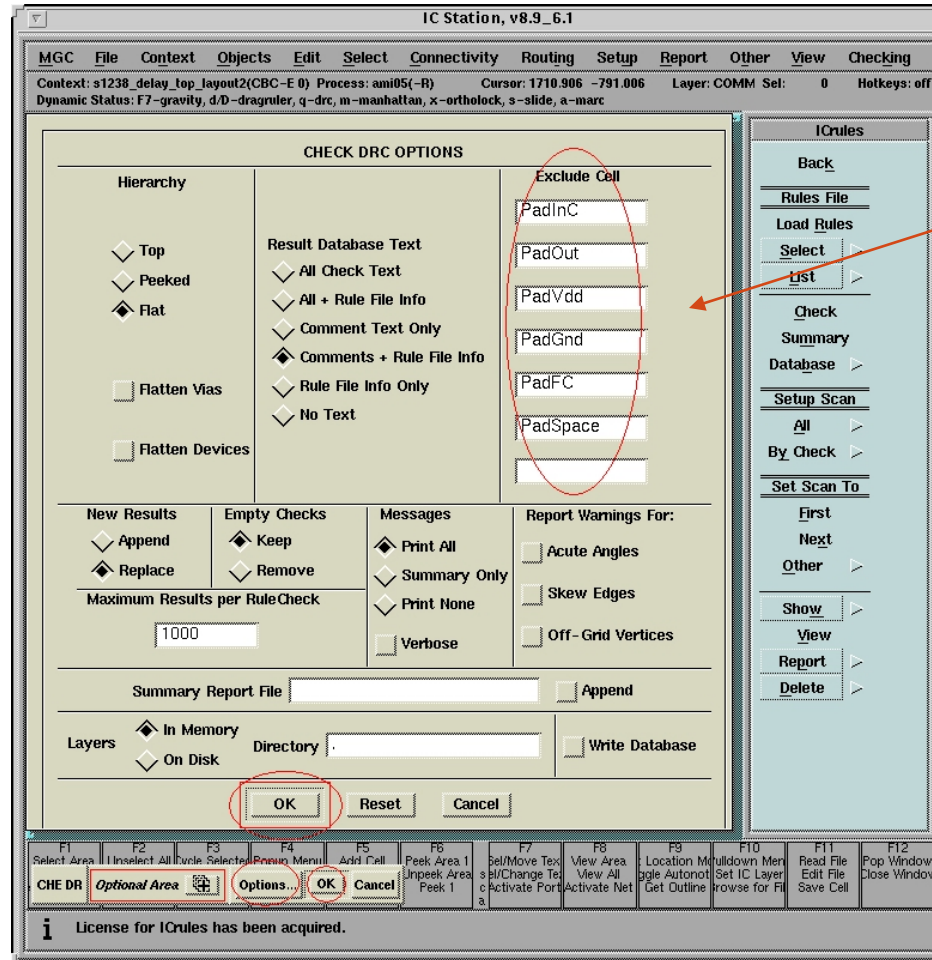
# Layout design rule check (DRC)

- 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

DRC options



Cells to exclude

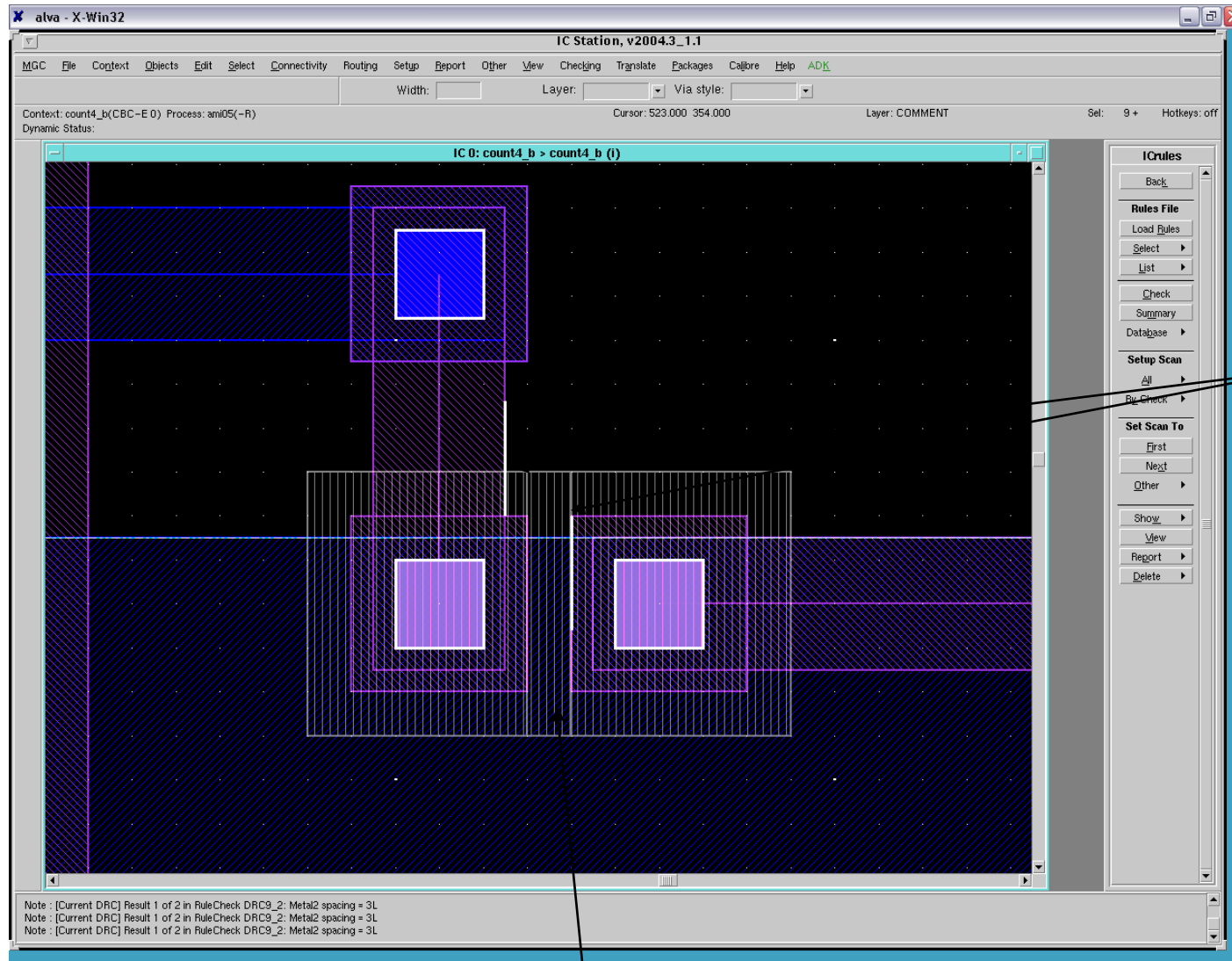
License for ICrules has been acquired.

# Fix errors detected by DRC

- To fix, click on *First* in palette to highlight first error
  - Error is highlighted in the layout
  - Click *View* to zoom in to the error (see next)
  - Example: DRC9\_2: Metal2 spacing = 3L
  - Fix by drawing a rectangle of metal2 to fill in the gap between contacts that should be connected
- Click *Next* to go to next error, until all are fixed

**NOTE:** The layout must be free of DRC errors if MOSIS is to fabricate the chip; they will run their own DRC.

# Error: DRC9\_2 metal2 spacing = 3L



Draw  
rectangle  
of metal2  
to fill gap

It also called contact-to-contact metal 2 spacing DRC9\_2 error