ASIC Physical Design
Top-Level Chip Layout

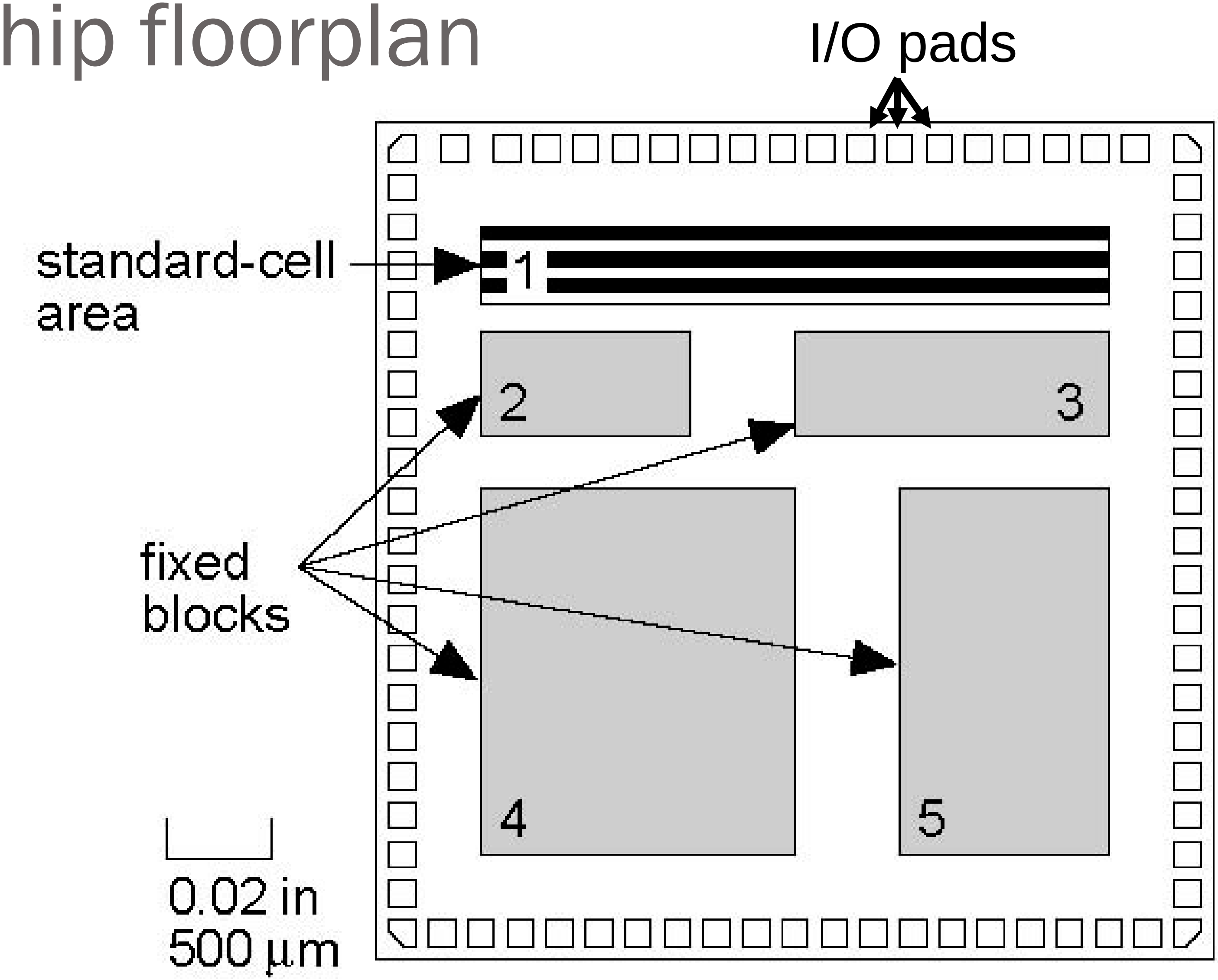
References:

- M. Smith, *Application Specific Integrated Circuits*, Chap. 16
- Cadence Virtuoso User Manual

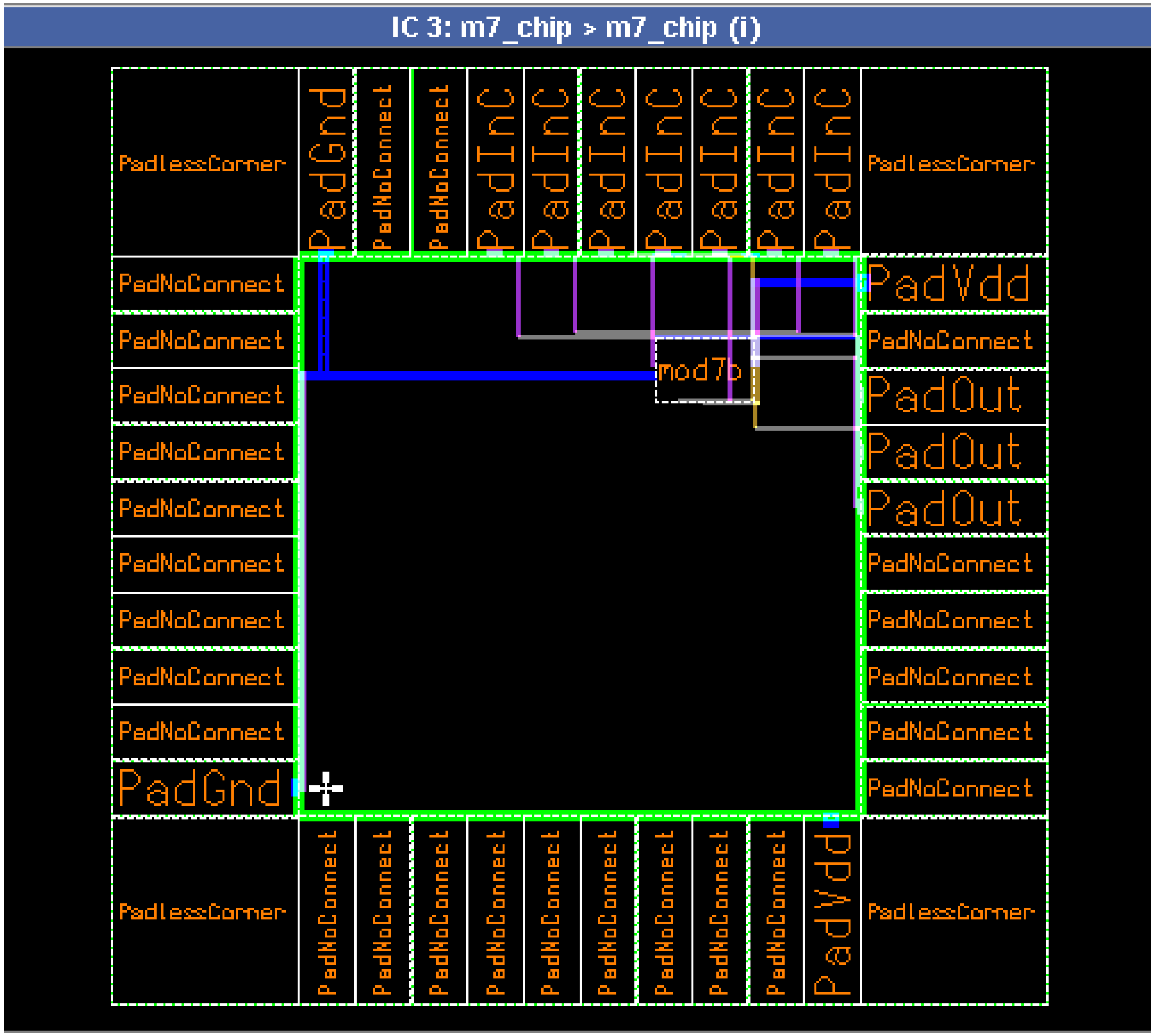
Top-level IC design process

- 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



Modulo-7 counter in pad frame

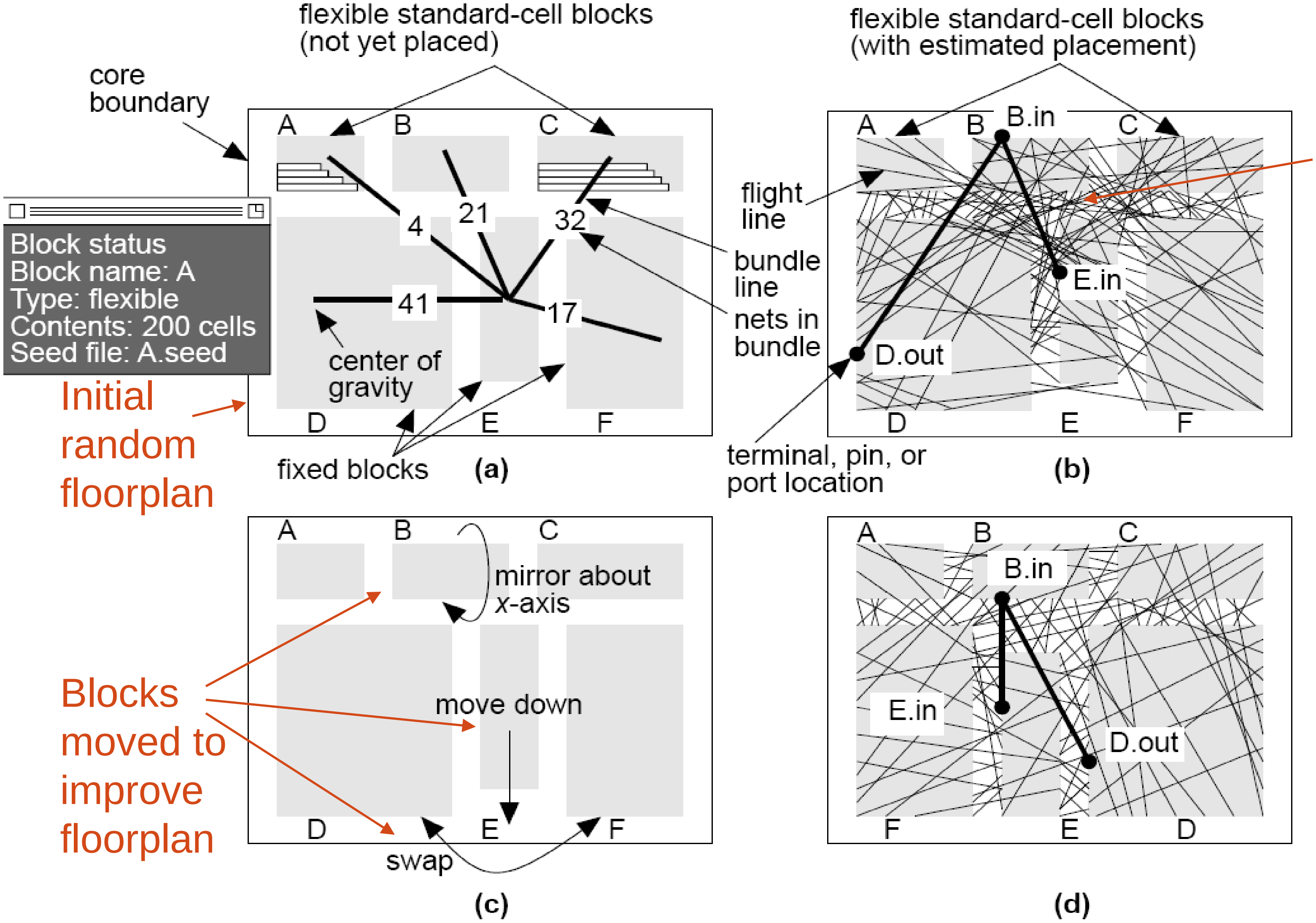


Floorplanning (Smith text chap. 15, 16)

- 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



Block status
Block name: A
Type: flexible
Contents: 200 cells
Seed file: A.seed

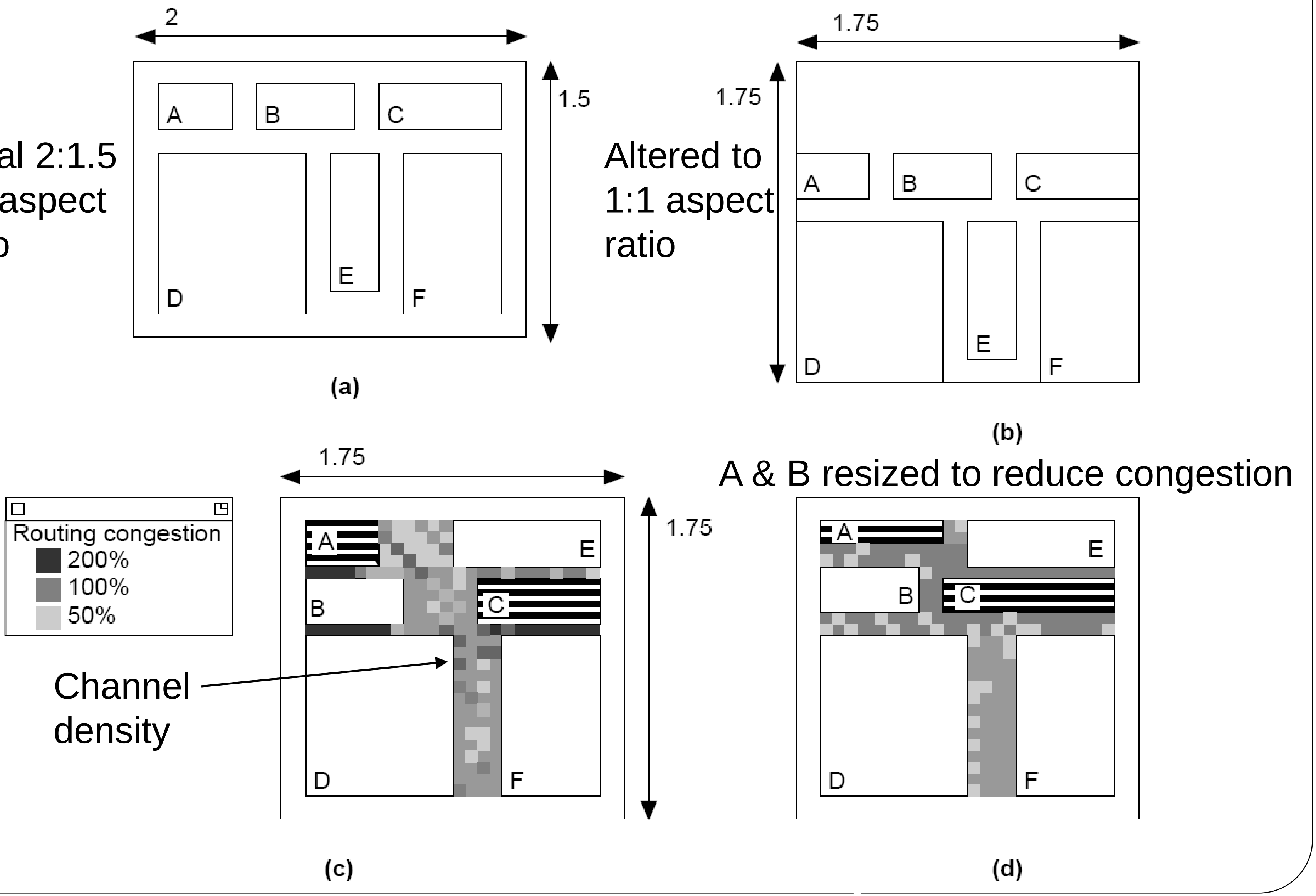
Initial random floorplan

Blocks moved to improve floorplan

Heavy congestio below B

Reduced congestio after changes

Congestion analysis (Fig. 16.7)



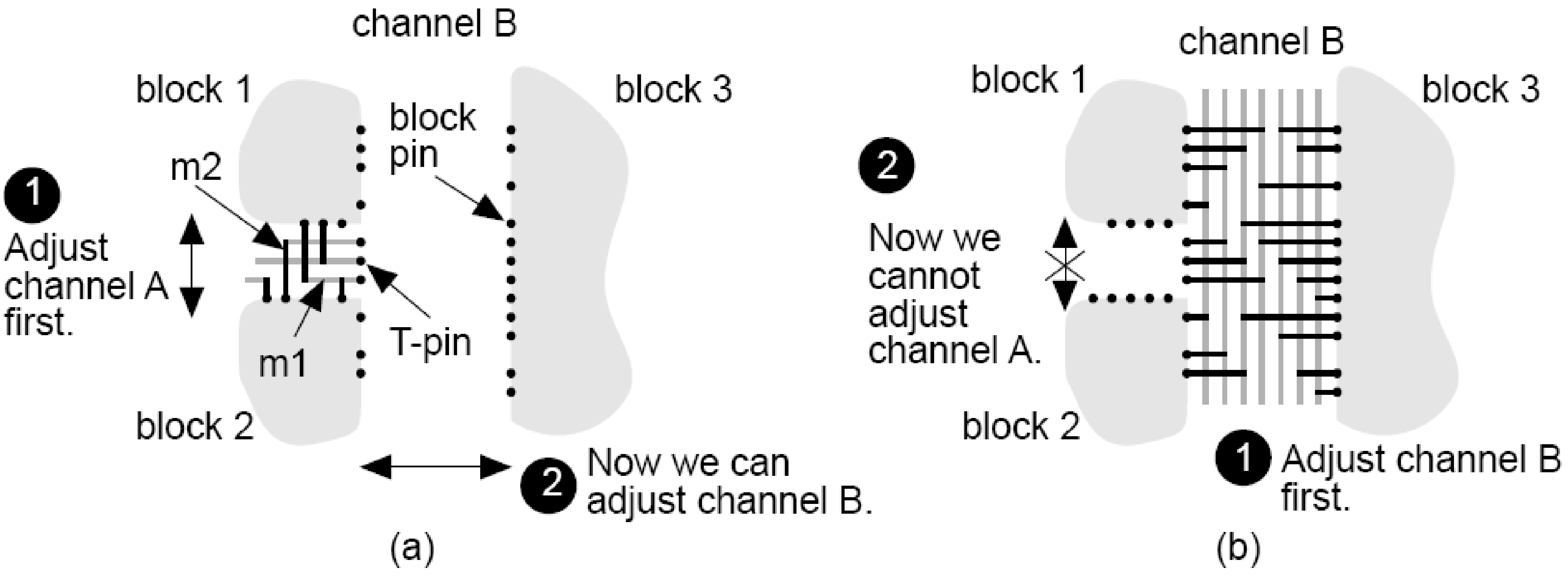
Initial 2:1.5 die aspect ratio

Altered to 1:1 aspect ratio

A & B resized to reduce congestion

Channel density

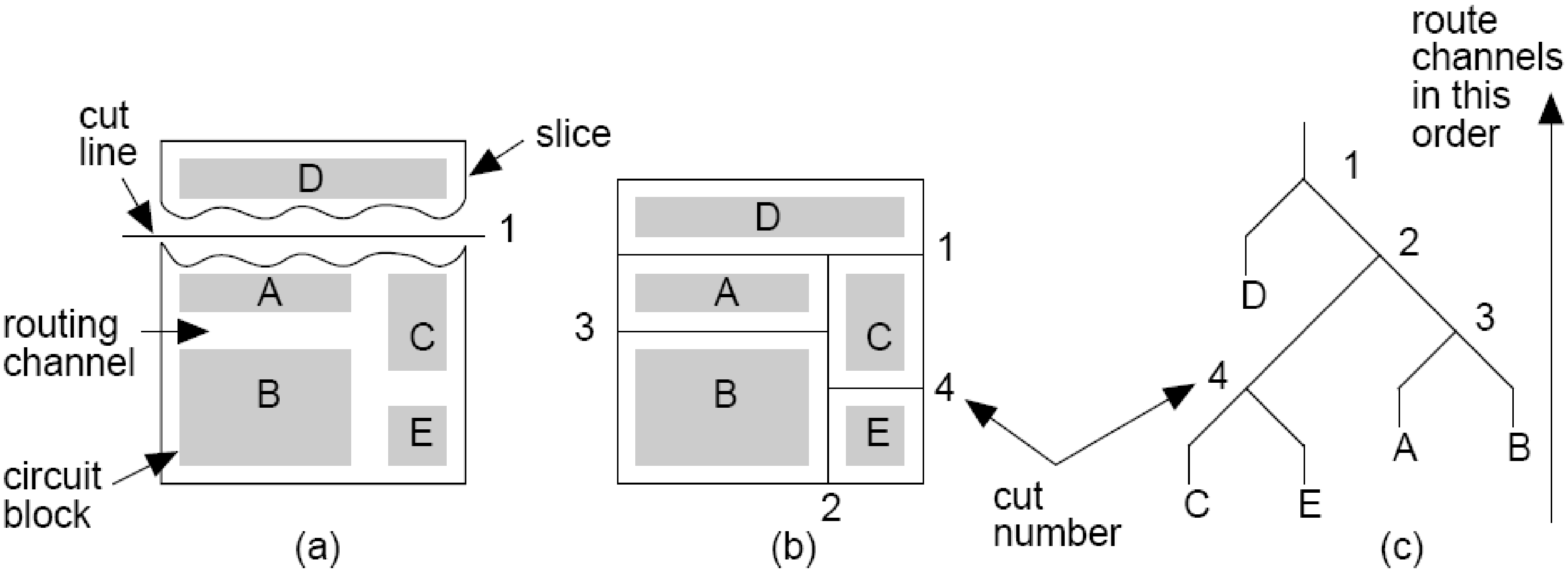
Routing a T junction



Preferred

Constraining

Define channel routing order

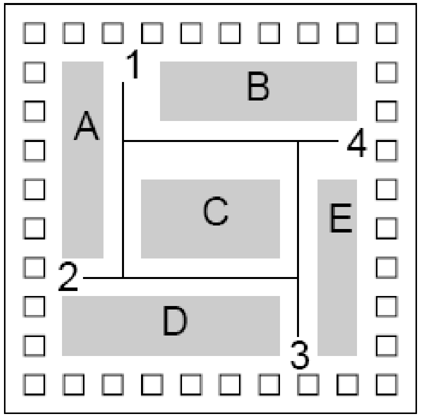


•Make “cuts” (slice in two) to separate blocks

•Slicing tree, corresponding to sequence of cuts, determines routing order for channels

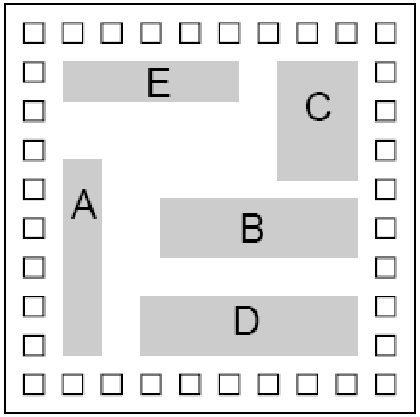
- route in inverse order of cuts

Non-slicing structure



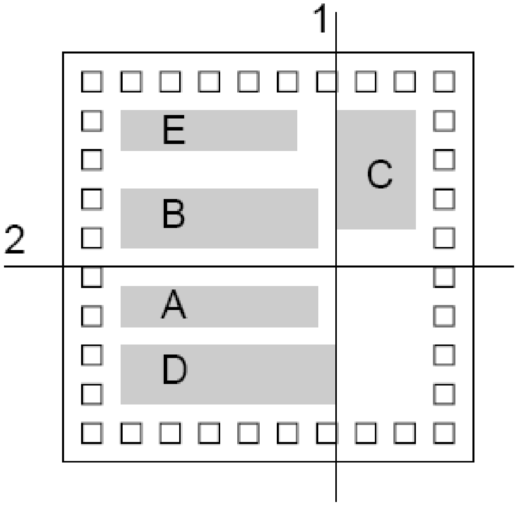
(a)

Cyclic constraint prevents channel routing



(b)

Cannot find slicing floorplan without increasing chip area



(c)

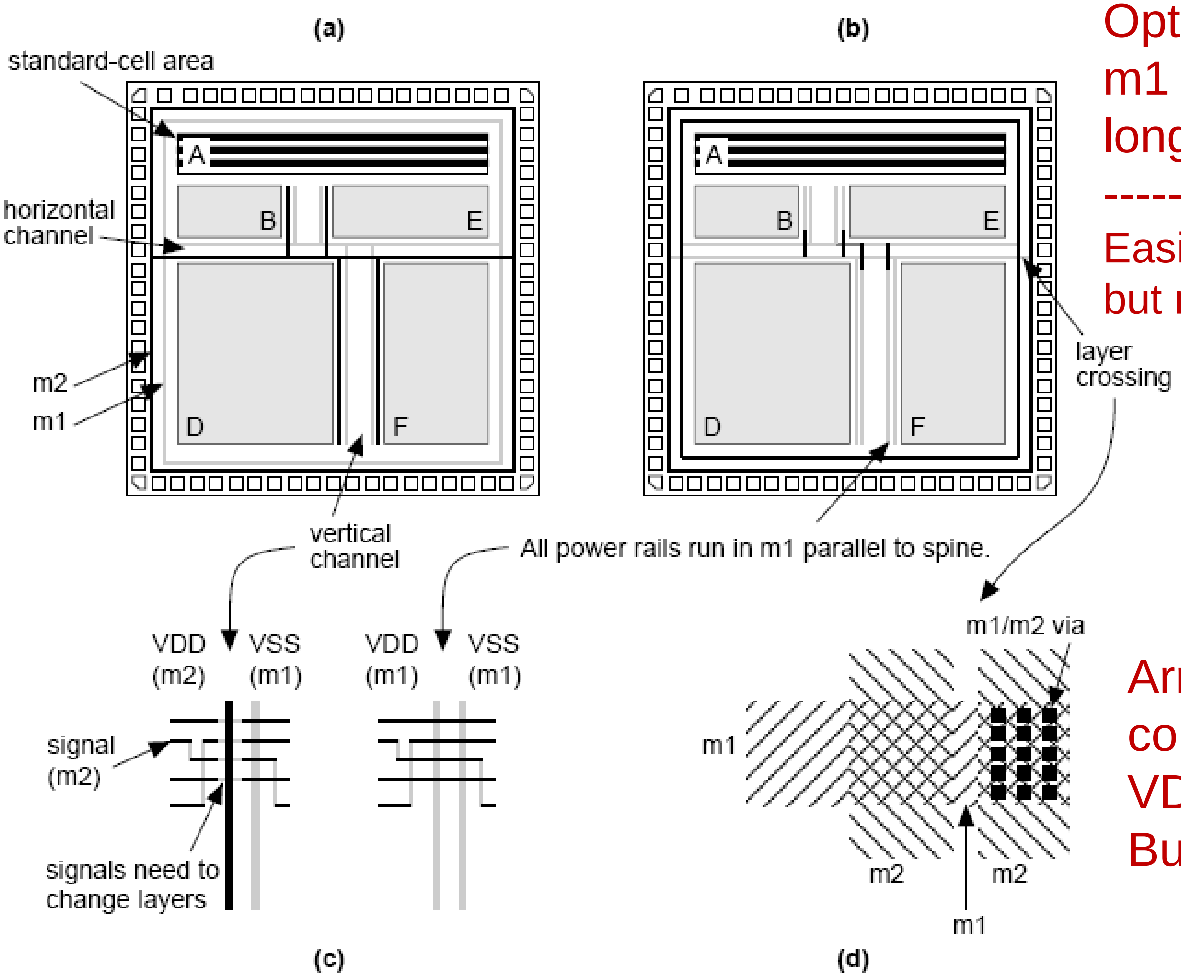
Slicing floorplan possible, but inefficient in use of chip area

Power distribution

Uses special power pads, wires, routing

Option a:
m1 for VSS
m2 for VDD

Potential problems in routing channel



Option b:
m1 parallel to longest side

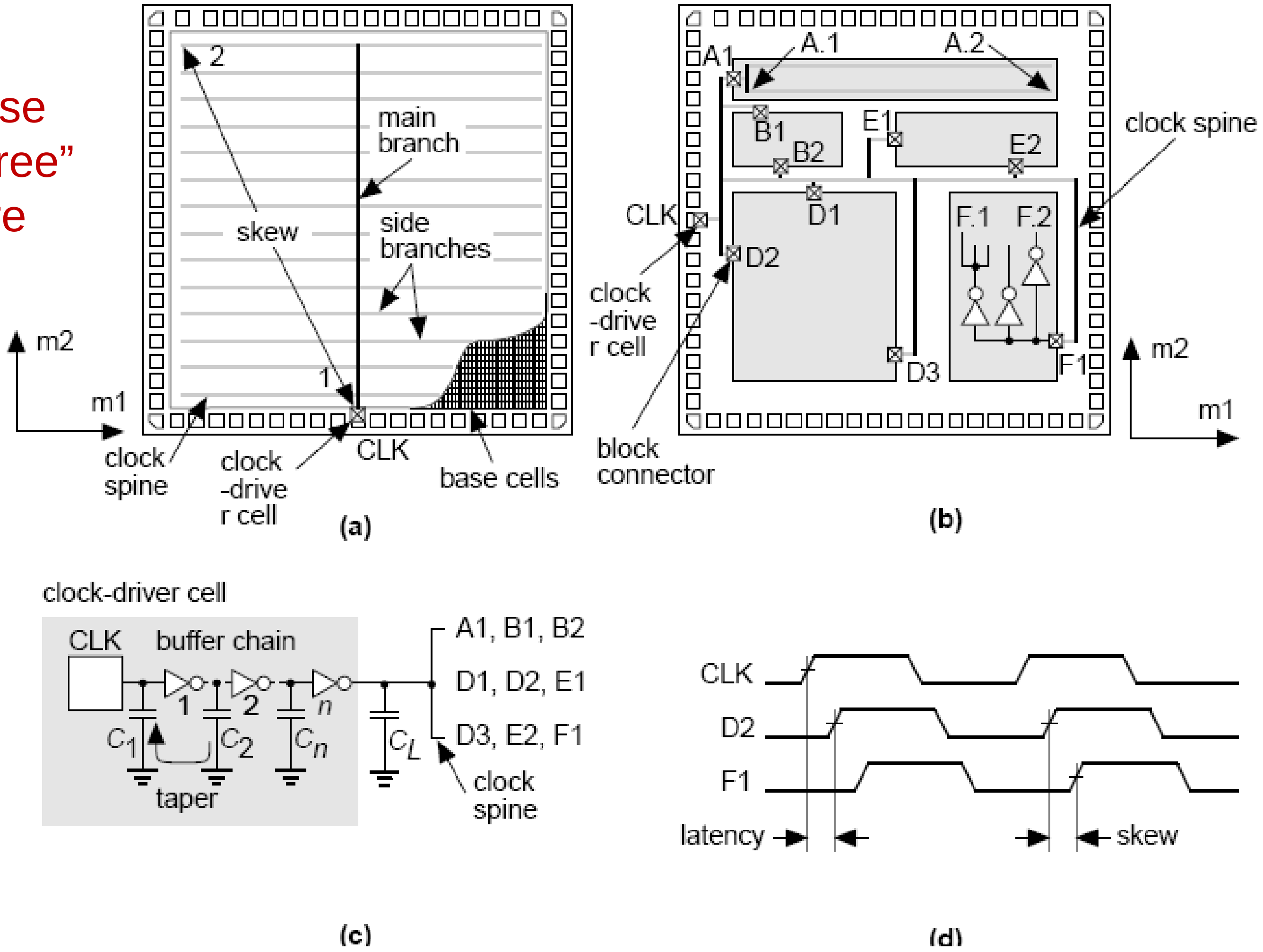
Easier routing but more vias

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

Array of via contacts for VDD/VSS Buses.

Clock distribution (minimize skew)

Often use “clock tree” structure

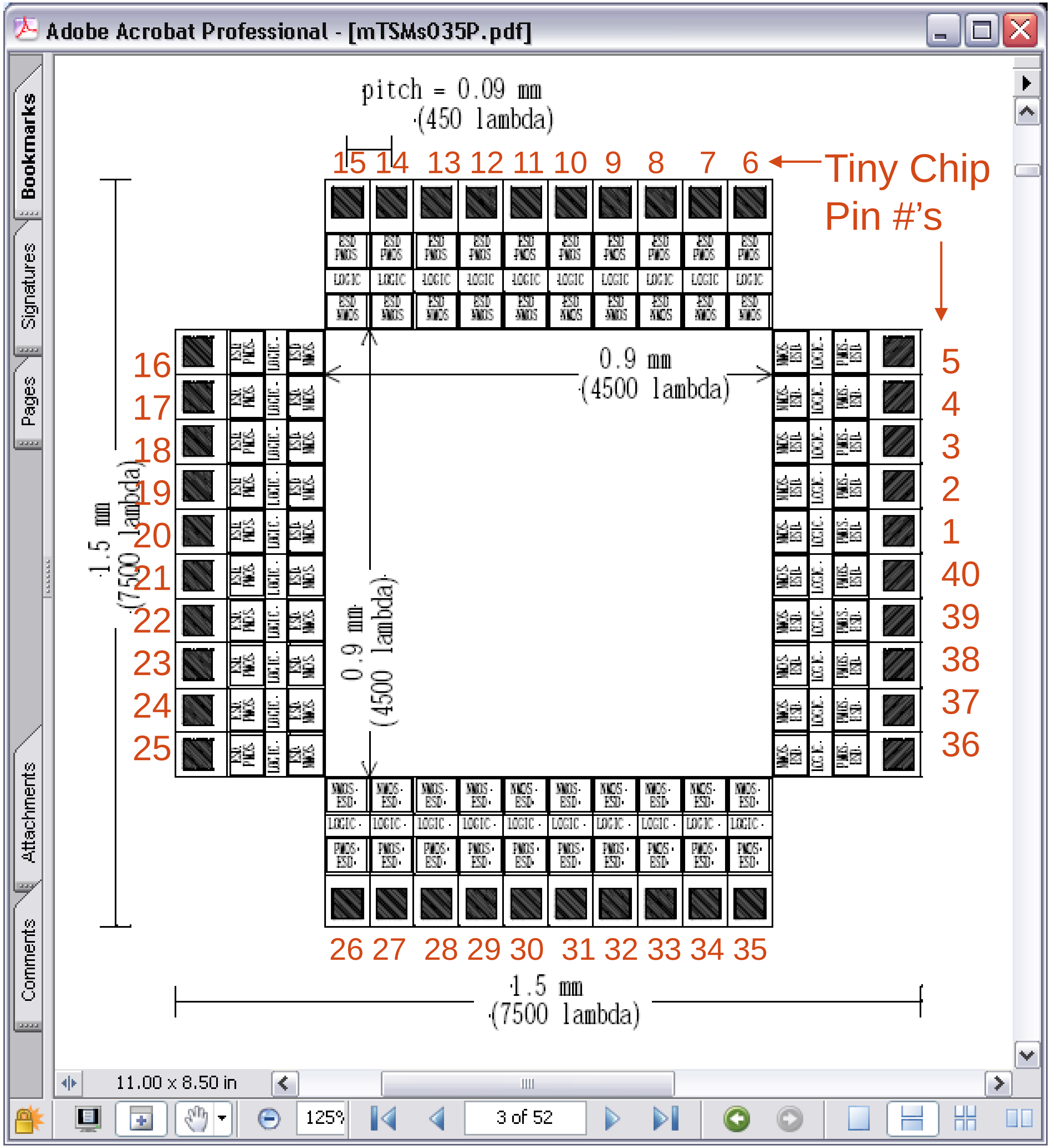


MOSIS SCMOS Pad Library

- 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

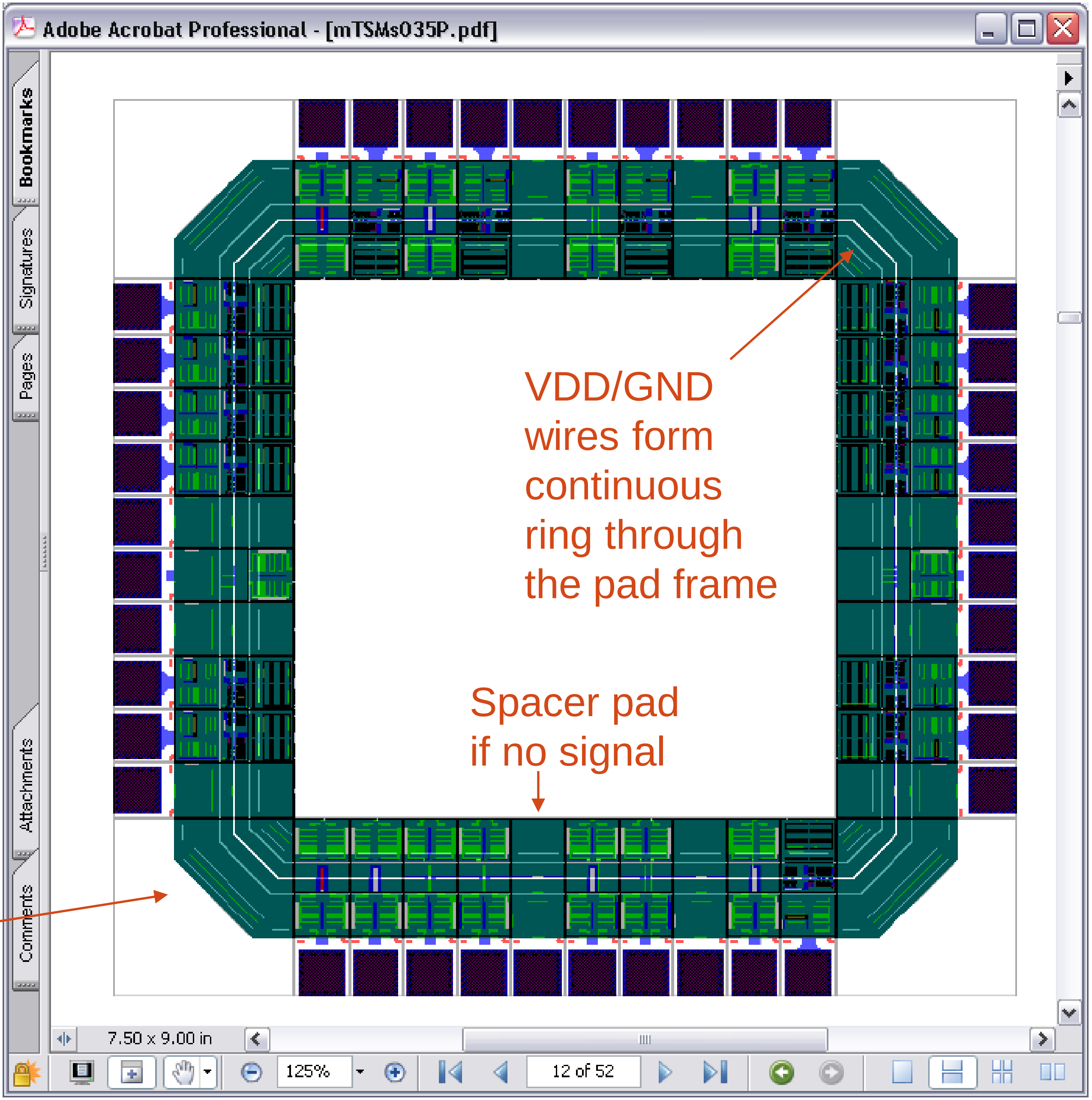
MOSIS
TSMC 0.35um
Hi-ESD
Pad Frame

(l) lambda=0.30um



MOSIS
TSMC 0.35um
Hi-ESD
Pad Frame

Physical layout

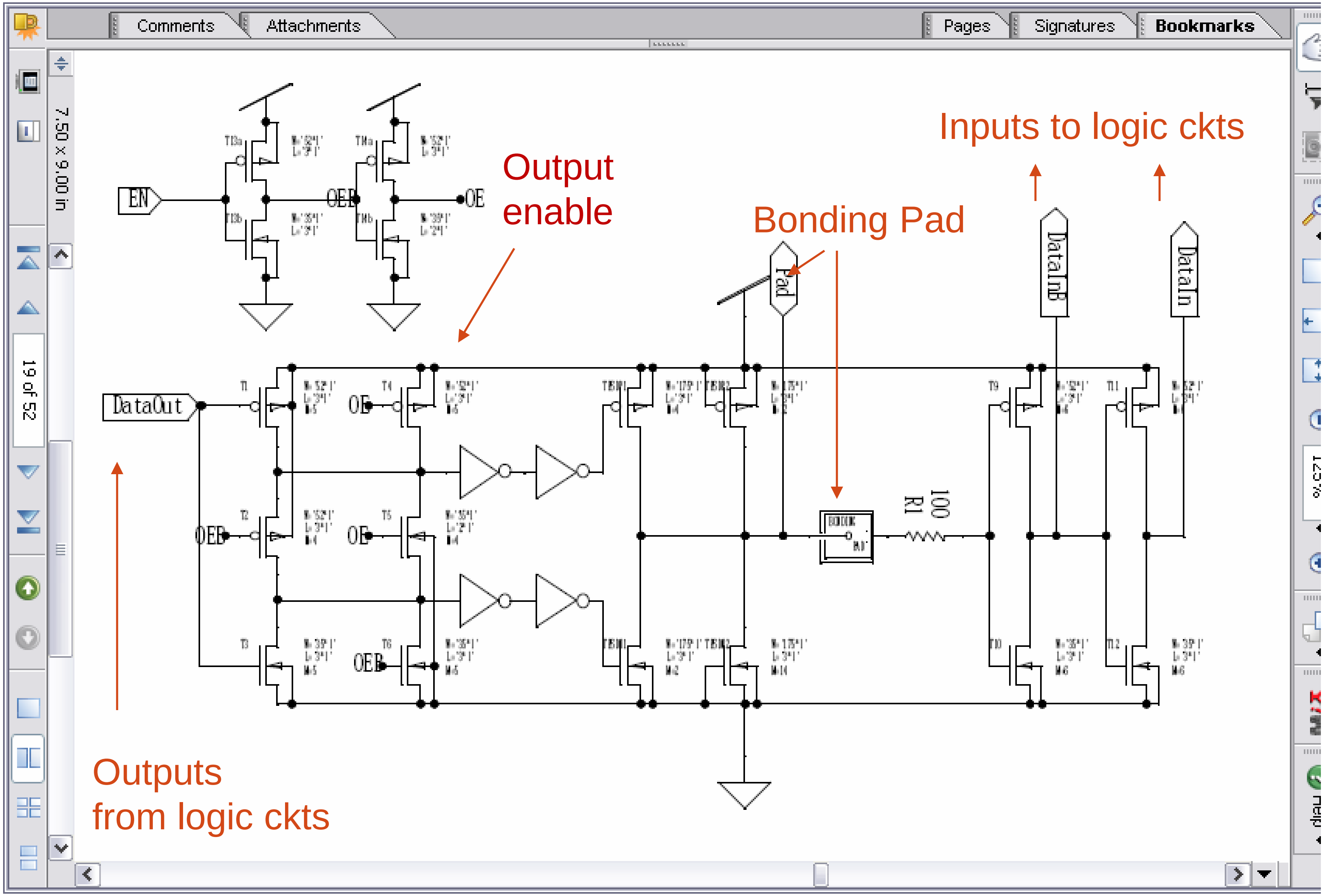


VDD/GND wires form continuous ring through the pad frame

Spacer pad if no signal

Corner pad (passes VDD/GND)

MOSIS I/O Pad Schematic



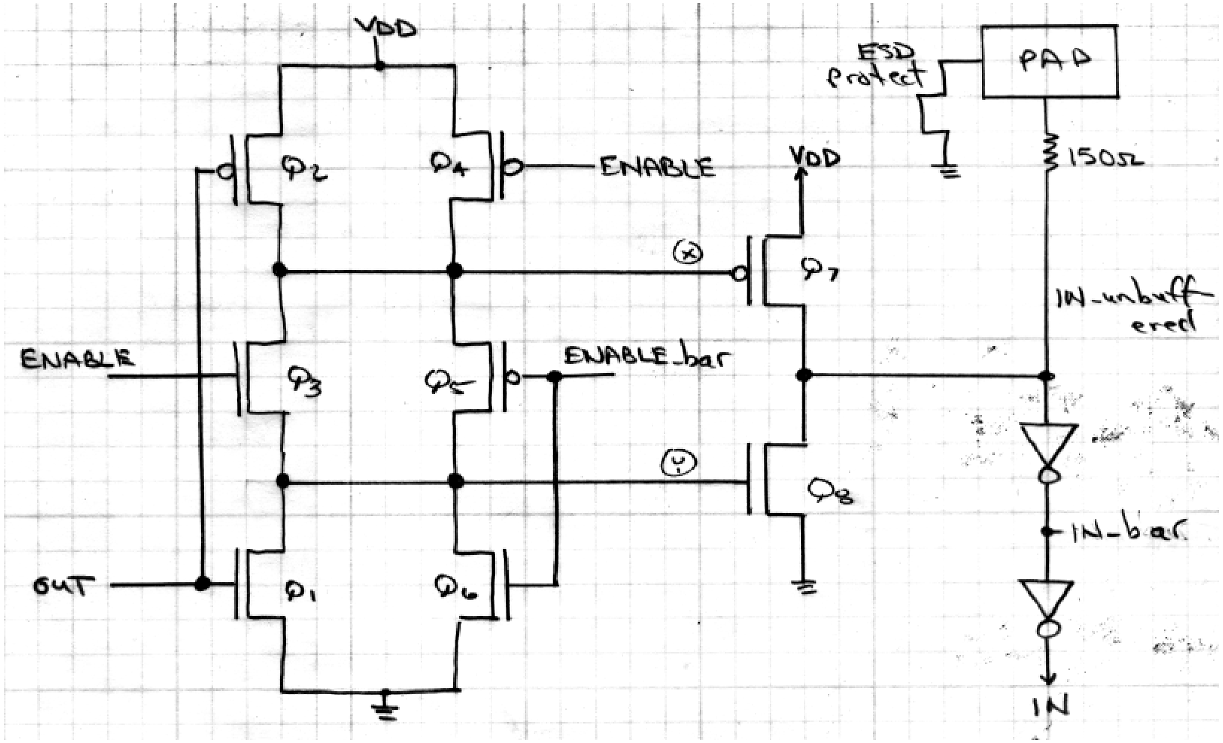
Output enable

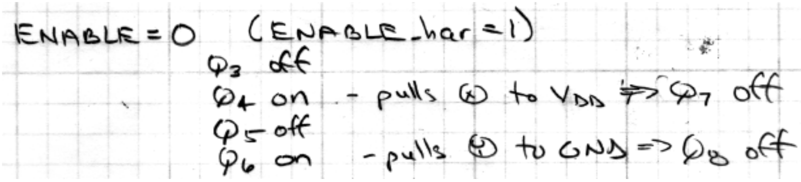
Bonding Pad

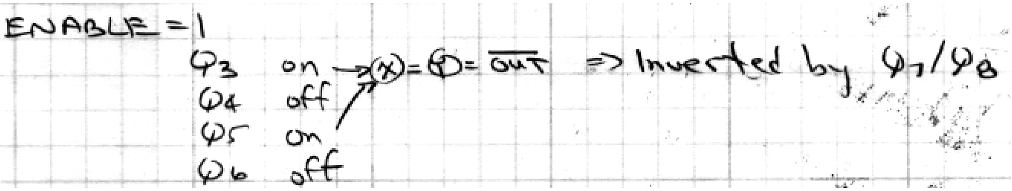
Inputs to logic ckts

Outputs from logic ckts

Simplified pad circuit




ENABLE = 0 (ENABLE_bar = 1)
Q3 off
Q4 on - pulls (x) to VDD ⇒ Q7 off
Q5 off
Q6 on - pulls (y) to GND ⇒ Q8 off


ENABLE = 1
Q3 on → (x) = (y) = $\overline{\text{OUT}}$ ⇒ Inverted by Q7/Q8
Q4 off
Q5 on
Q6 off

MOSIS 1.6 um bidirectional pad

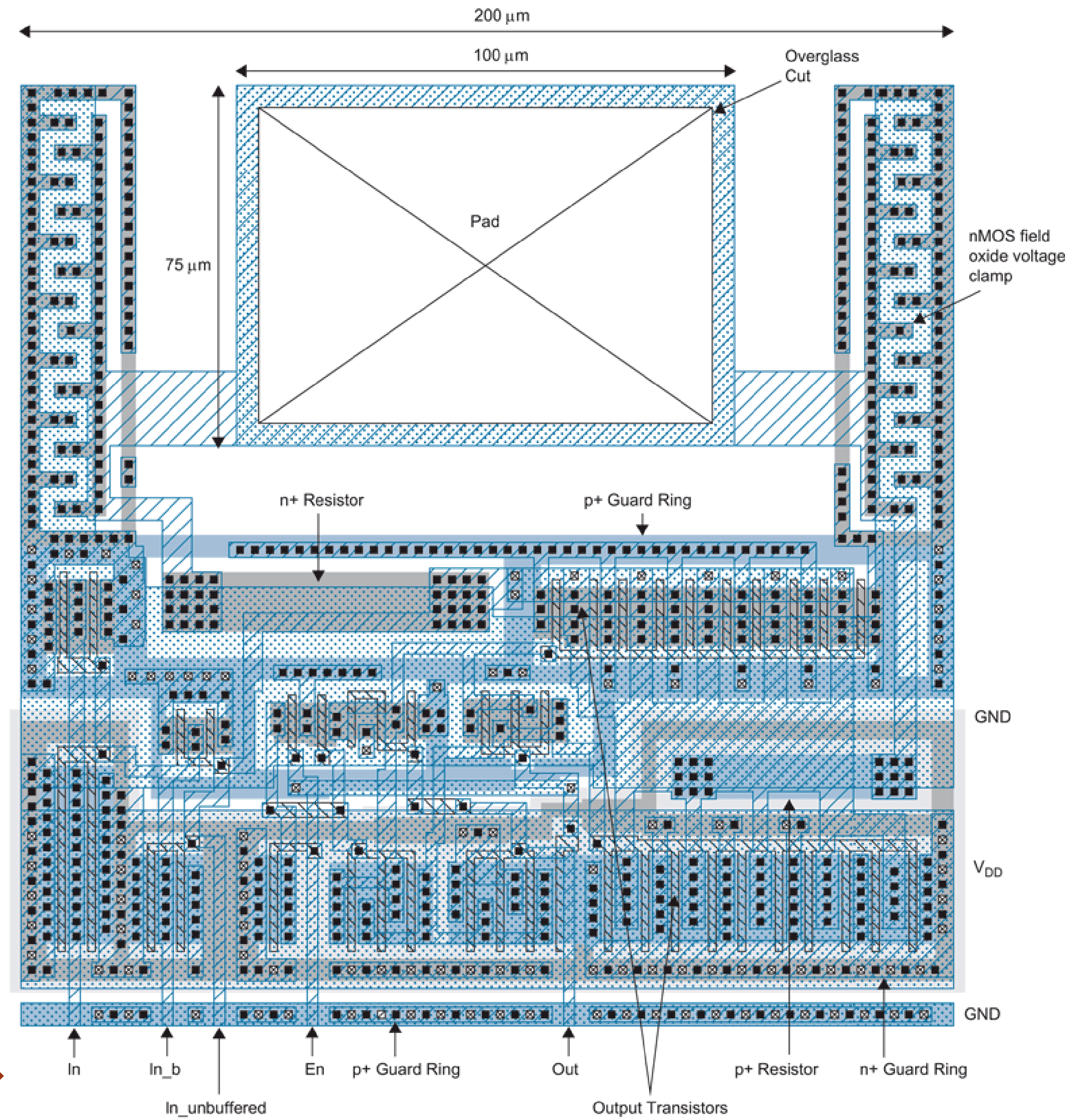


FIG 12.23 MOSIS 1.6 µm bidirectional pad. Color version on inside front cover.

To Core

Source: Weste, “CMOS VLSI Design”

ASIC frame + core in Virtuoso

Process:

1. Create "core" block
2. Create pad frame
3. Connect them

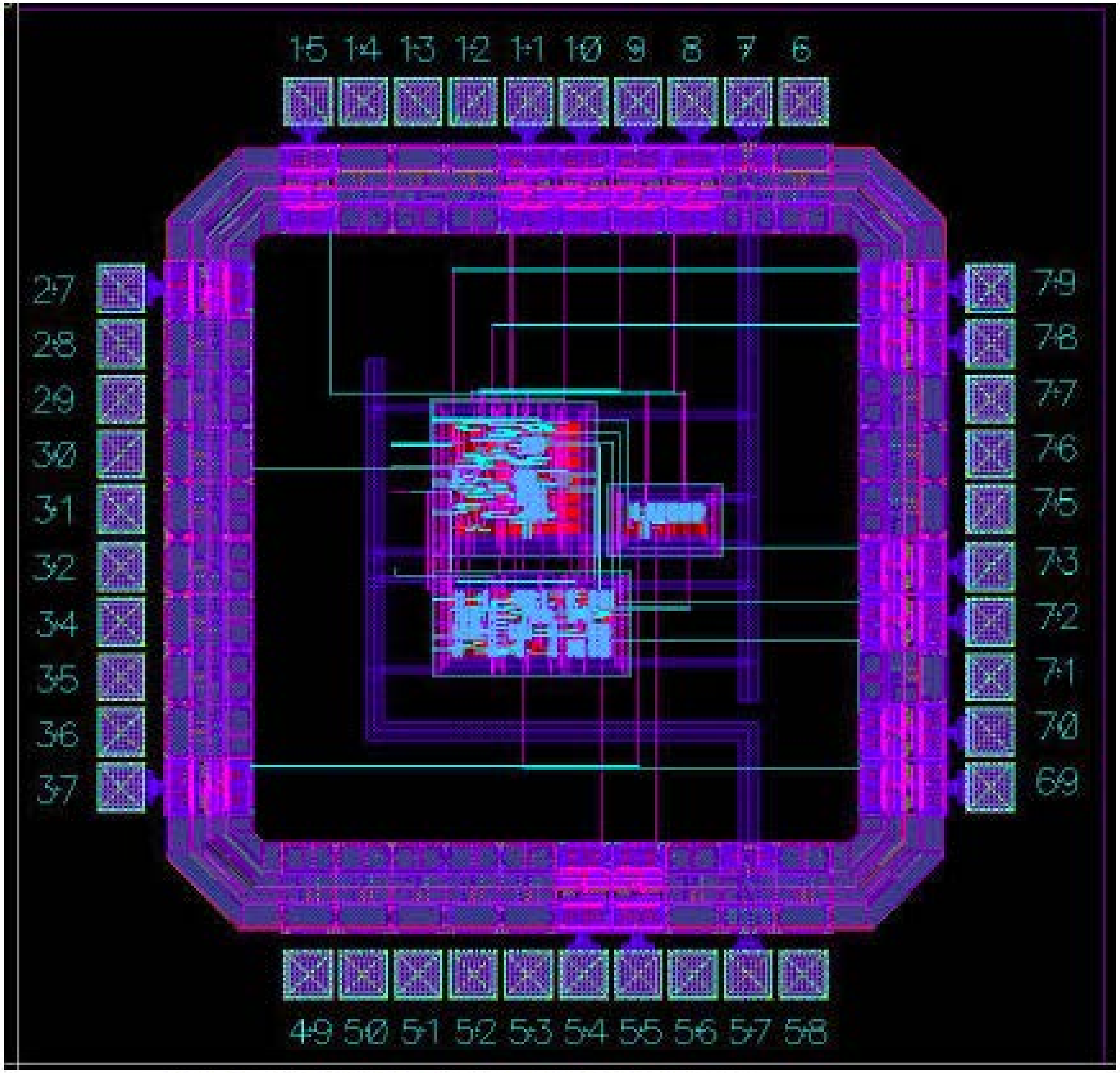
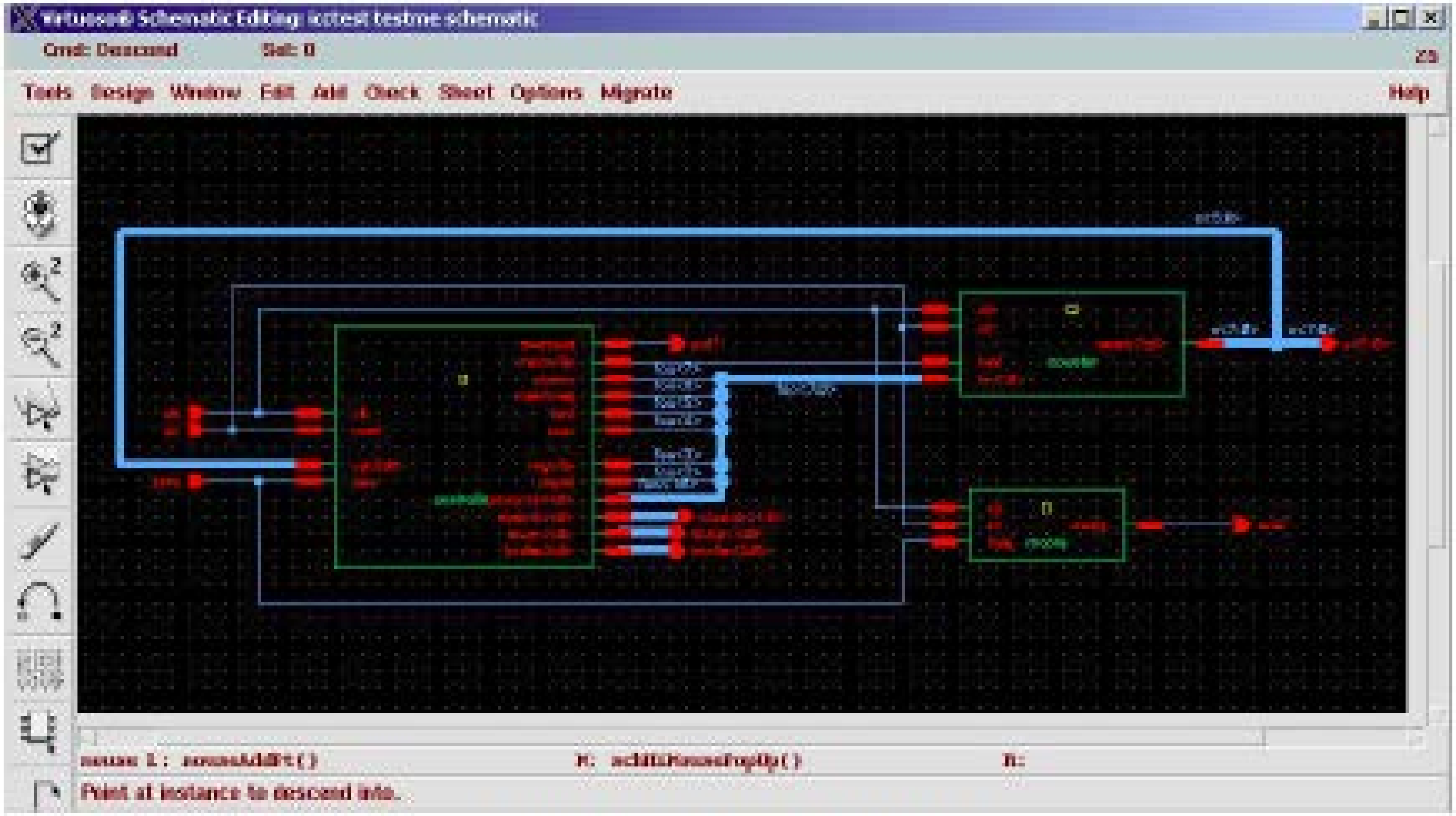


Figure 12.22: Frame and core after routing in *Virtuoso*

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”

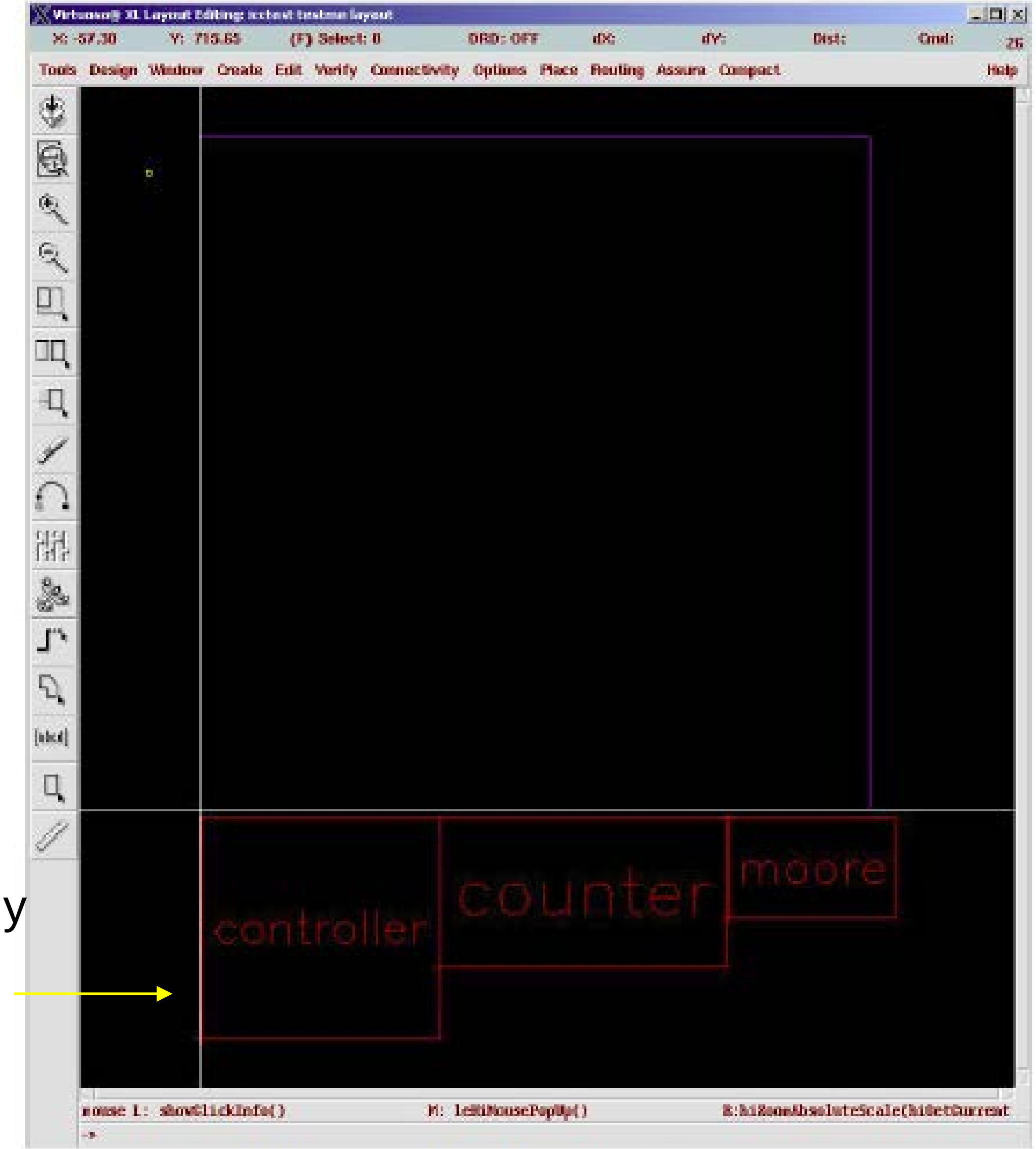


Point at instance to descend into.

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 12.1: Starting schematic showing the three connected modules

Before module and I/O placement



Virtuoso® XL Layout Editing: ccchest testme layout

X: -57.30 Y: 715.65 (F) Select: 0 DRD: OFF dX: dY: Dist: Cmd: 26

mouse L: showClickInfo() M: leHiMousePopUp() R:hiZoomAbsoluteScale(hiGetCurrent

Blocks initially outside boundary

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 12.3: Initial layout before module and I/O placement

After placing modules and pins

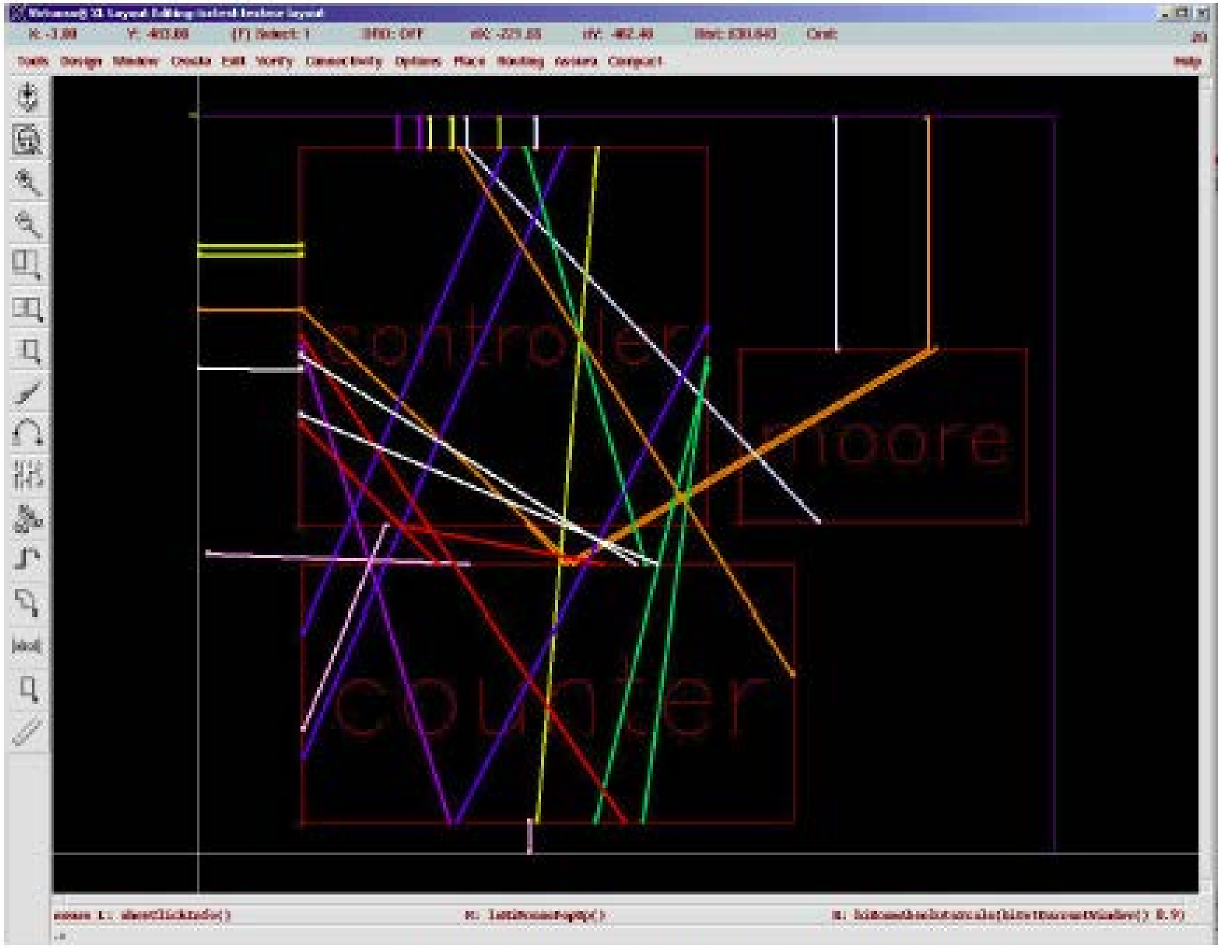
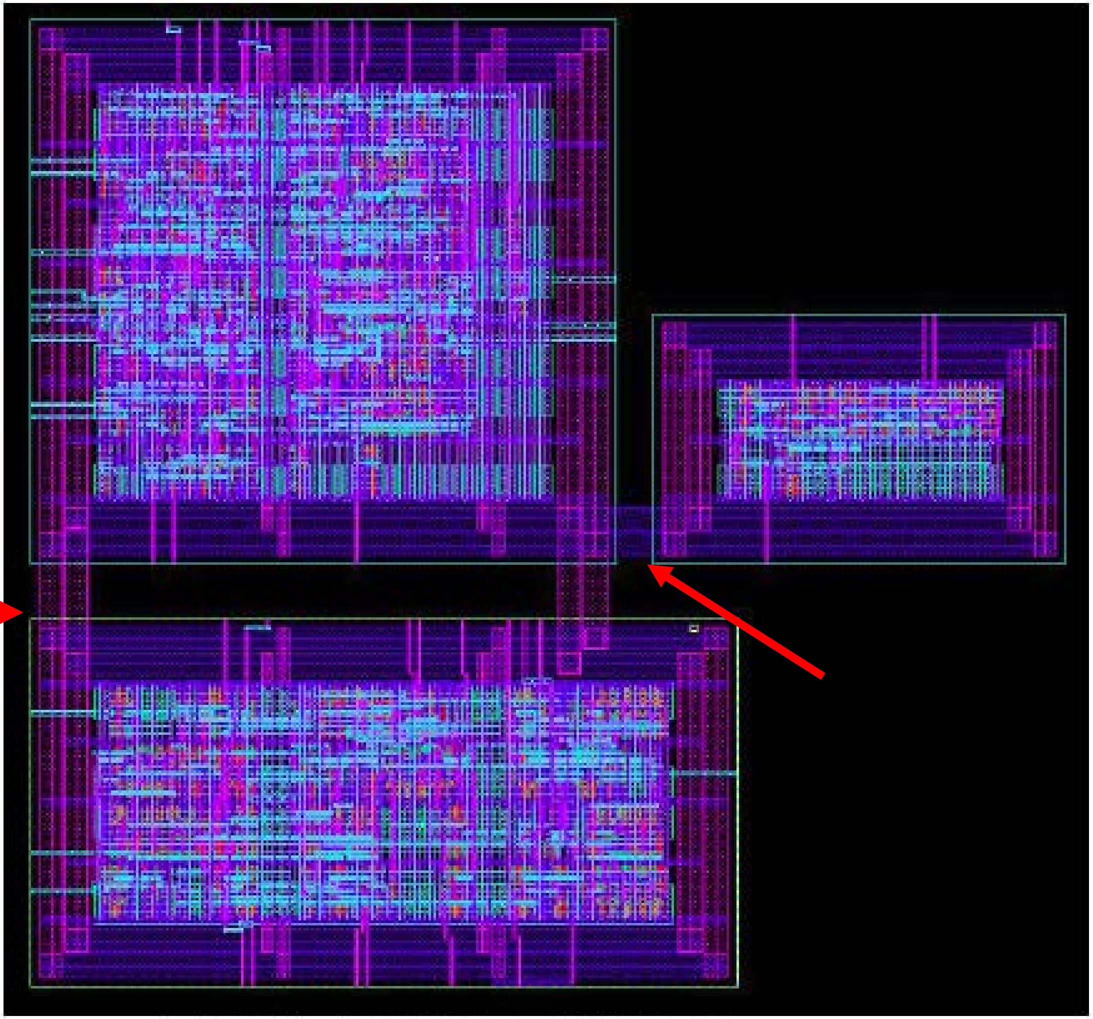

Tools Design Window Create Edit Verify Connectivity Options Place Routing Assura Compact
Help
controller
moore
counter

Figure 12.4: A placement of modules and IO pins with unrouted nets turned on

Power routing between blocks

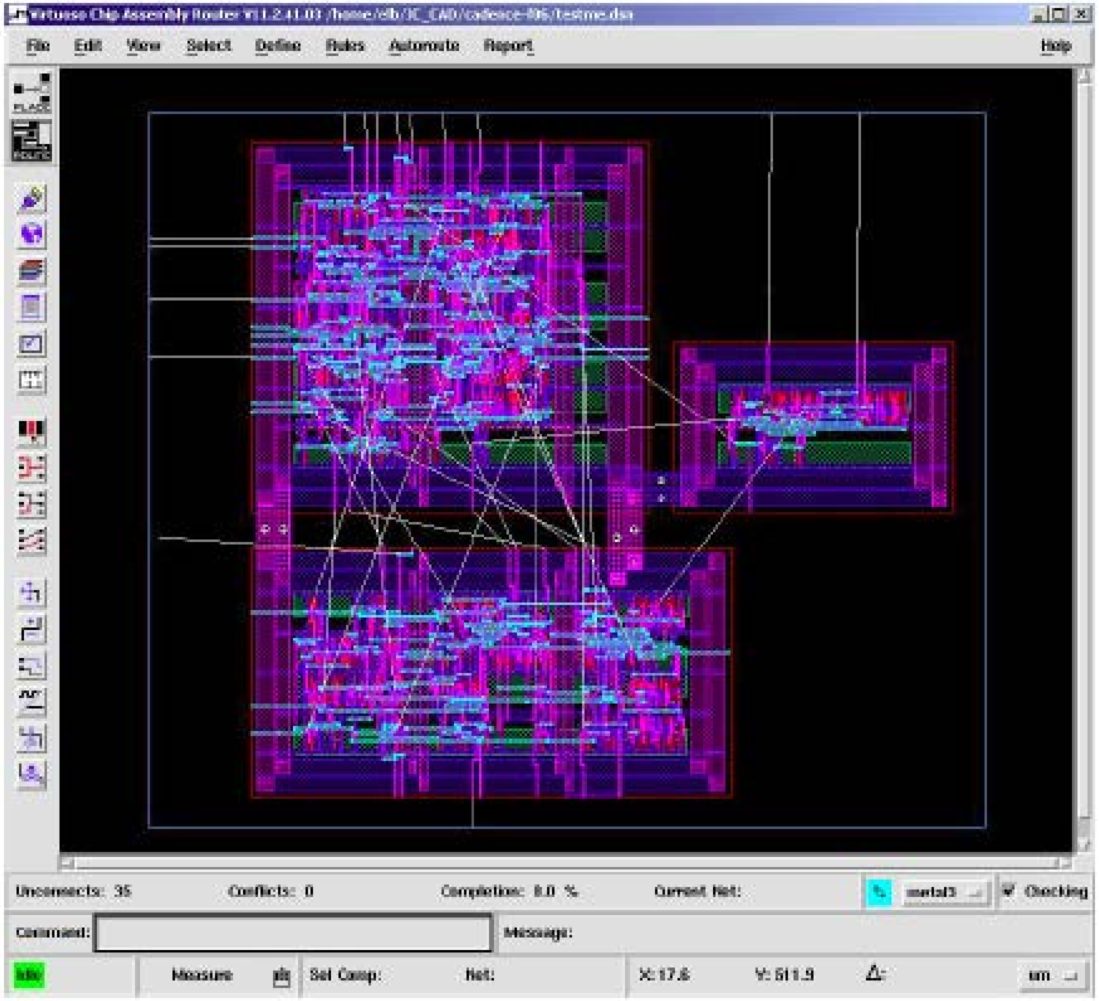


Connect power rings

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 12.5: Layout showing placement and power routing before routing

Nets shown as “overflows”



Unconnects: 35 Conflicts: 0 Completion: 0.0 % Current Net:

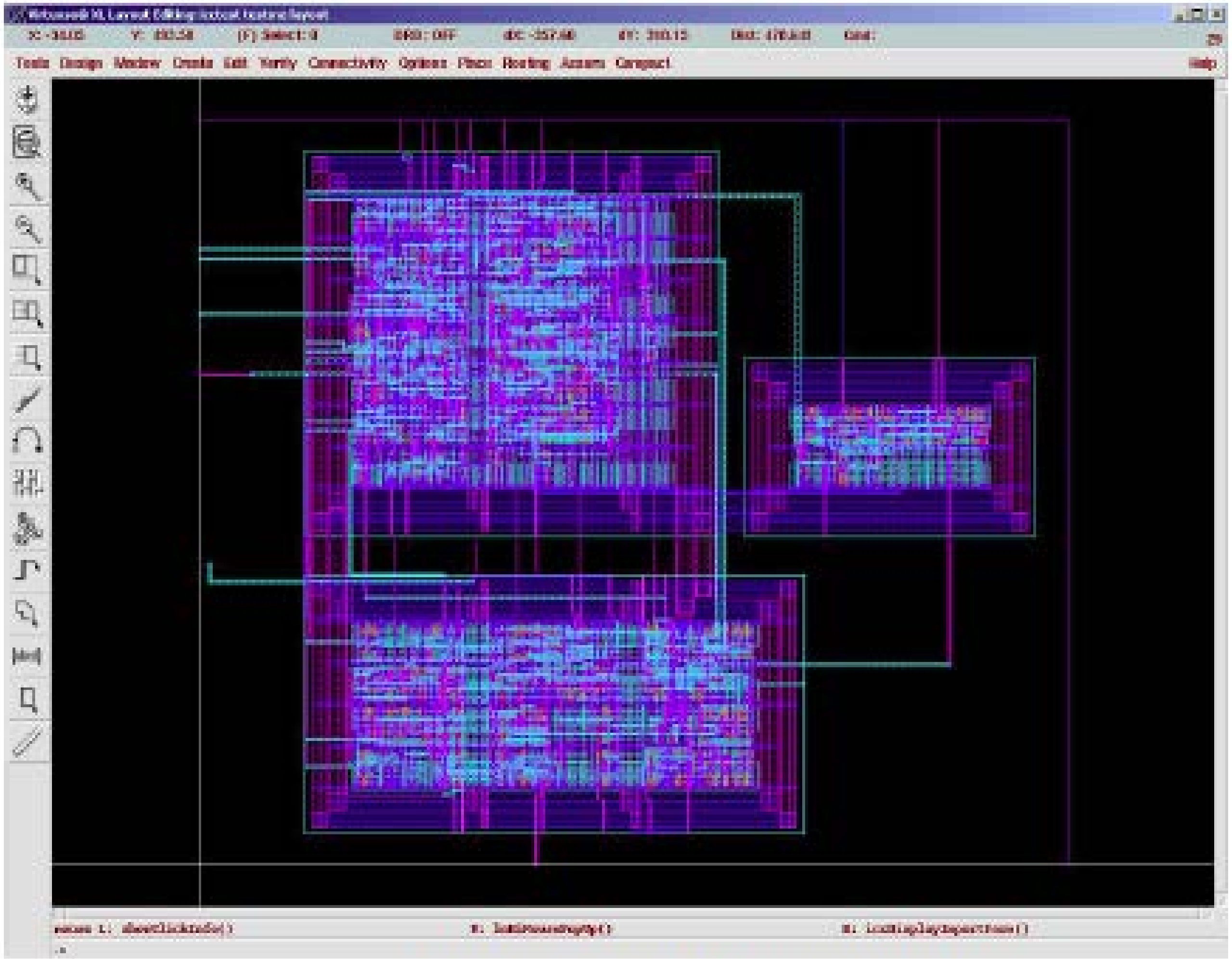
Command: Message:

Sel Comp: Net: X: 17.6 Y: 511.9 Δ:

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 12.7: Initial ccar window

Routed circuit block



(Copyright © 2006, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 12.10: Final routed circuit (shown in *Virtuoso* window)

Block symbol (to connect to I/O pads)

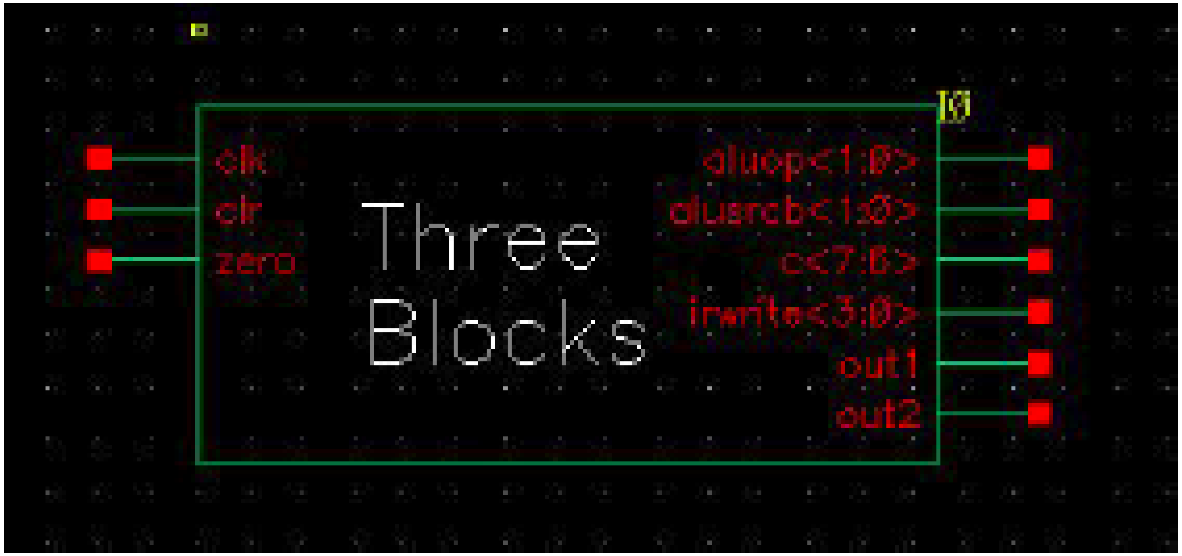


Figure 12.11: Symbol for the Three Blocks example core

Pad frame with signal wires

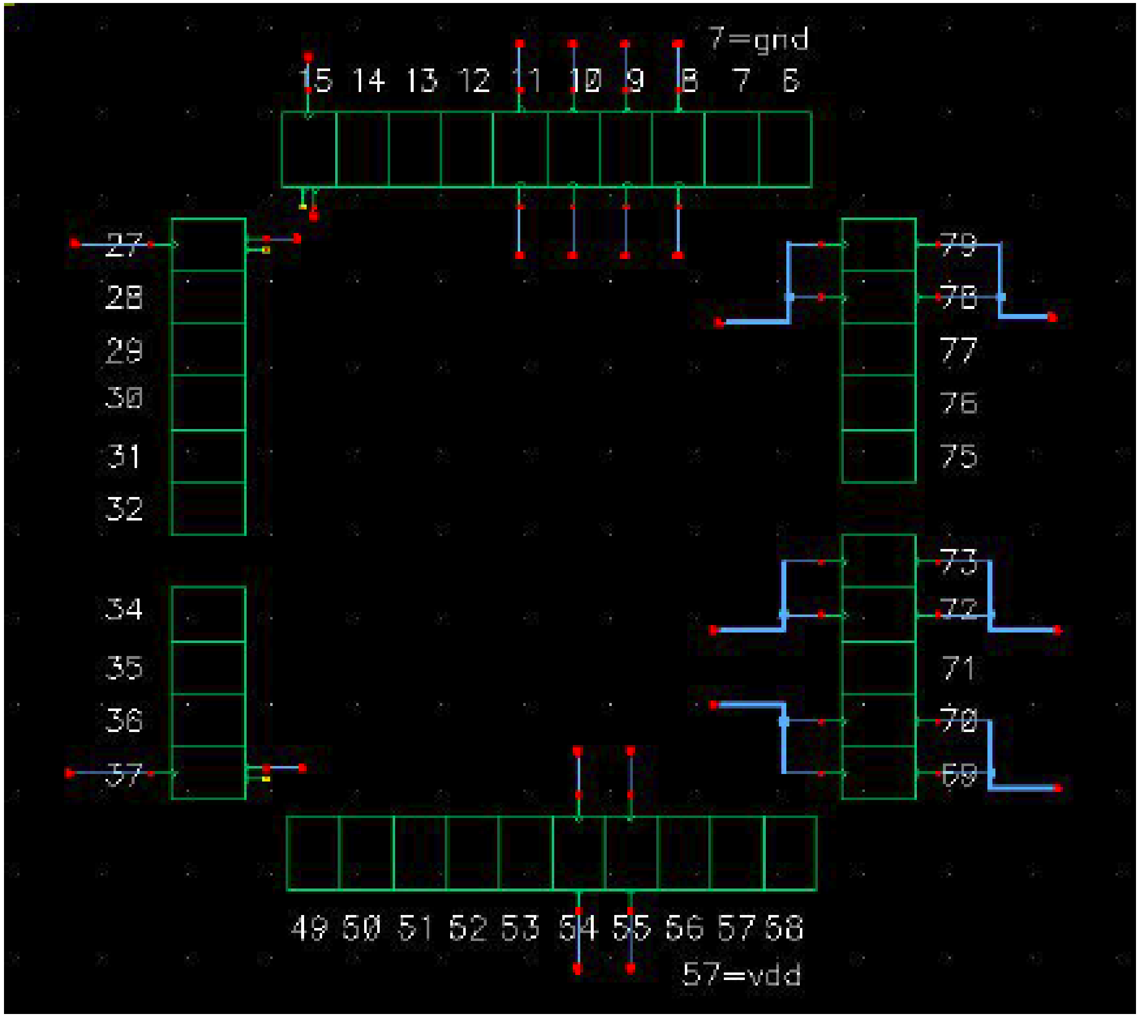


Figure 12.12: Pad frame with signal wires

Zoomed view of pad frame

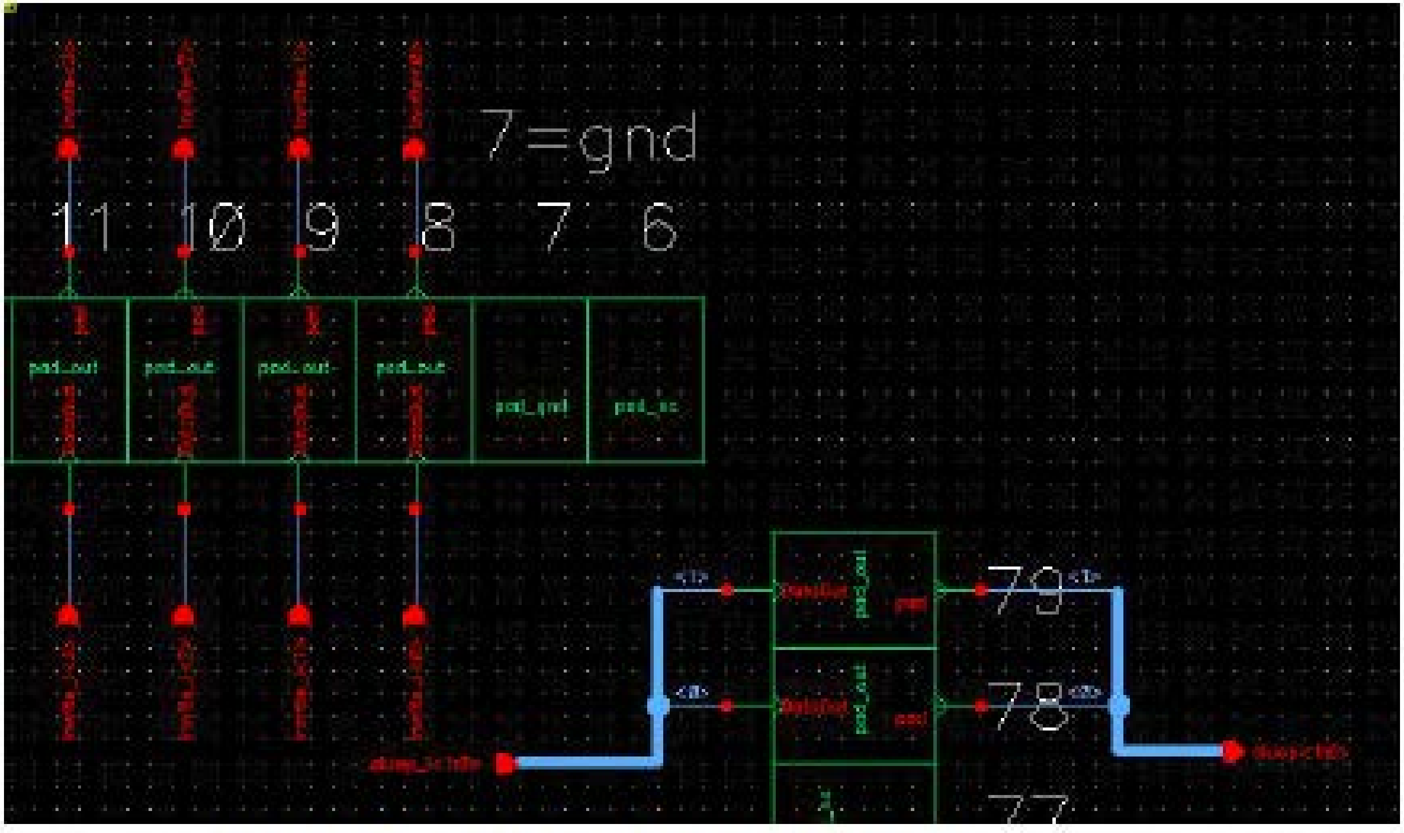


Figure 12.13: Pad frame with signal wires (zoomed view)

Schematic: block + pad frame

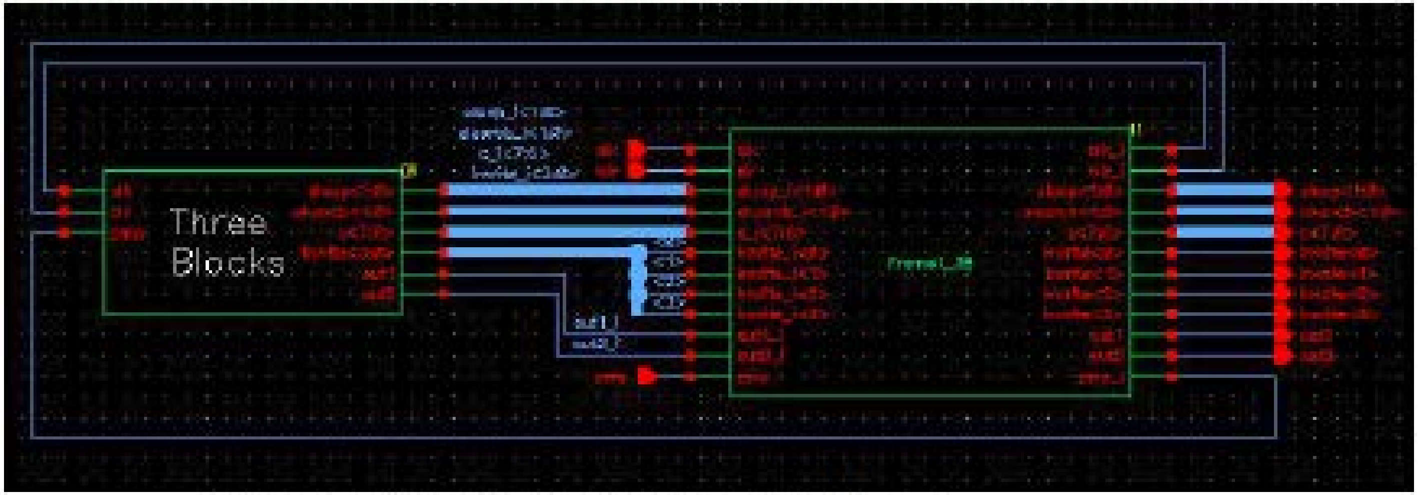


Figure 12.14: Frame and core components connected together

Placement of frame and core

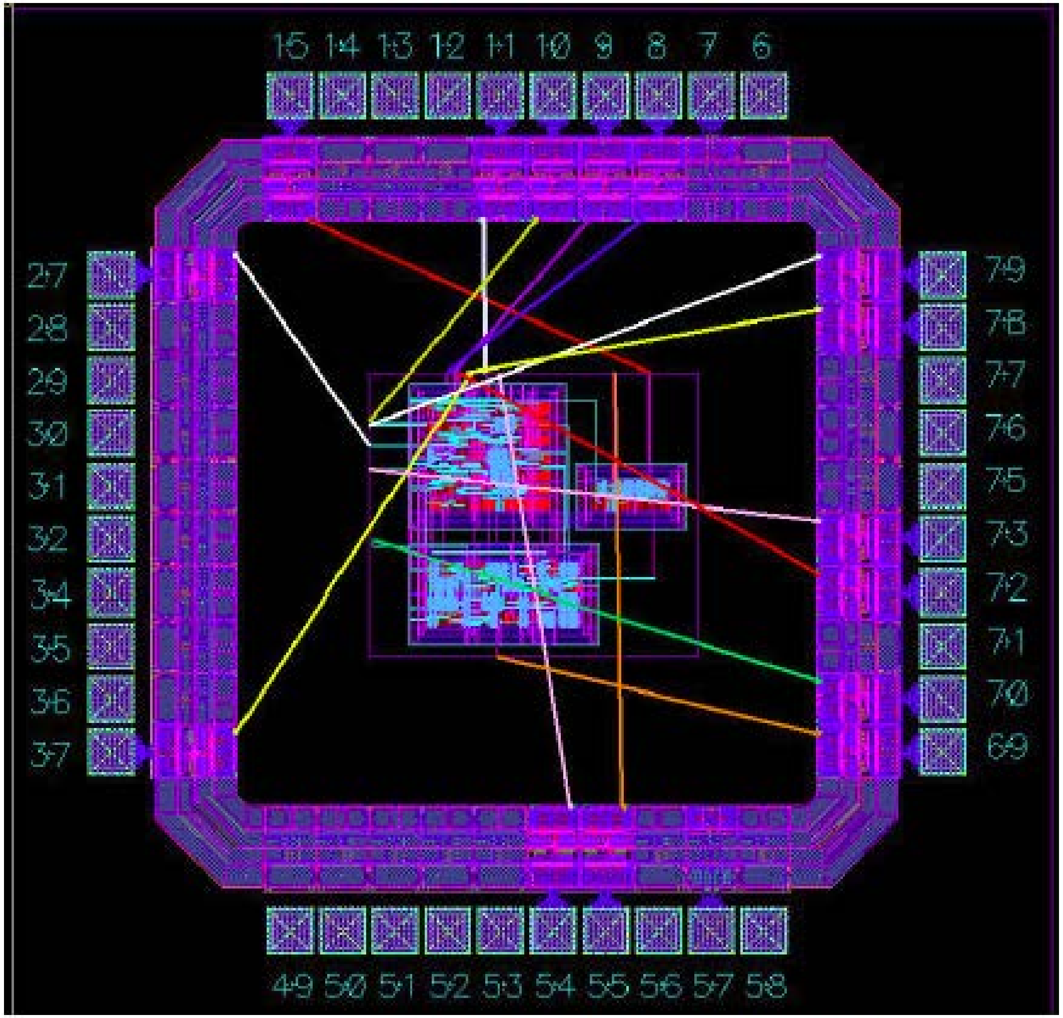


Figure 12.19: Frame and core placed in *Virtuoso-XL*

Power/ground routed manually

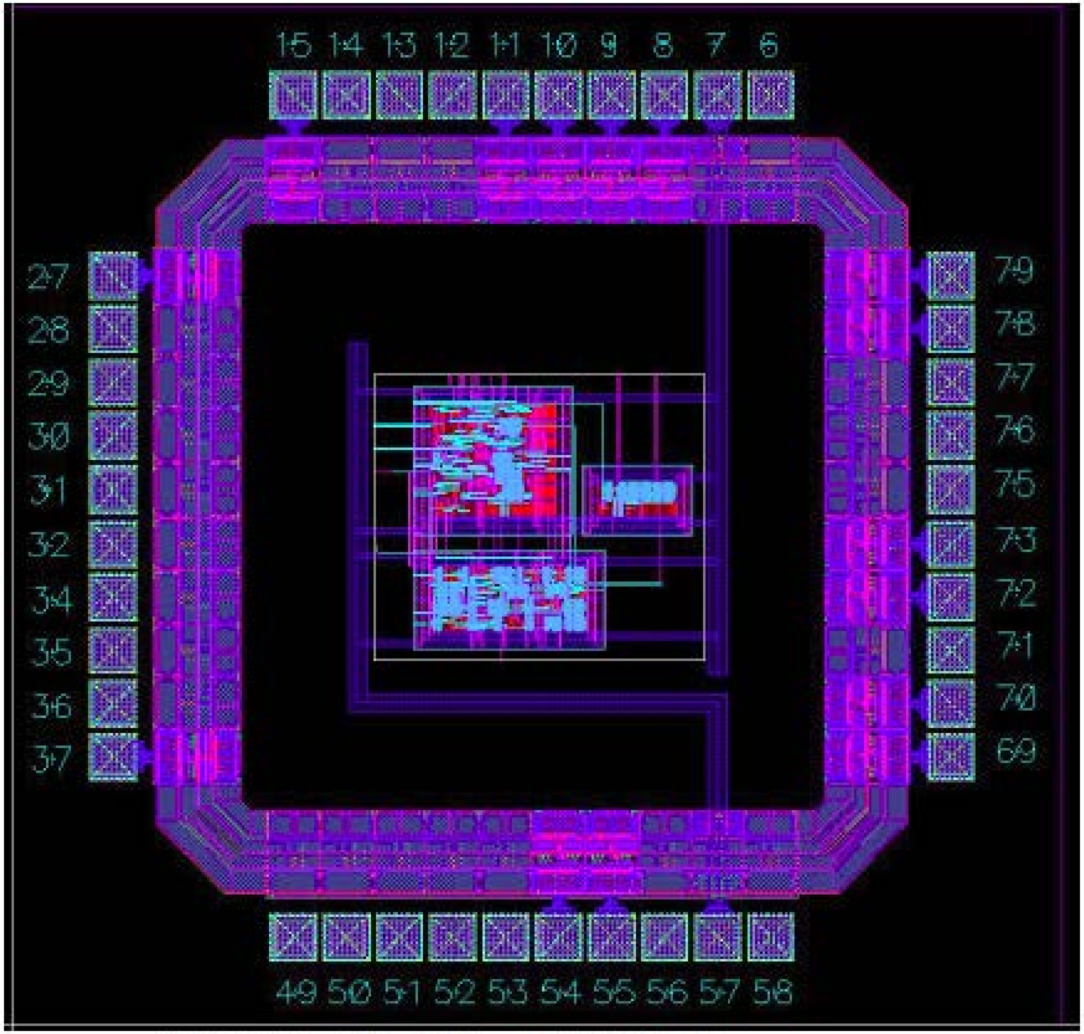
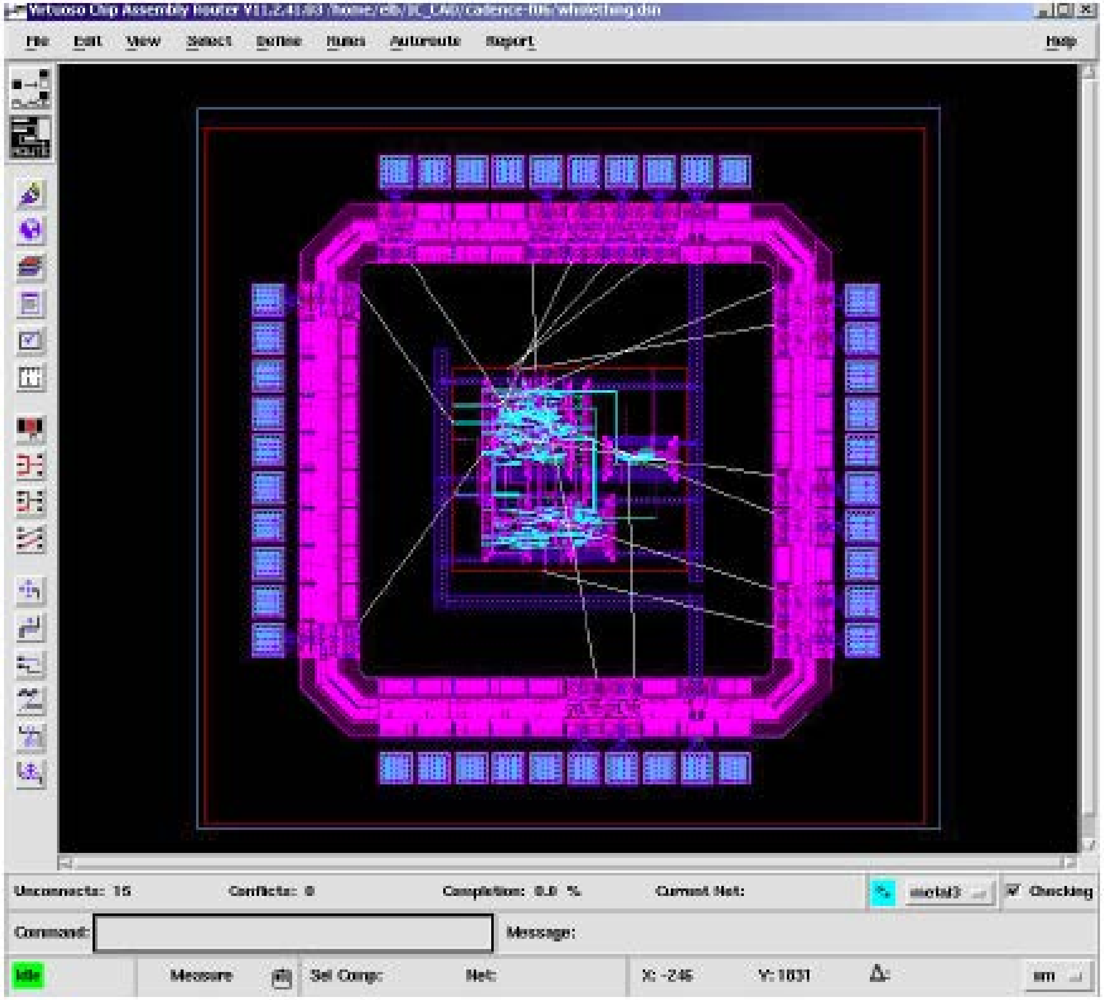


Figure 12.20: Frame and core placed in *Virtuoso-XL* with **vdd** and **gnd** routing completed

Before signal routing


Virtuoso Chip Assembly Router V11.2.41.03 /home/elb/IC_CAD/cadence-f06/whatething.dsn
File
Edit
View
Select
Define
Rules
Autoroute
Report
Help
metal3
Idle
Measure
m

Unconnects: 15 Conflicts: 0 Completion: 0.0 % Current Net:

☑ Checking

Command:

Message:

Sel Comp: Net: X: -246 Y: 1031 Δ:

Figure 12.21: Frame and core before routing in *ccar*

After routing – final layout

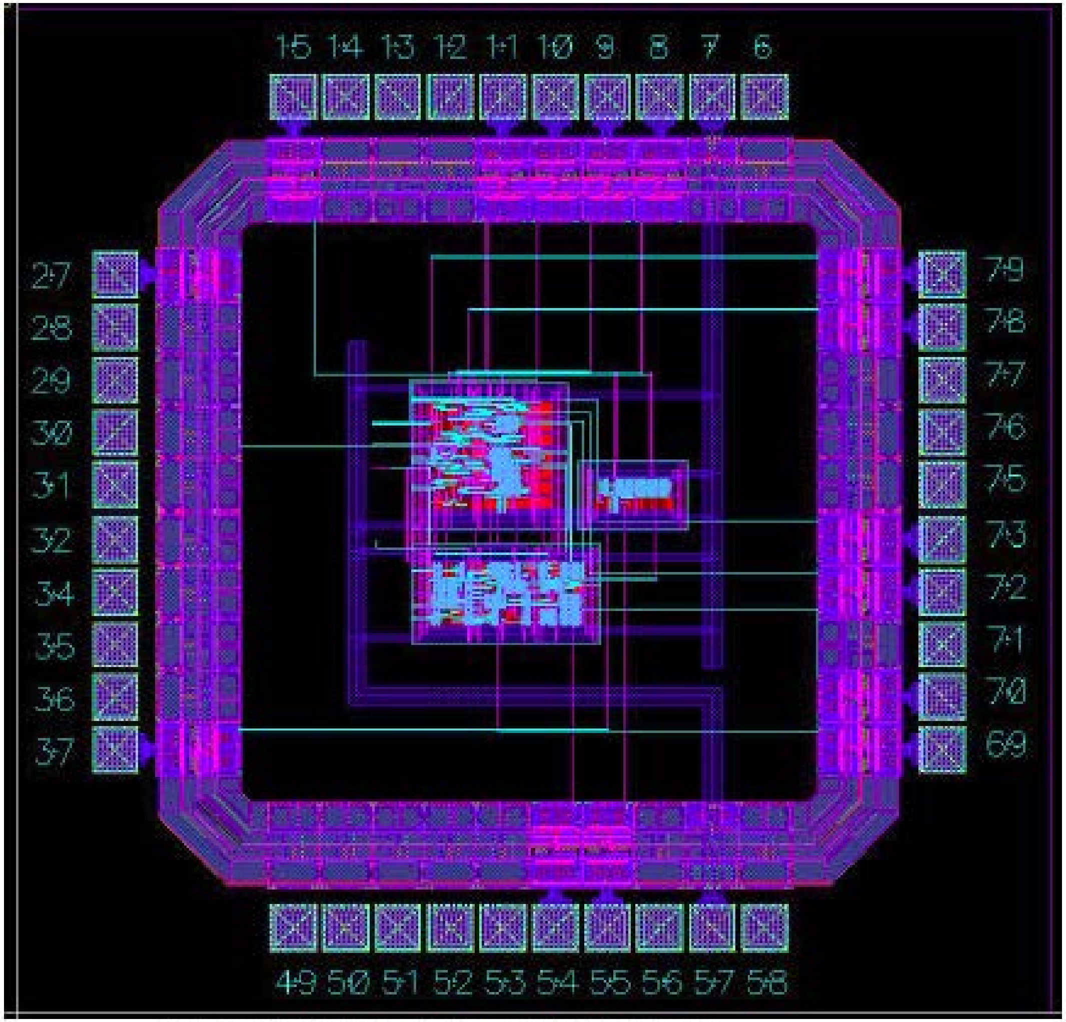


Figure 12.22: Frame and core after routing in *Virtuoso*