

Design Digitaler Schaltkreise

Place and route 1

Asic and Detector Lab - IPE

Prof. Ivan Peric ivan.peric@kit.edu

Richard Leys richard.leys@kit.edu

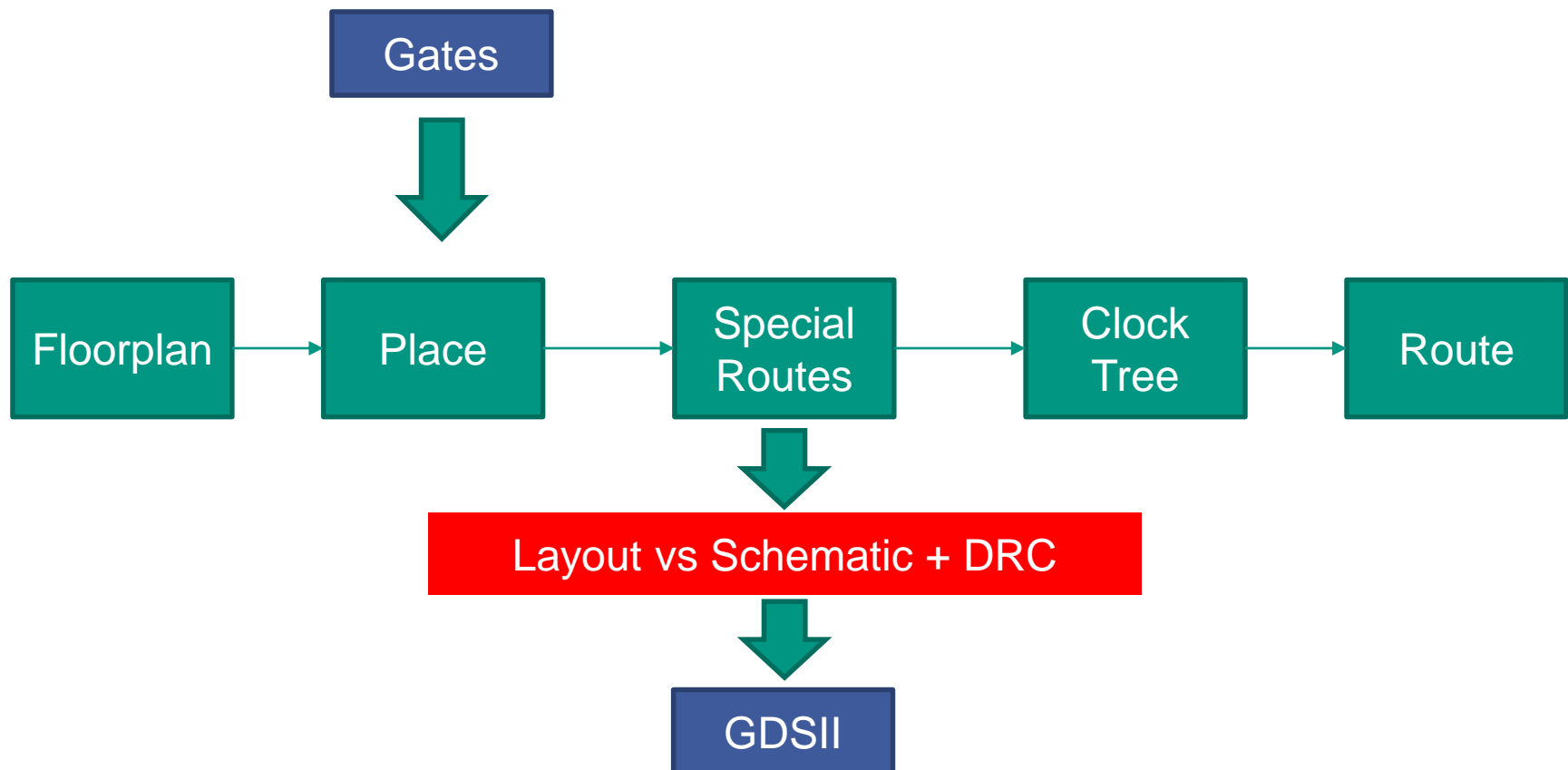


Lecture Goal

- Get an overview of place and route steps
- Understand floorplaning steps to prepare top level
- After that: Ready to go for final implementation
- Talk about timing later

Place and Route Goal

- After Synthesis, we have gates and interconnection information
- Physically place the circuit to create an ASIC



P&R Strategy

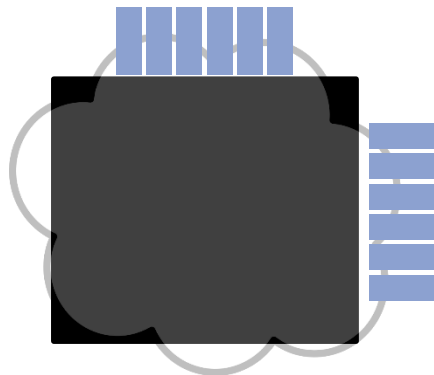
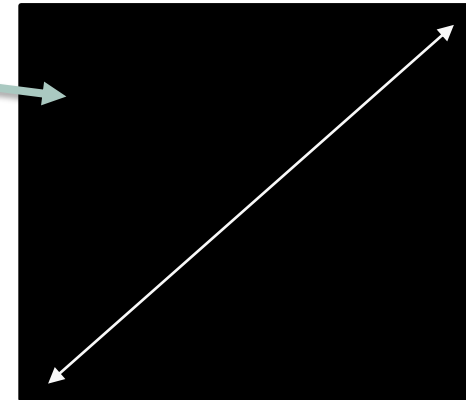
- Try to create the best conditions for an easy implementation
- Ex: Not too big (long wires), Not too small (less optimisation freedom)
- But take into account the design constraints, like embedded special blocks (PLL, Memories, Test controllers), cost etc
- Let the software perform as much automated work as possible
- Timing is checked the whole time
 - Each step have small adjustment steps to improve timing

Floorplaning Overview

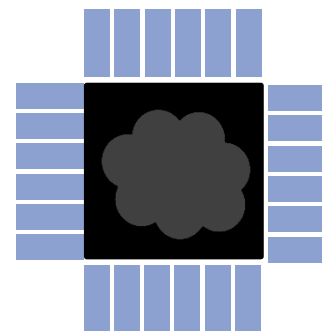
- Prepare the design for physical implementation: set the foundations
- The best the floorplaning, the less effort required by automated tasks
- Usually heavily scripted because a lot of small components must be added to the design for physical rules
 - If any re-synthesis is necessary, just re-run the script
- During floorplaning:
 - Prepare the die dimensions
 - Prepare the I/O
 - Pre-Place hard macros
 - Route special wires
 - Create Partitions (if necessary)
 - Add spare cells, well tap etc...
 - Ready to continue!

Die Size Configuration

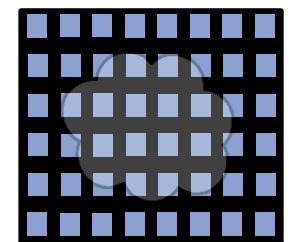
- Core Die Area
- Size constraint:
 - Cost
 - Technical limitation (Multi layer Mask)
 - Minimal size sometimes
- Size Definition:
 - I/O bound: Too many pins
 - Core bound: Too much logic



Core Area Bound

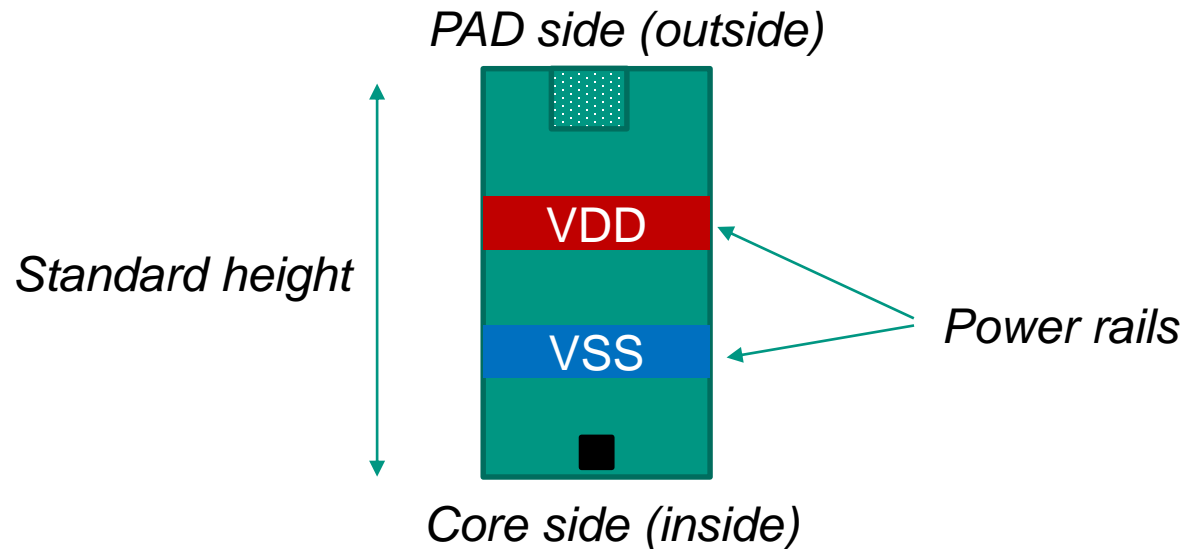


I/O Bound



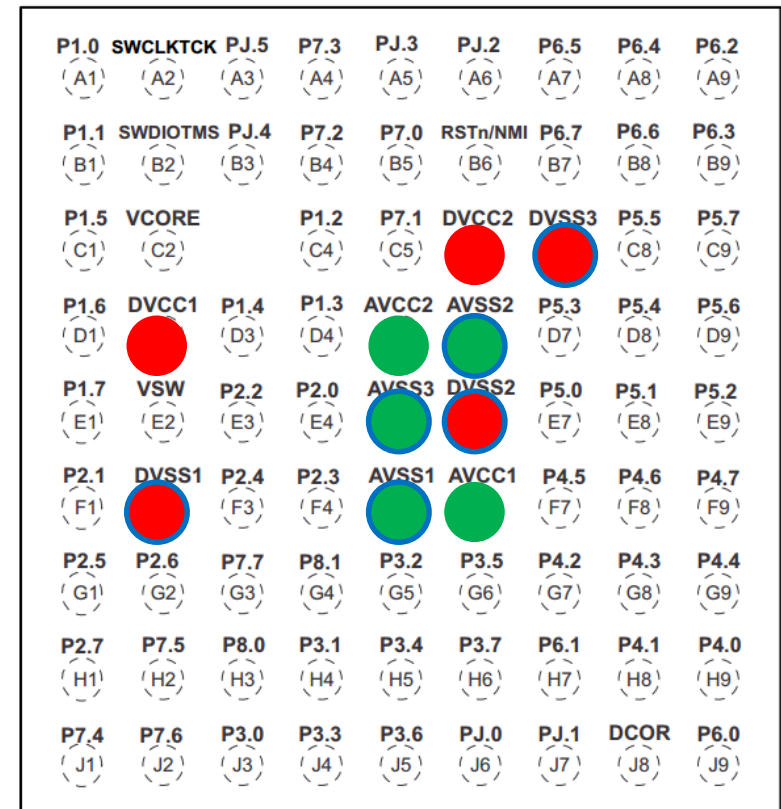
I/O Placement and Pads

- I/O provide interface to external world
- In smaller technology node, there is an IO voltage:
 - UMC65: 1.8 V for IO typically, ~1V for core logic
- I/O cells are big and provide voltage translation, as well as ESD protection
- Like Standard cells, they are designed to be abutted to each other



Power Domains and IO

- Power Domains are used to separate various supplies:
 - Typical: Analog and Digital Domains
 - Possible: Multiple digital power domains
- Explanation:
 - Digital and Analog are separated to avoid power supply cross-talk
 - A digital domain can have a lower VDD to save power, and another one a higher VDD for better performances
- Advanced options:
 - Digital Power domains can interface using level shifters placed by the tool
 - Digital Power domain can be “shutdown”; like a sleep mode



TI MSP432 Microcontroller PADS

IO Power rows design

- IO Cells have specific rules for power rail connections
- The Technology documentation provide information
- Design by abutment, power rails must respect some rules:
- All power domains must be separated

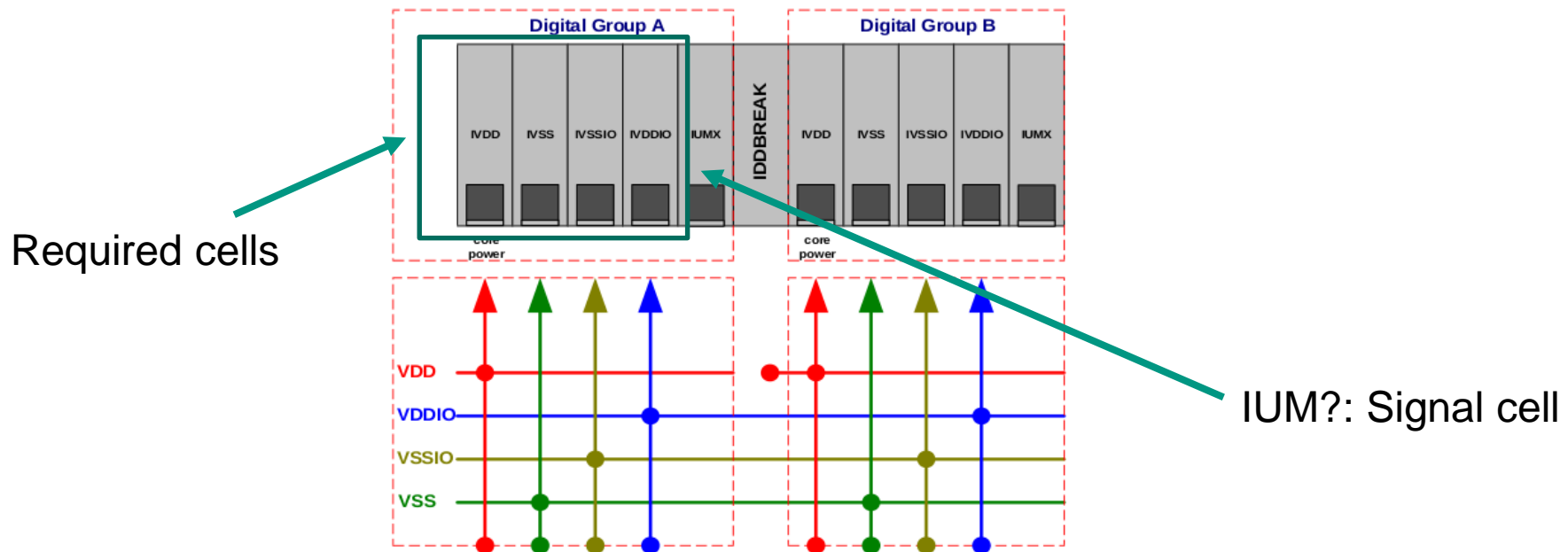
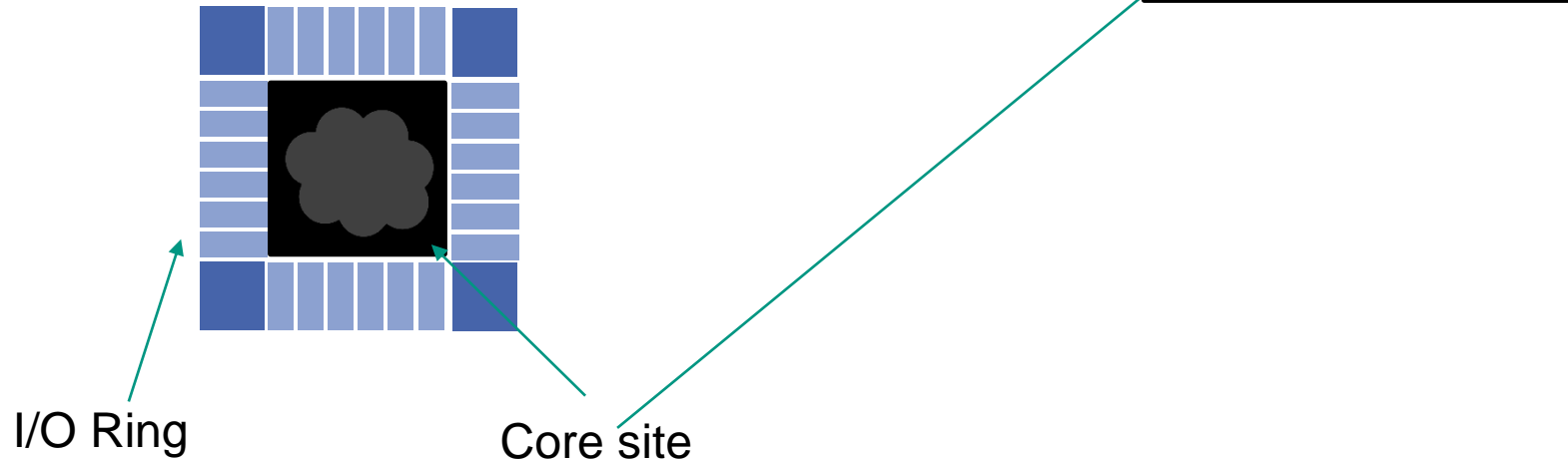


Figure 4-3-A. An Example of a Digital-to-Digital Interface

Example from UMC65: Multiple Digital Domains

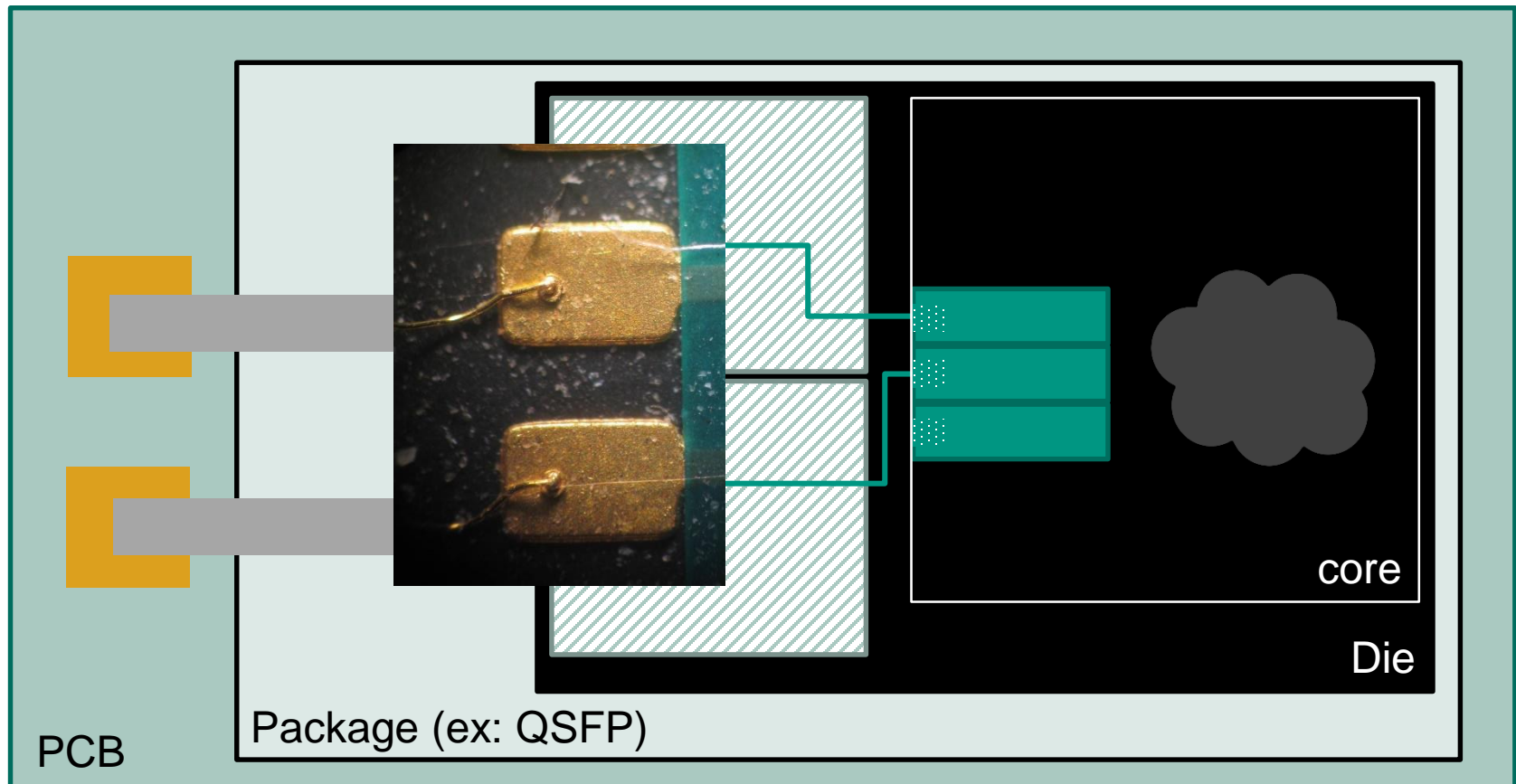
IO Placement types

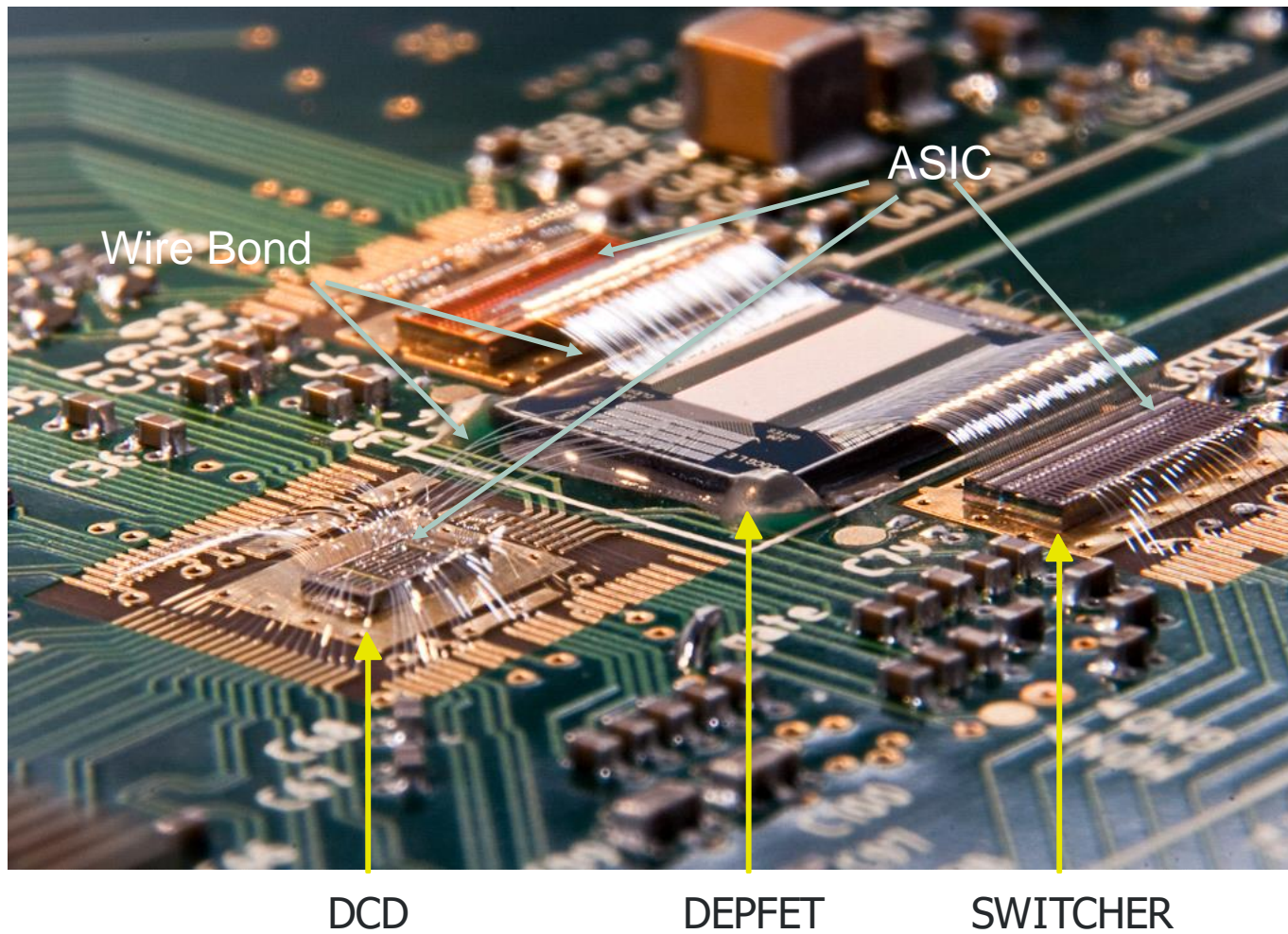
- IO cells can be placed:
 - outside the core area: Ring IO
 - Inside the core area: Area I/O
 - If design is big, it gives more freedom
- Connection PADS can be placed:
 - around the die: PADS for wire bonding
 - on the area for flip-chip connection



PAD type: Wire Bonding Pads

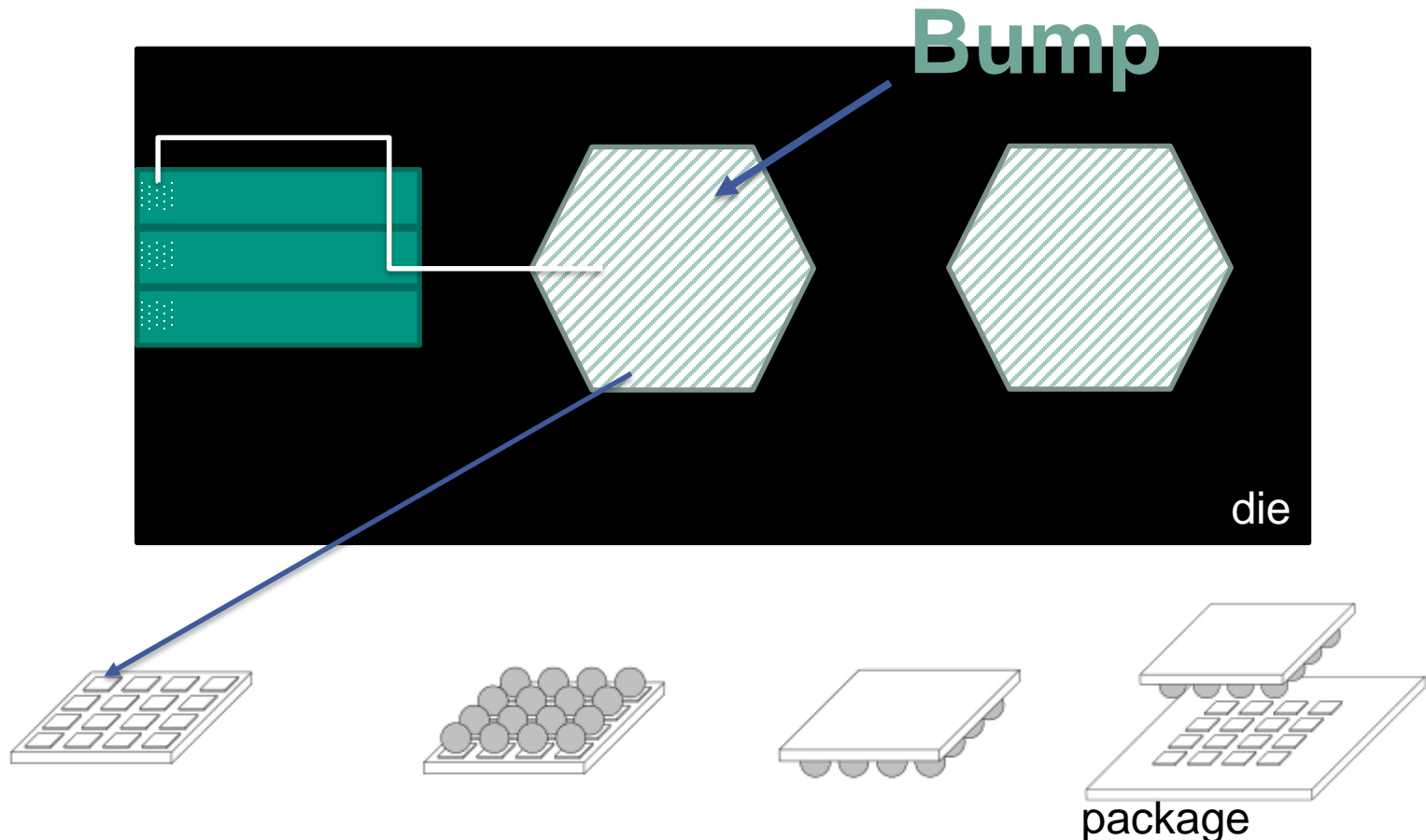
- Connection PADS can be around the core, with AREA IO or not





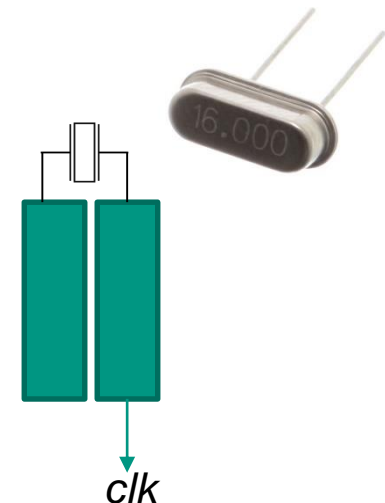
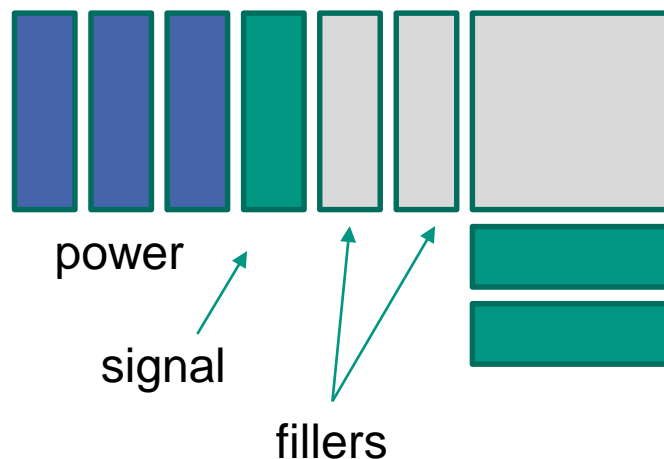
PAD type: AREA IO for Flip-Chip

- Area IO and Bumps are used for Flip-Chip mounting
- Bump to IO Cell is routed over a Redistribution Thich Metal layer



IO Cell Types

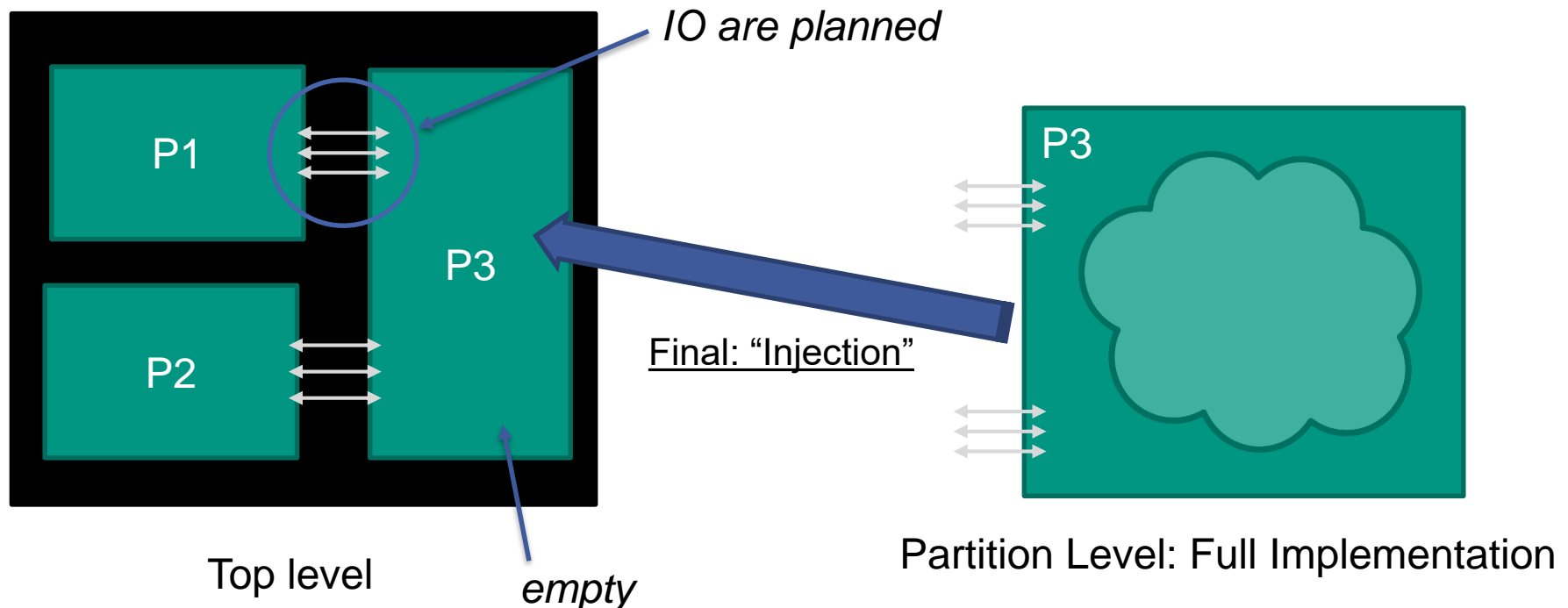
- IO Cells can be of various types, depending on the technology library
- Standard types:
 - Power Connections: VDD,VSS,VDDIO,VSSIO
 - Fillers: FILLER1, FILLER10, CORNER
 - Signal Cell for Digital/Analog Signals
- Special Cells:
 - Differential Signal (ex: LVDS), Oscillator cells etc...



Special Cells example: Oscillator

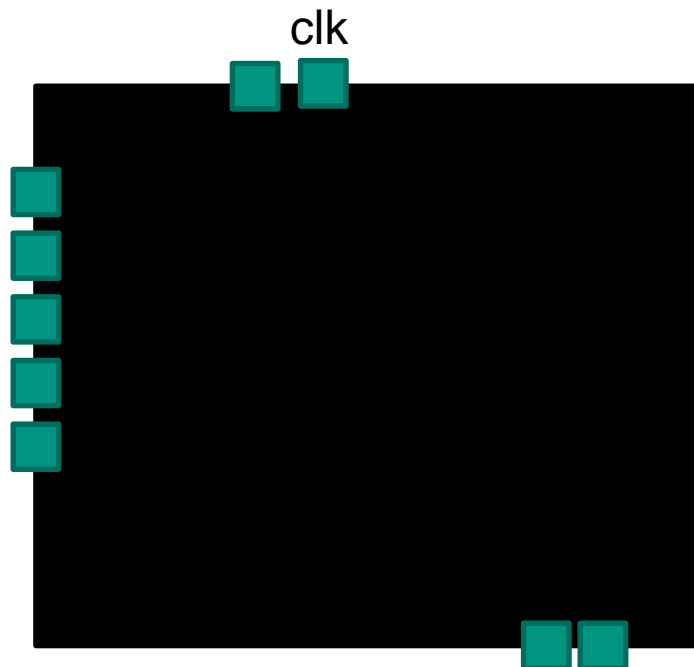
Partitioning: Definition and re-implementation

- Partitions are used to cut-off the design in smaller subset
- Each Subset is synthesised and implemented separately.
- The top level prepares IO Placement between partitions
- Final timing is done at top level with last adjustments



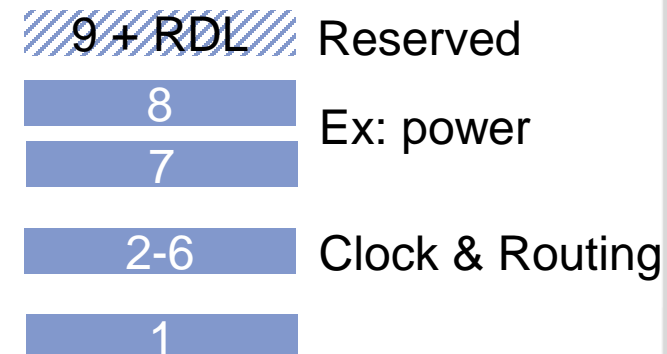
Partition Level implementation

- Like top level but constraints and IO are provided
- Proceed to normal implementation
- Leave topmost layers in technology for top level connections



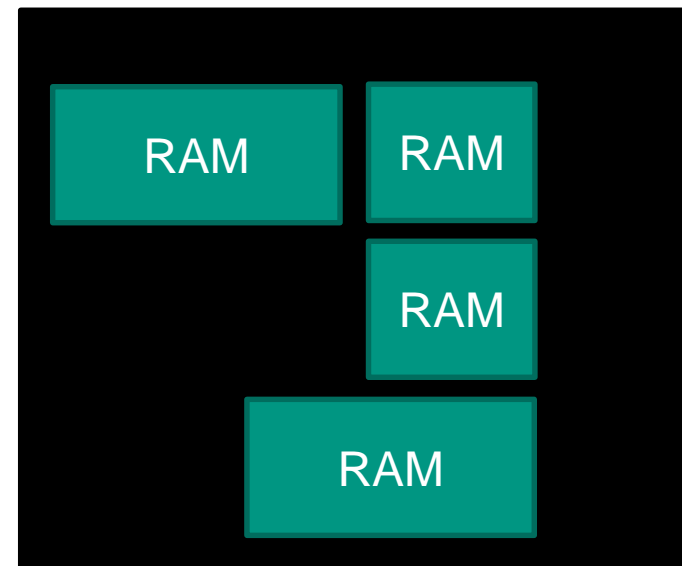
Partition Base Design

Partition Layers strategy example



Hard Macros

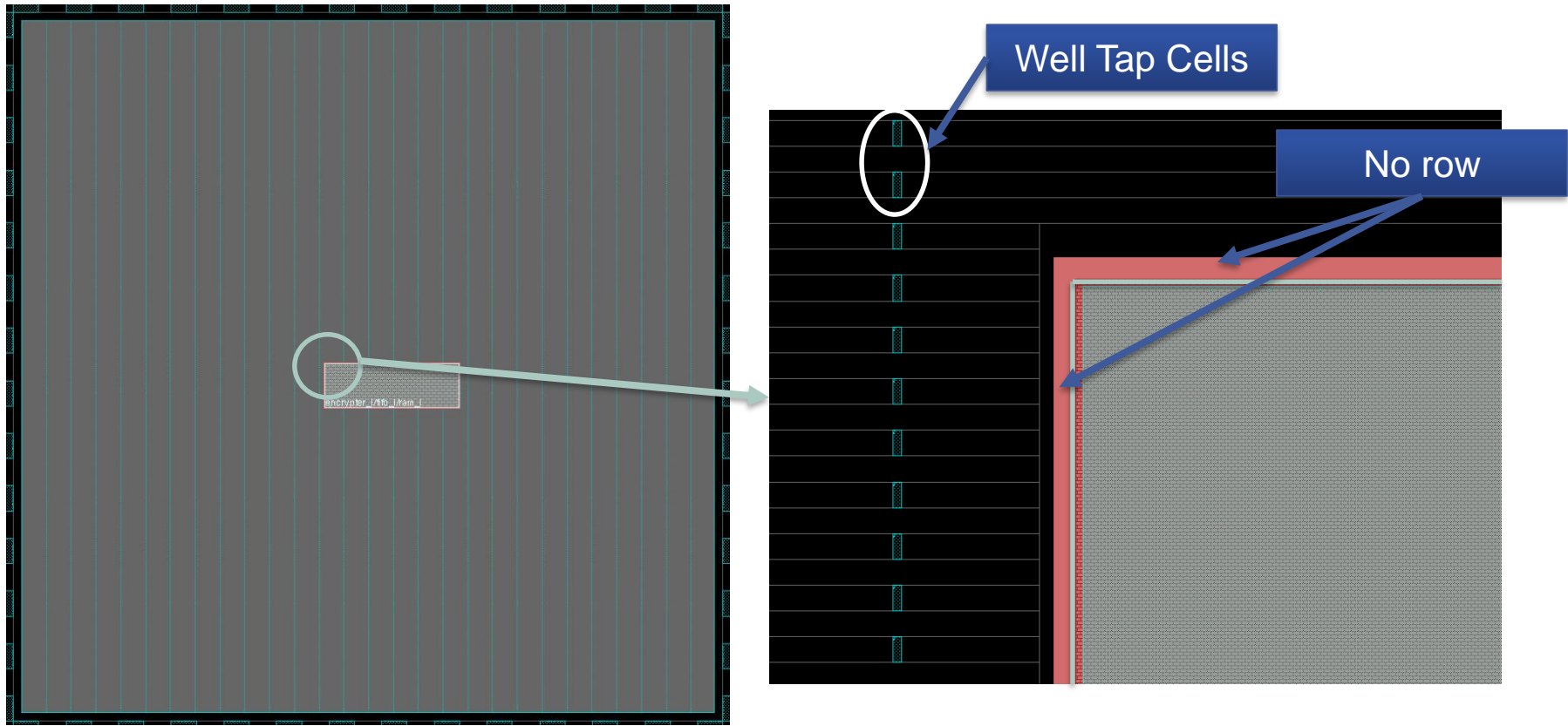
- Hard Macros are like partitions, but ready to use
- LEF file provides size and IO information
- Most common cases:
 - SRAMS
 - PLL
 - Temperature sensor
- Place objects as any other



Sometimes design have many rams

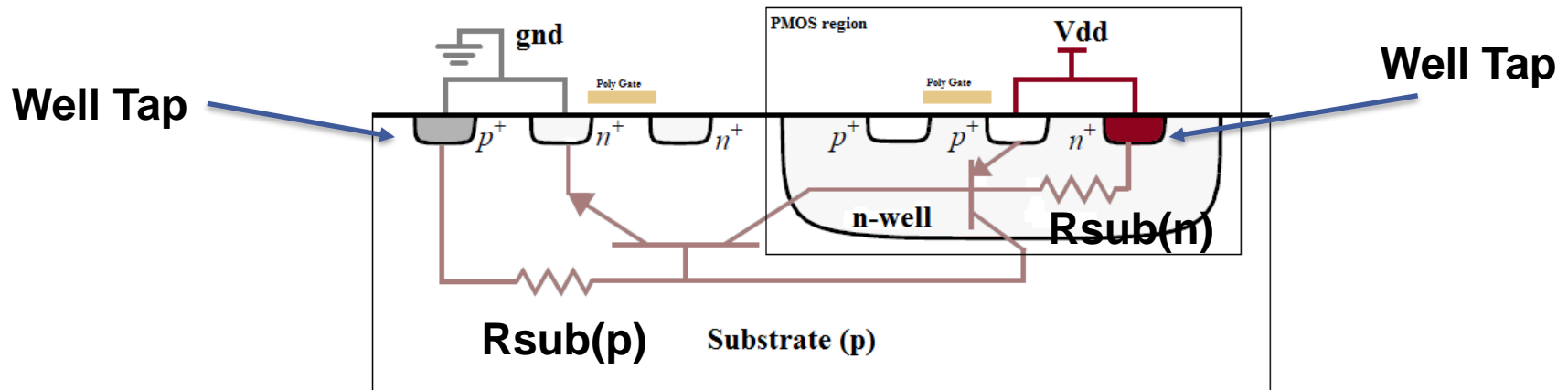
Pre-Placement : Finish floorplan

- Cut the Rows to leave space around the hard macros
- Add specifics like Well-Taps: See Technology documentation



Pre-Placement: WellTap example

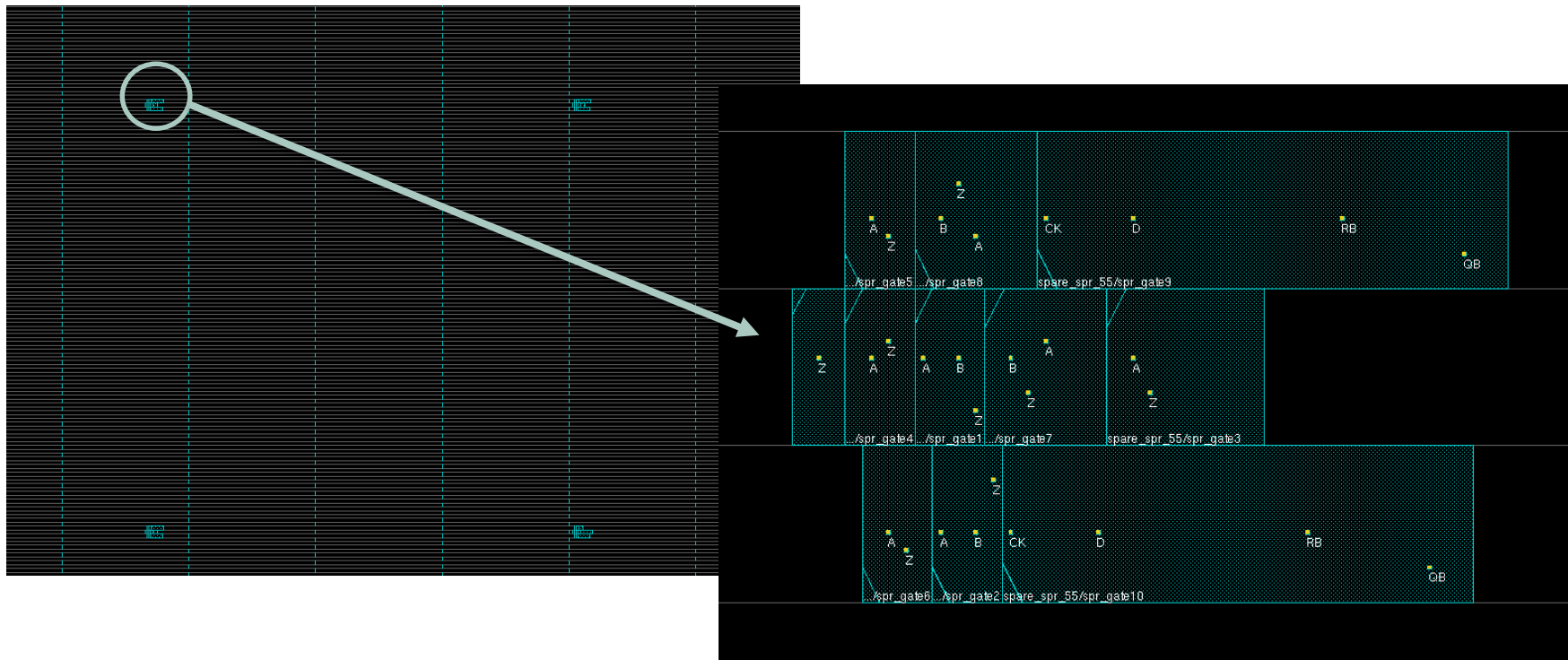
- Latchup: VDD/GND short due to parasitic BJT
- Prevent Latchup: Require enough Well contacts to keep R_{sub} low
- See Technology documentation
- Links to read:
 - <http://www.analog.com/library/analogDialogue/archives/35-05/latchup/index.html>
 - <http://www.ece.drexel.edu/courses/ECE-E431/latch-up/latch-up.html>



<https://en.wikipedia.org/wiki/Latch-up>

Pre-Placement: Spare Cells example

- Some clusters of cells can be added on the core area for latter fixes.
- The production masks can be slightly modified to fix an issue. The spare cells can be used to modify the logic a bit.

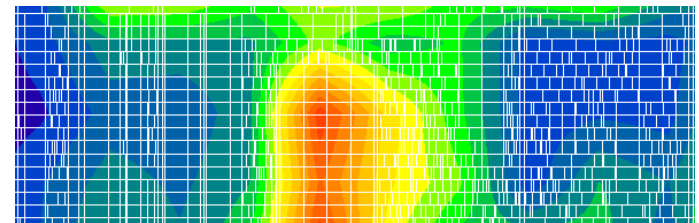
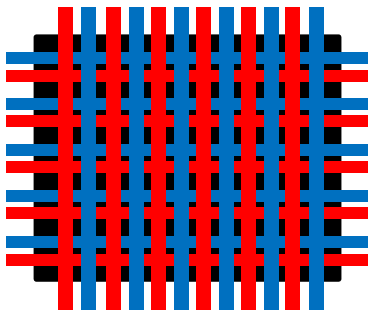


Placement in the Tools

- How does placement happen in the tools?
- Simply programming:
 - Find the object in the design hierarchy if they were synthesised
 - `find /path/to/object`
 - Manually Create Instances for special cells
 - Place with coordinates
 - `placeInstance 100 200 $ram`
- The tools sometimes have special commands to place extra structures:
 - Typically: Fillers
- Remember: This is just software, placement could be prepared anywhere and a script generated, or a DEF file generated etc...

Power planning and Power Analysis

- Power domains:
 - Analog and digital
 - Multiple Digital power domains are possible as well
- Power connections are important
 - If not enough of them the voltage drop across the design can be huge: IR drop picture
- Problem: Functional Power analysis are not simple, runtimes are long
- Solution: Add as much as you can

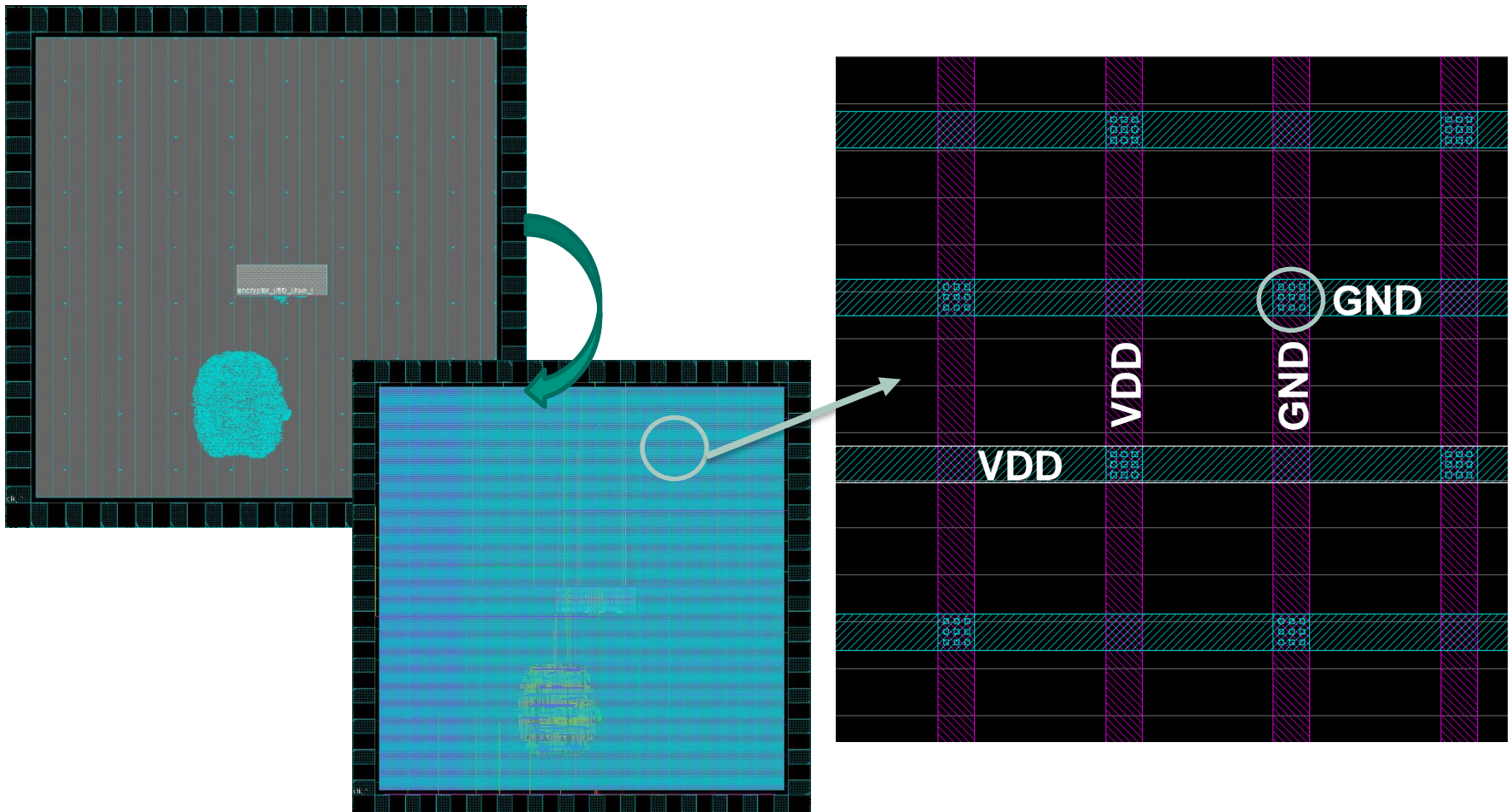


IR Drop example

Source: semiwiki.com

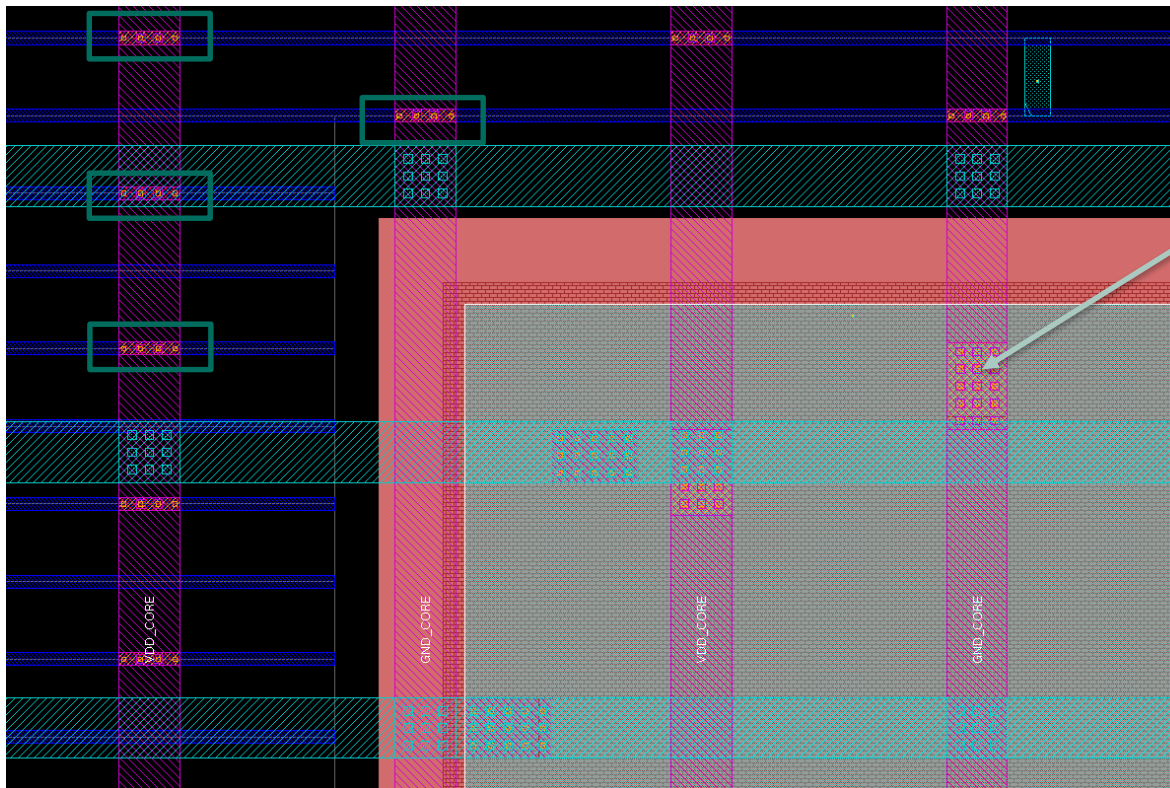
Power Grid: Global Grid

- Use the addStripe command to add power stripes on high metal layers



Power Grid: Rows and Macros connections

- Use the editPowerVia command to add VIA down to the power pins of the standard rows and macros.
- Consult the technology files (like LEF) to find out on which layer the power pins are available



```

PIN GND
DIRECTION INOUT ;
USE GROUND ;
SHAPE RING ;
PORT
  LAYER ME3 ;
  RECT 0.000 101.520 321.350 103.520 ;
  RECT 0.000 1.000 321.350 3.000 ;
  RECT 319.350 1.000 321.350 103.520 ;
  RECT 0.000 1.000 2.000 103.520 ;
END
END GND

```

SRAM LEF

Standard Cells Placement

- Quite trivial, automatically place the cells where can be
 - -> placeDesign in tool
- Tries to optimize timing
- Floorplaning drives the placements
 - I/O
 - Hard macro

Placement constraints

- Standard cells can be constraints to regions and fences
- Relates to FPGA floorplaning as well (box in Xilinx ISE)
- Usually not needed, sometimes useful



Next time

- Timing Configuration
- Clock tree synthesis
- Routing
- Timing Fixes along the design flow

End Slide