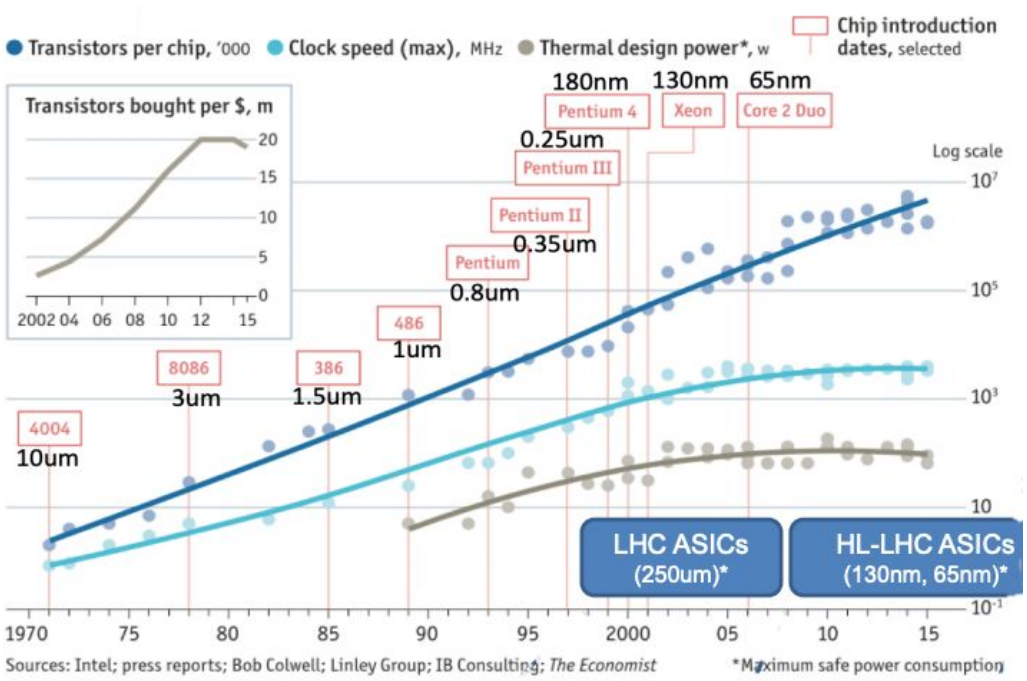
Digital on Top methodology

Prospectives IP2I 2023 sur Futur Collisionneurs FCC

14 - 09 - 2023

B. Nodari

why DoT methodology?



source: https://ep-news.web.cern.ch/chips-new-ep-ese-service-hep-community

With the **decrease in the size of the technology,** accompanied by a proportional saving in power consumption, the number of transistors that can be integrated on a single chip largely increases.

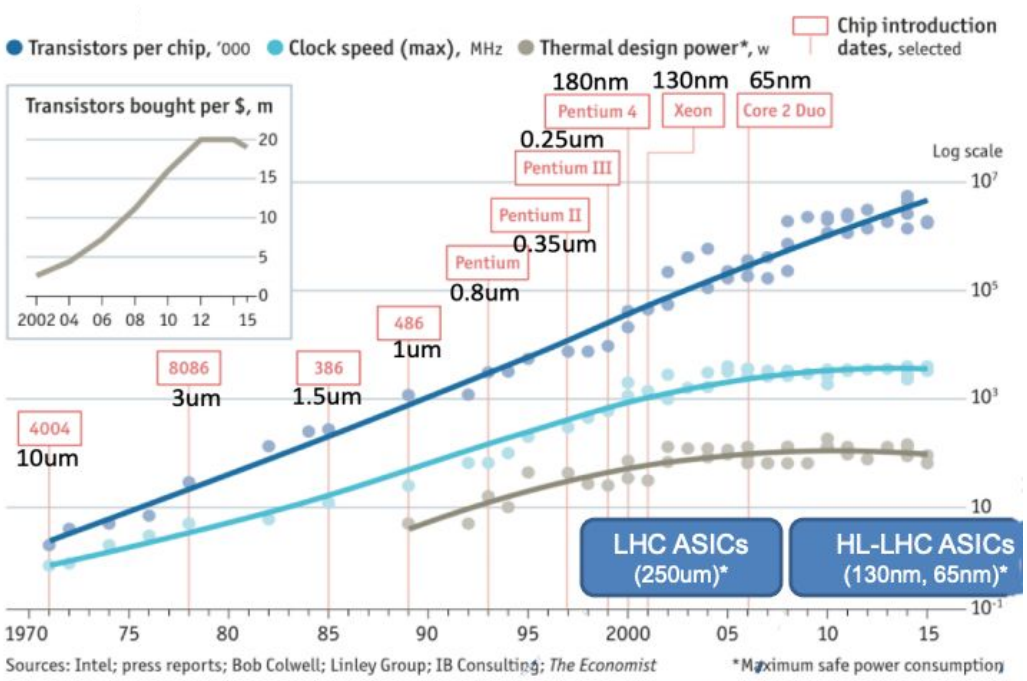
Reliably manufacturing multi-billion fully-functional transistors per chip requires many more processing steps.

This increased complexity in technology also affects circuit design.

Designs might be confronted with the tangible **risk of failure or delay** with potentially severe consequences on the physics programme.

The new and more sophisticated design tools and design flows that are now available to chip designers must be used in these technologies.

why DoT methodology?



source: https://ep-news.web.cern.ch/chips-new-ep-ese-service-hep-community

Chip designers have to face the challenges associated with the new level of design and verification complexity.

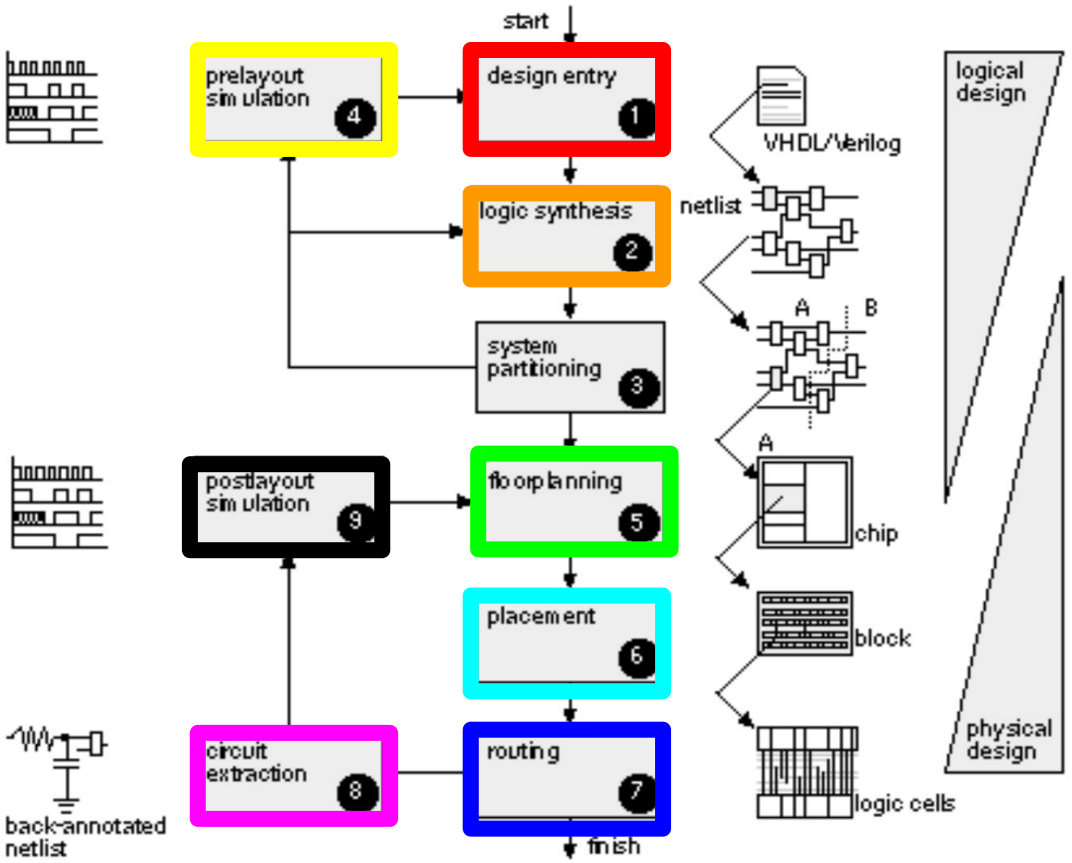
This means developing and following scrupulously the **Digital-on-Top design methodology**.

In this methodology, **a fully scripted automated flow** is used to drive the digital tools to design and implement the digital circuitry around the analog blocks.

*The **DoT design implementation** and **System level verification** methodologies **must be adopted** systematically to avoid expensive and time-consuming errors even if this implies a significant increase in design time and resources.*

Digital design flow

A **design flow** is the sequence of the steps to design an ASIC following the DoT methodology: based on a series of scripts it permits to reach the final file (GDS) for the Tape-out phase (final step before the ASIC fabrication) starting from the behavioral description of the architecture (RTL).

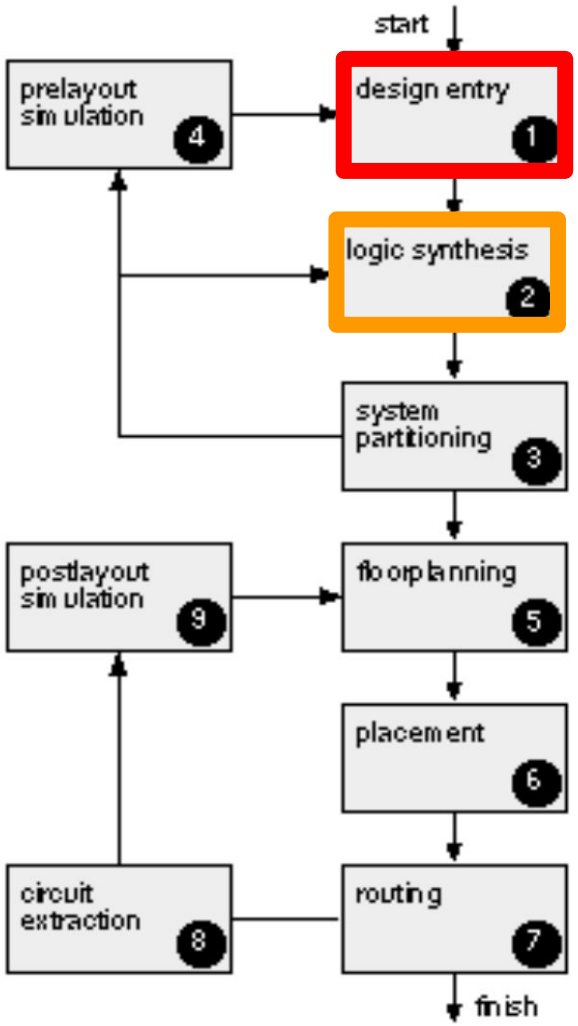


Cadence tools for DoT:

- GENUS
- INNOVUS
- TEMPUS
- VOLTUS
- VIRTUOSO
- CALIBRE
- QUANTUS
- XCELIUM

source: “ASICs” Michael J.S. Smith Addison-Wesley ‘97

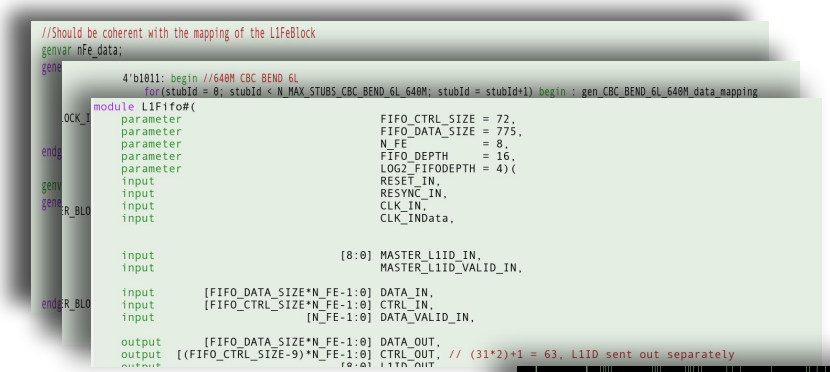
Digital design flow



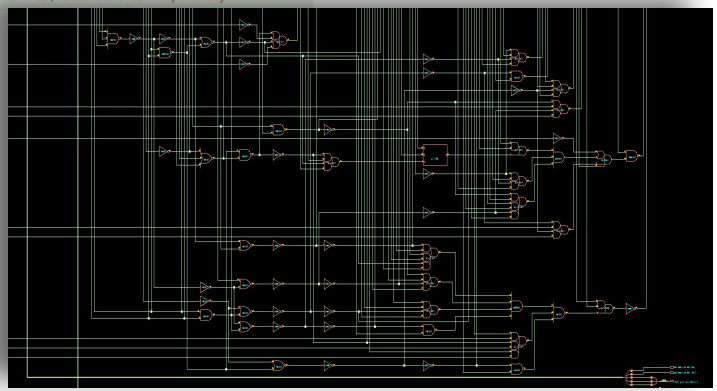
1. Design entry: input description of the circuit (RTL) given through a hw description language (VHDL or Verilog/SystemVerilog).

2. Logic synthesis: the process that converts RTL into a technology-specific gate-level netlist, optimized for a set of pre-defined constraints. [GENUS]

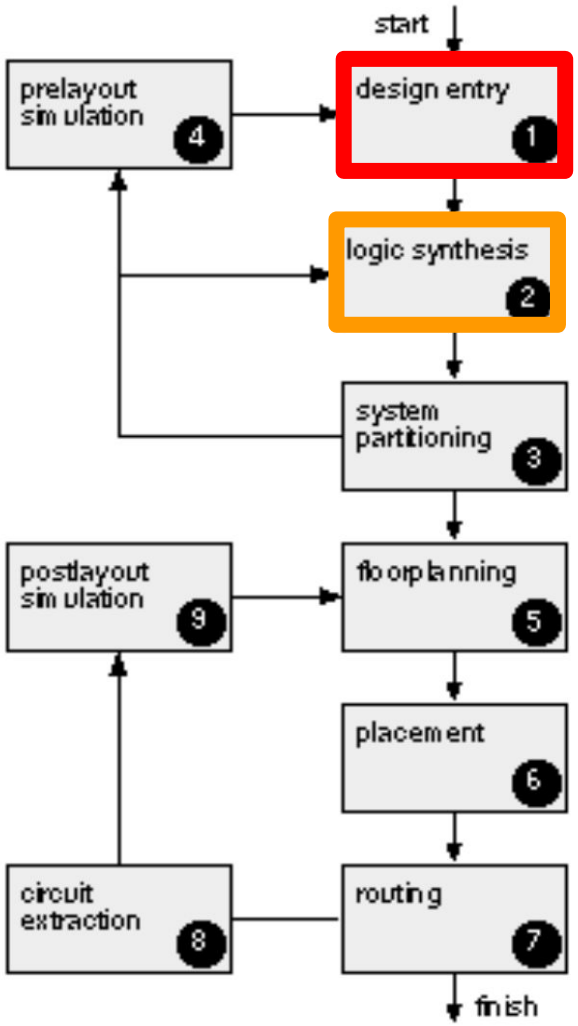
CIC RTL code in SystemVerilog language:



```
//Should be coherent with the mapping of the L1FeBlock
genvar nFe_data;
gene
        4'b1011: begin //640M CBC BEND 6L
            for(stubId = 0; stubId < N_MAX_STUBS_CBC_BEND_6L_640M; stubId = stubId+1) begin : gen_CBC_BEND_6L_640M_data_mapping
module L1Fifo#(
    parameter                               FIFO_CTRL_SIZE = 72,
    parameter                               FIFO_DATA_SIZE = 775,
    parameter                               N_FE           = 8,
    parameter                               FIFO_DEPTH     = 16,
    parameter                               LOG2_FIFODEPTH = 4)(
    input                                   RESET_IN,
    input                                   RESYNC_IN,
    input                                   CLK_IN,
    input                                   CLK_INData,

    input                             [8:0] MASTER_L1ID_IN,
    input                                   MASTER_L1ID_VALID_IN,

    input       [FIFO_DATA_SIZE*N_FE-1:0] DATA_IN,
    input       [FIFO_CTRL_SIZE*N_FE-1:0] CTRL_IN,
    input                   [N_FE-1:0] DATA_VALID_IN,

    output      [FIFO_DATA_SIZE*N_FE-1:0] DATA_OUT,
    output  [(FIFO_CTRL_SIZE-9)*N_FE-1:0] CTRL_OUT, // (31*2)+1 = 63, L1ID sent out separately
```



Digital design flow



1. Design entry: input description of the circuit (RTL) given through a hw description language (VHDL or Verilog/SystemVerilog).

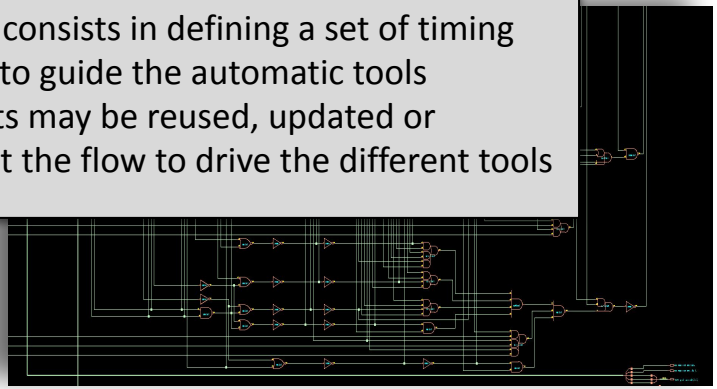
2. Logic synthesis: the process that converts RTL into a technology-specific gate-level netlist, optimized for a set of pre-defined constraints. [GENUS]

CIC RTL code in SystemVerilog language:

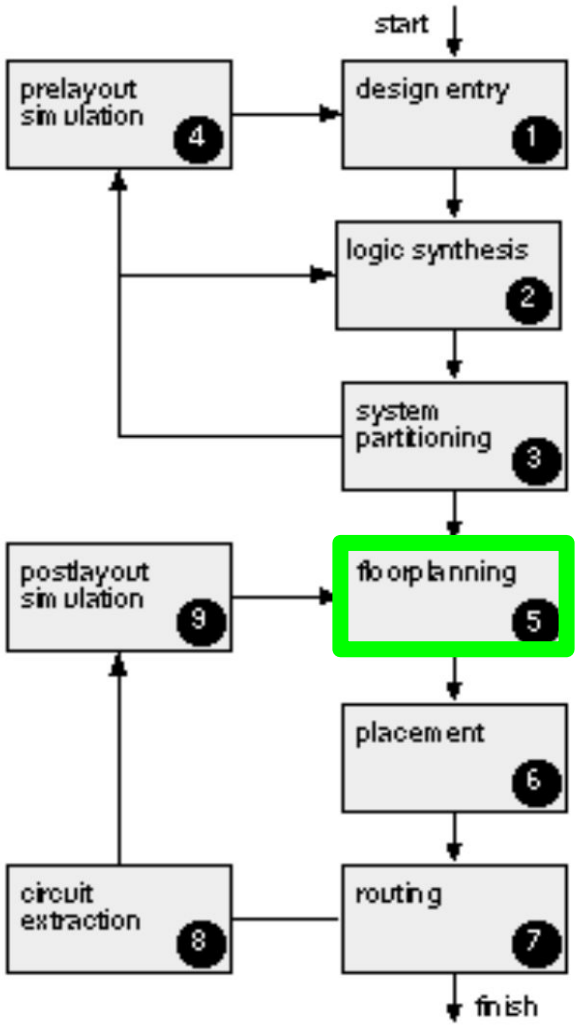
```
//Should be coherent with the mapping of the L1FeBlock
genvar nFe_data;
gene
        4'b1011: begin //640M CBC BEND 6L
            for(stubId = 0; stubId < N_MAX_STUBS_CBC_BEND_6L_640M; stubId = stubId+1) begin : gen_CBC_BEND_6L_640M_data_mapping
module L1Fifo#(
    parameter                       FIFO_CTRL_SIZE = 72,
    parameter                       FIFO_DATA_SIZE = 775,
    parameter                       N_FE           = 8,
    parameter                       FIFO_DEPTH     = 16,
    parameter                       LOG2_FIFODEPTH = 4)(
    input                           RESET_IN,
    input                           RESYNC_IN,
    input                           CLK_IN,
    input                           CLK_INData,
```

digital design flows are performance-driven:

- speed/power/area are the three common factors
- the first design step consists in defining a set of timing constraints in order to guide the automatic tools
- this set of constraints may be reused, updated or modified throughout the flow to drive the different tools

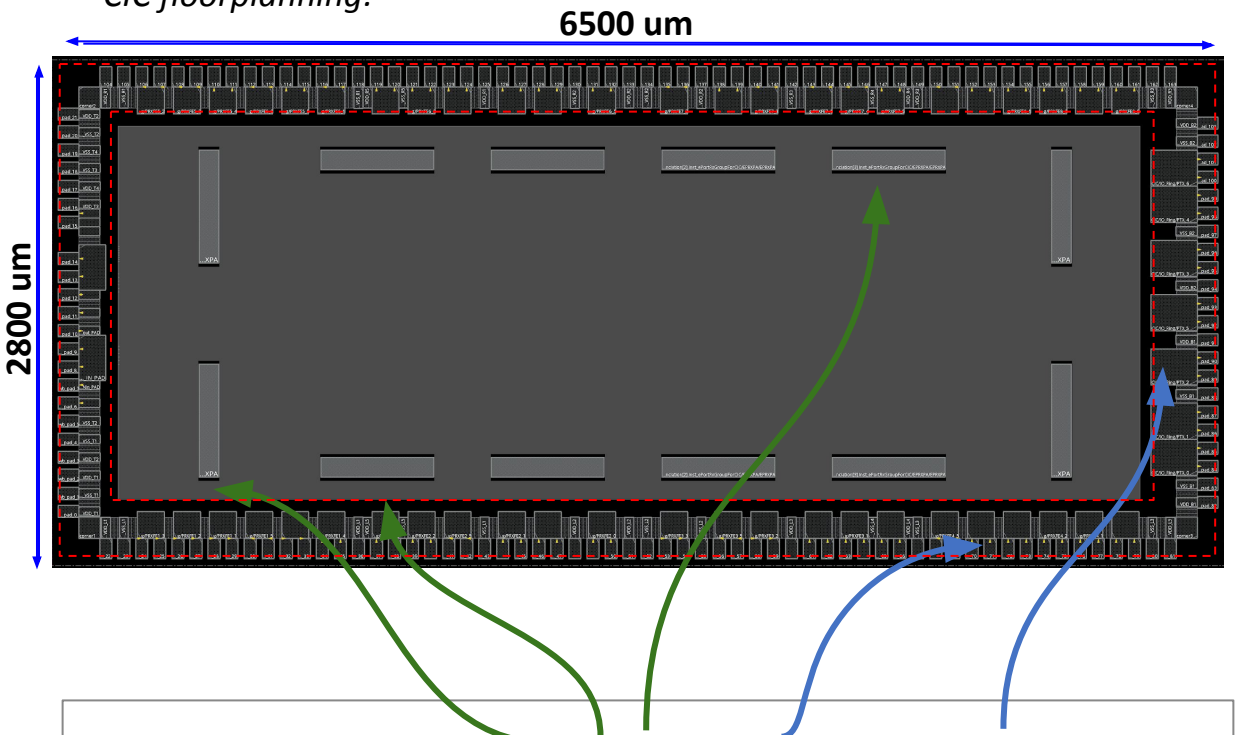


Digital design flow



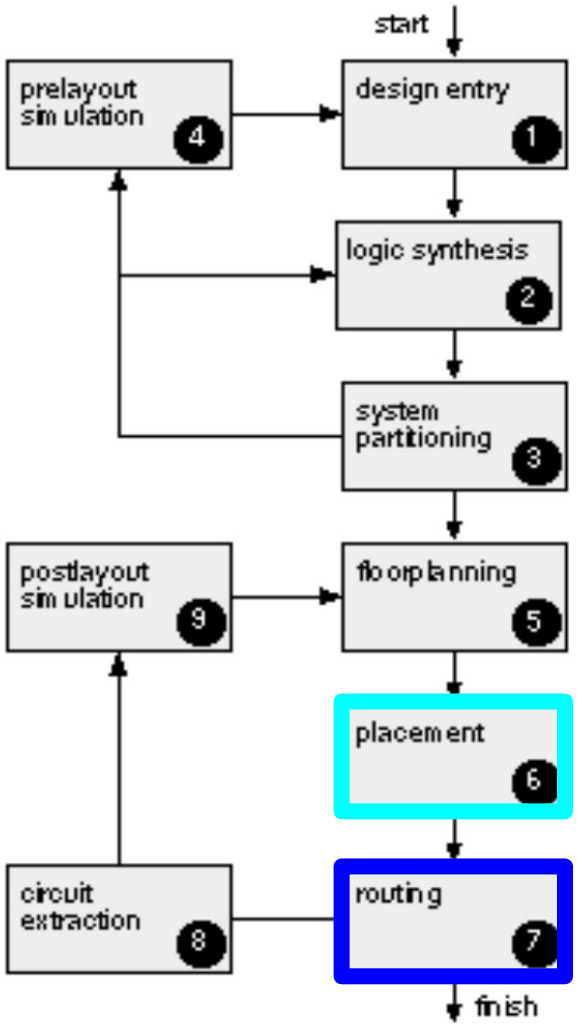
5. Floorplanning: the main blocks of the netlist are arranged all over the available area taking into account the design performances criterias. [*INNOVUS*]

CIC floorplanning:



- IP (analog macros) used (phaseAligner, sLVS receiver + transmitter cells, ESD protections).
- Periphery ring (48 sLVS receiver pads, 9 sLVS transmitter pads, 8 CMOS pads).

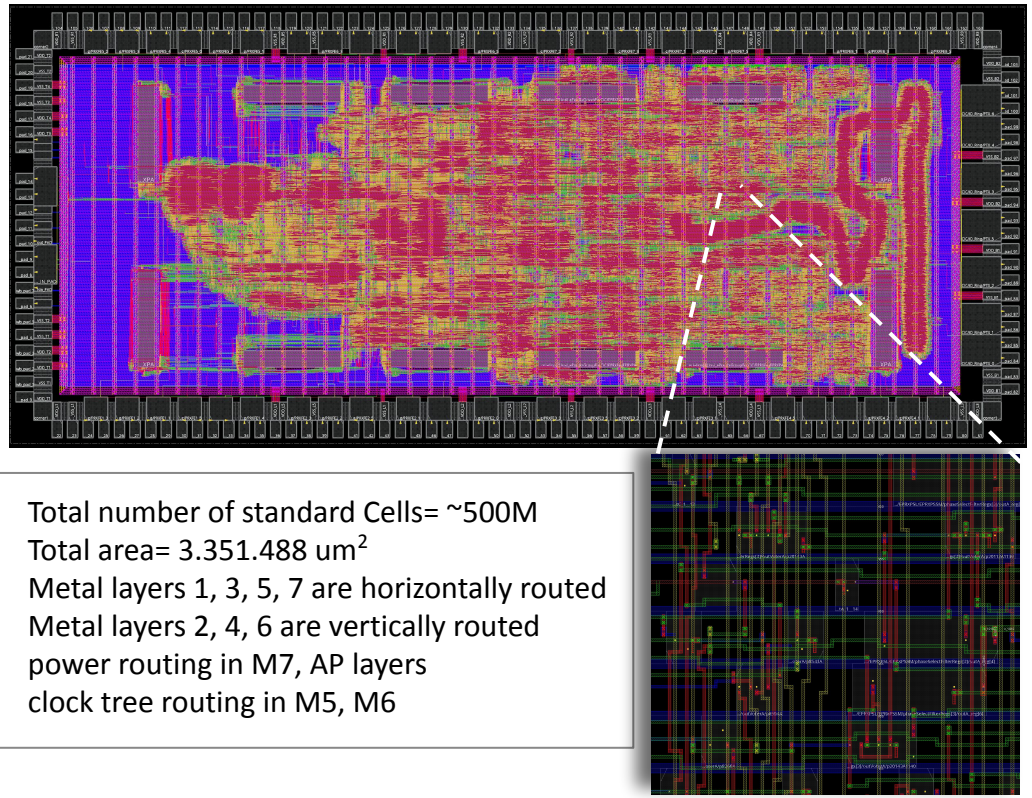
Digital design flow



6. Placement: the standard cells and any other macro block are placed following the design constraints. [*INNOVUS, TEMPUS, VOLTUS*]

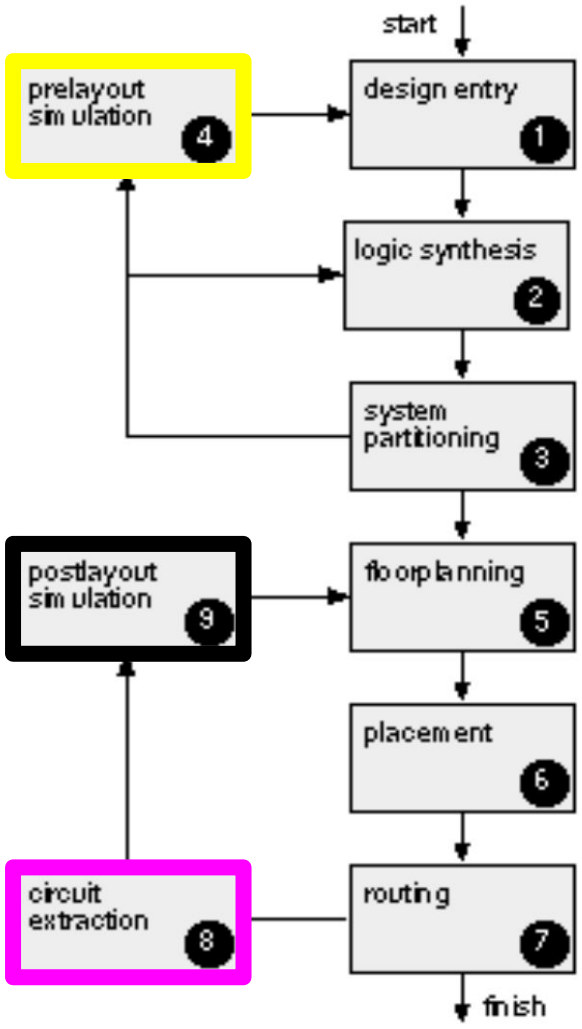
7. Routing: the physical interconnections between cells are traced in the available metal layers. [*INNOVUS*]

CIC placement and routing:



- Total number of standard Cells= ~500M
- Total area= 3.351.488 um^2
- Metal layers 1, 3, 5, 7 are horizontally routed
- Metal layers 2, 4, 6 are vertically routed
- power routing in M7, AP layers
- clock tree routing in M5, M6

Digital design flow

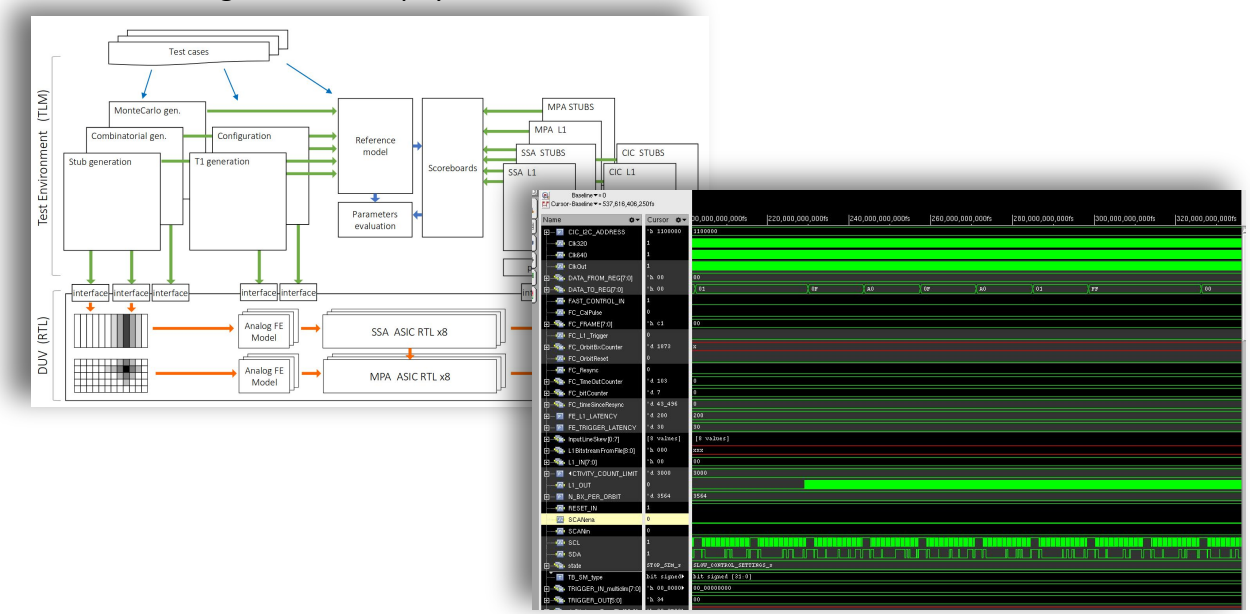


All functionalities and timing constraints are often checked during the design flow by simulations with testbenches: the goal is to test design functionalities within a script-based framework (python) or UVM.

4. Pre-layout simulation: functionality of the netlist is checked and timing constraints are verified. [*XCELIUM*]**.**

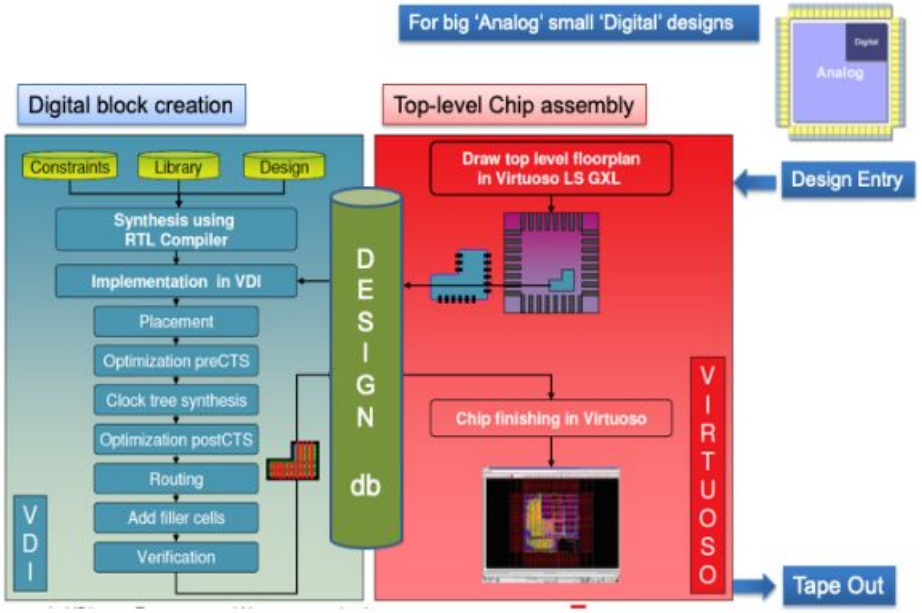
8. Circuit extraction: the parasitics created by the metal interconnections are evaluated and written back into the netlist (back-annotation). [*QUANTUS*]

9. Post-layout simulation: functionality and timing constraints are again checked this time including all available physical informations.

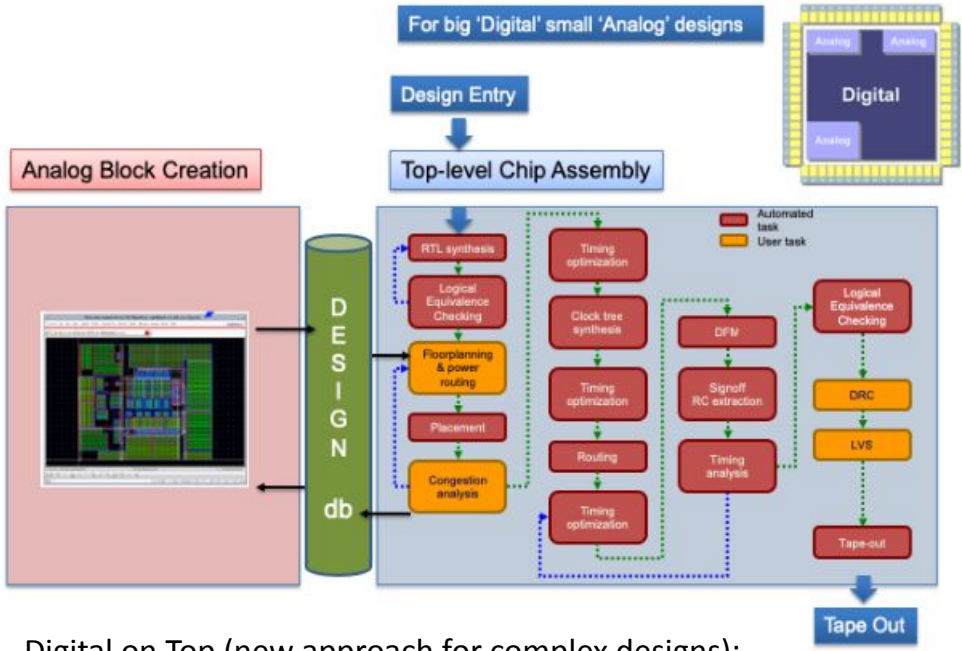


Digital vs Analog on Top flows

Analog on Top workflow



Digital on Top workflow



Analog on Top (historical design flow):

- refers to the design flow where the top level of the design is represented as a schematic.
- *Virtuoso* tool is typically the environment where the entire top level design is assembled.
- Blocks are prepared separately then added one-by-one to the design.
- Ok for small designs but extremely risky for complex ASICs.

Digital on Top (new approach for complex designs):

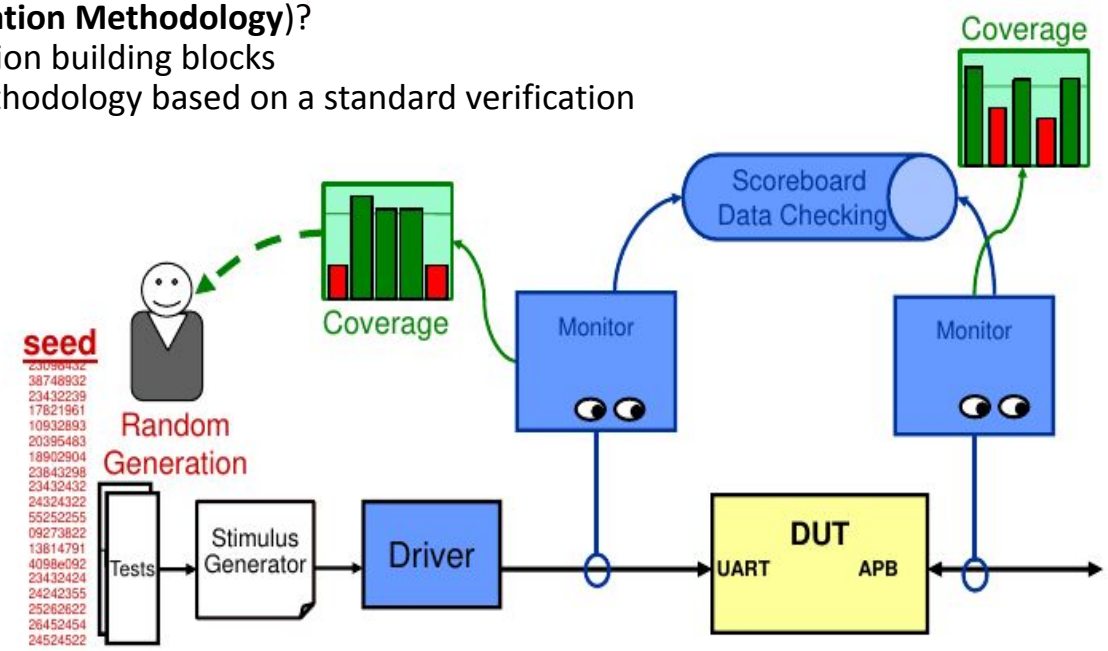
- refers to the design flow where the top level of the design is represented as a Verilog/SystemVerilog netlist.
- *Innovus* tool is typically the environment used for assembling the entire top level design.
- High level simulation and verification throughout design requires different skill set and generally more resources
- strong mitigation of risk if all steps are fully followed

System level verification

- Verification: the key to unlocking quality and innovation in future ASICs for HEP experiments.
- Verification accounts to 70% of overall design costs and total ASIC design time.
- Reasons for verification:
  - complex designs will have complex bugs;
  - cost of prototyping will be higher in future so we need to reduce the number of prototyping runs (typically 2 or more engineering runs before production);
- System level verification framework :
  - build a realistic world (= verification environment) around a design (=Design Under Test)
  - exercise the DUT in all possible ways
  - check all the outcomes of the DUT are as expected
- what is UVM (**Universal Verification Methodology**)?
  - A class library of verification building blocks
  - A proven verification methodology based on a standard verification component architecture

Components of a MDV environment

- Automated Stimulus Generation
- Independent Checking
- Coverage Collection



source: Cadence

Backup

	A	A/d	A/D	D/A	D/a	D
Methodology	Analog on Top (AoT)		Mixed-Signal on Top (MSoT)		Digital on Top (DoT)	
Design Flow	Schematic-Driven		Netlist-Driven			
Top level Connectivity	Schematic		Verilog		Verilog	
Design Characteristics	• Top level is analog • Standard cell digital designed in a digital flow		Analog blocks and standard cell digital mixed at the top level		• Top level is digital • Analog designed in an analog flow	
Floorplanning	Virtuoso floorplanner and Virtuoso Digital Implementation		Virtuoso floorplanner and Innovus		Innovus	
Analog Content	Main and Top Level		Co-Designed		Separate Hierarchy	
Digital Content	Separate Hierarchy		Co-Designed		Main and Top Level	
Routing	• VSR for top level and analog • NR for routing within the digital blocks		• VSR for analog blocks and NR for digital blocks • Top level uses VSR for analog and NR for digital		• Top level is digital • Analog designed in an analog flow	
Chip Integration	Virtuoso		Innovus		Innovus	
Signoff	MSPS		Tempus		Tempus	
Chip Finishing	Virtuoso		Virtuoso		Virtuoso/Innovus	

source: Cadence

Vocabulary

- **SOS**: integrated natively tool into Virtuoso to manage design data including IPs, PDKs etc from concept-to-tapeout.
- **LVS (= Layout vs Schematic):** one of the final step of DoT and AoT flow in order to verifies the connectivity and ensures the functionality of the layout.
- **DRC (= Design Rule Checks):** one of the final step of DoT and AoT flow in order to verifies that the given layout satisfies the design rules provided by the fabrication unit.
- **SDC (= Synopsis Design Constraints):** timing constraints (for DoT methodology) based on clock definitions (time period, duty cycle,..) and timing exceptions (false paths, asynchronous paths,..).
- **CTS (=Clock Tree Synthesis):** is one of the most important stage in PnR. It's the process of distributing the clock and balancing the load.
- **RTL**: descriptive code written in hw format (VHDL, Verilog, SystemVerilog).

ASICs ? FPGAs? I'm confused!

ASIC: an Integrated Circuit designed for a particular system or end-use

- Broad definition including analog, digital or memory circuits as long as custom designed
- In contrast to standard IC (memories, controllers, microprocessors)
- The designer is (often) the customer
- It's both a class of IC products and a design methodology
- (some) types of ASICs:
  - ***Full-Custom***: some or all of the cells, circuits or layouts are specifically designed
  - ***Standard-Cell Based***: predesigned logic cells are placed and interconnected

FPGA/(or PLD): standard IC that may be configured or programmed for a specific application

- It provides the customizability of a digital ASIC without the need to design and fabricate a new device for each application