ELECTRONIC SYSTEMS DESIGN METHODOLOGY

Santosh S Malagi

Contents

- Motivation
- Electronic Systems – an Overview
- Electronic Systems Design
    - Standard Cell-Based ASIC Design
    - FPGA based Design
    - Analog VLSI Design
    - Mixed-Signal VLSI Design
- Executing Design Methodologies using EDA tools
- Summary
- References and More Resources

MOTIVATION

Who? What? and Why?

What?

- **Economics of Electronic Systems Design (ESD)**
  - Understand various market players - EDA/IP, fabless, fablite, foundry etc.
- **Introduction to basics of Electronics Systems Design Methodology**
  - VLSI design methodologies – Digital, Analog and Mixed-Signal
  - Various EDA tools involved in the design process
  - ‘Hardware-Software Co-design’ using ‘Platform SoCs’ or ‘MpSoC’s’

What is NOT covered?

- Transistors, semiconductor physics etc. deliberately left out to focus on higher levels of abstraction
- Memory design will be included in future revisions
- Analog design (the presenter is a ‘Software Engineer’ who writes C++ but knows a bit of DFT/ATPG) ☺
- Advanced topics in Physical Design and Implementation – beyond author’s level of expertise

Why?

- ‘Software Engineers’ interested to learn more about VLSI and chip design flow
- Many want to learn about different EDA tools and how they fit in the overall design flow
- Stuff you don’t learn in EE/ECE/CE curriculum but are expected to pick up on your own
- Clarify basic terminology and three-four letter acronyms - VLSI, ASIC, ASSP, SoC etc.
- Provide pointers and useful references for those looking for more

Collection of ‘**Personal Notes**’ and ‘**Illustrations’** attempts to cover the following topics:

- **Electronic Systems Design Methodology**
- **EDA Software**
- **VLSI Test Engineering**

- The depth of material is akin to a ‘motivational’ or ‘101’ lecture given in relevant EE/CE courses
- Limited to my knowledge and experience gained in the past 4.5 years

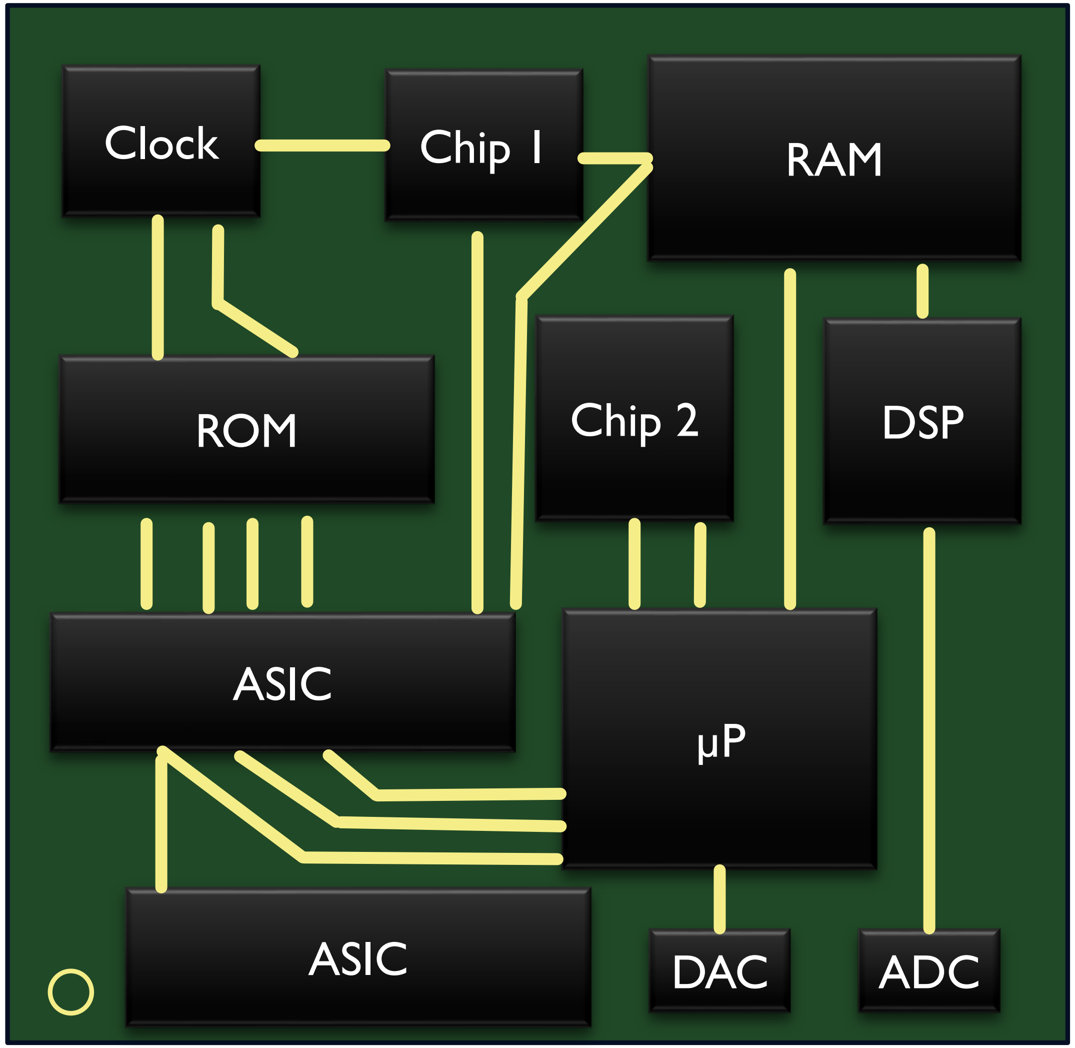
ELECTRONICS SYSTEMS – an OVERVIEW

Integrated Circuits (IC), economic impact and the design ecosystem

Preliminaries

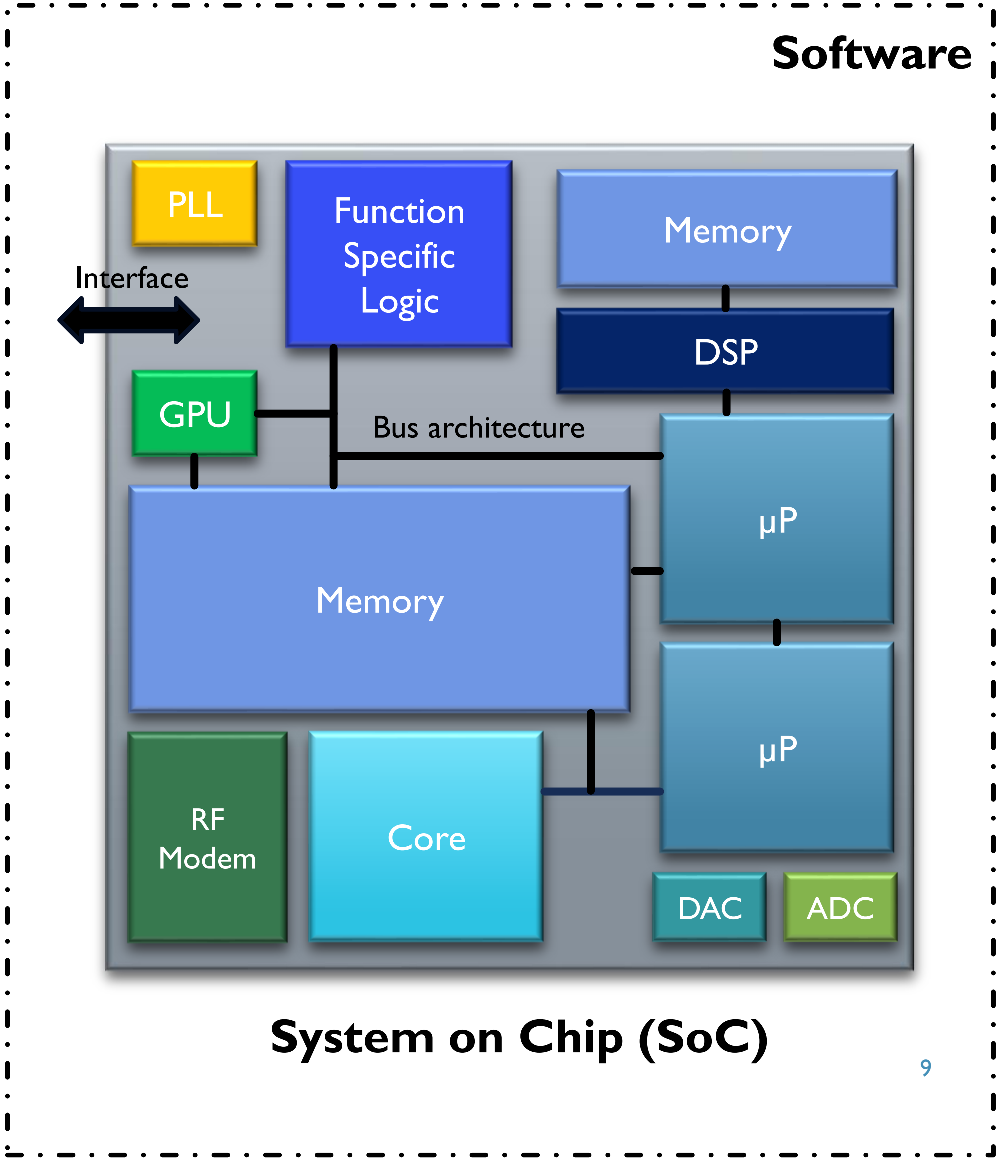
- Integrated Circuit (IC) – electronic circuit fabricated on a single piece of semiconductor material
- Wafer – extremely thin circular disk of semiconductor material used to fabricate multiple ICs
- Die – individual IC ‘sliced’ from a wafer, ‘naked die’ before packaging
- Chip (aka microchip) – an IC which is hermitically sealed/packaged
- Printed Circuit Board (PCB) – polysilicon material used to interconnect and hold together several chips to form an ‘Electronic System’

Electronic Systems Evolution



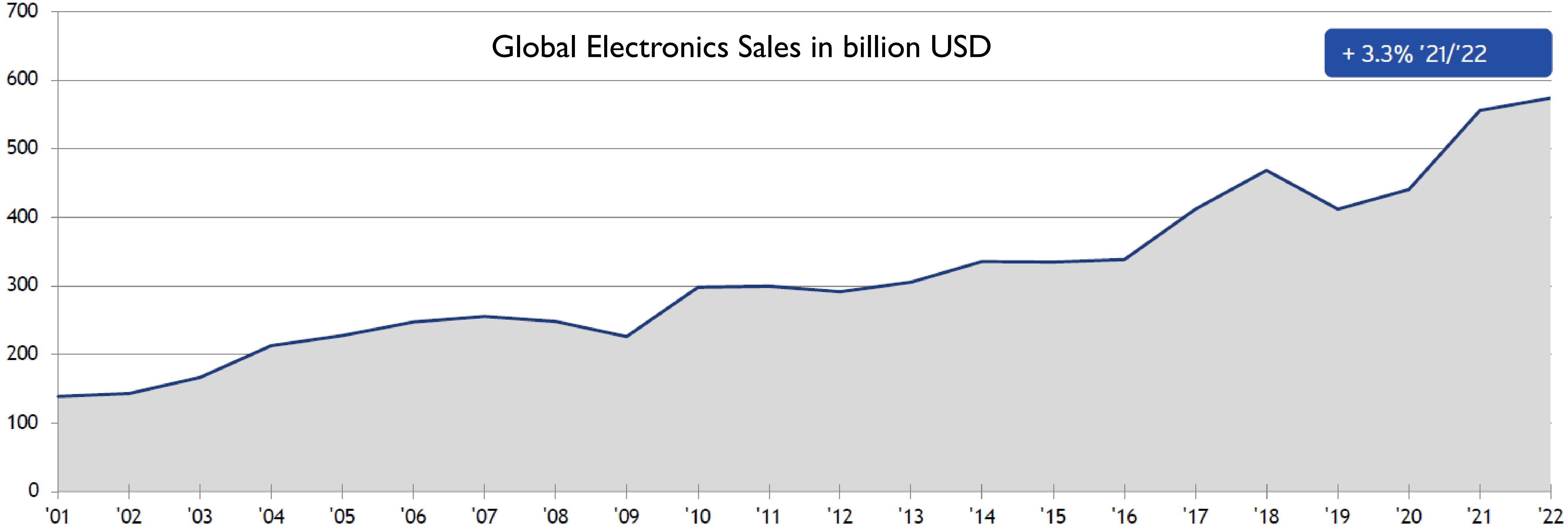
Printed Circuit Board with multiple ICs

Platform SoC



System on Chip (SoC)

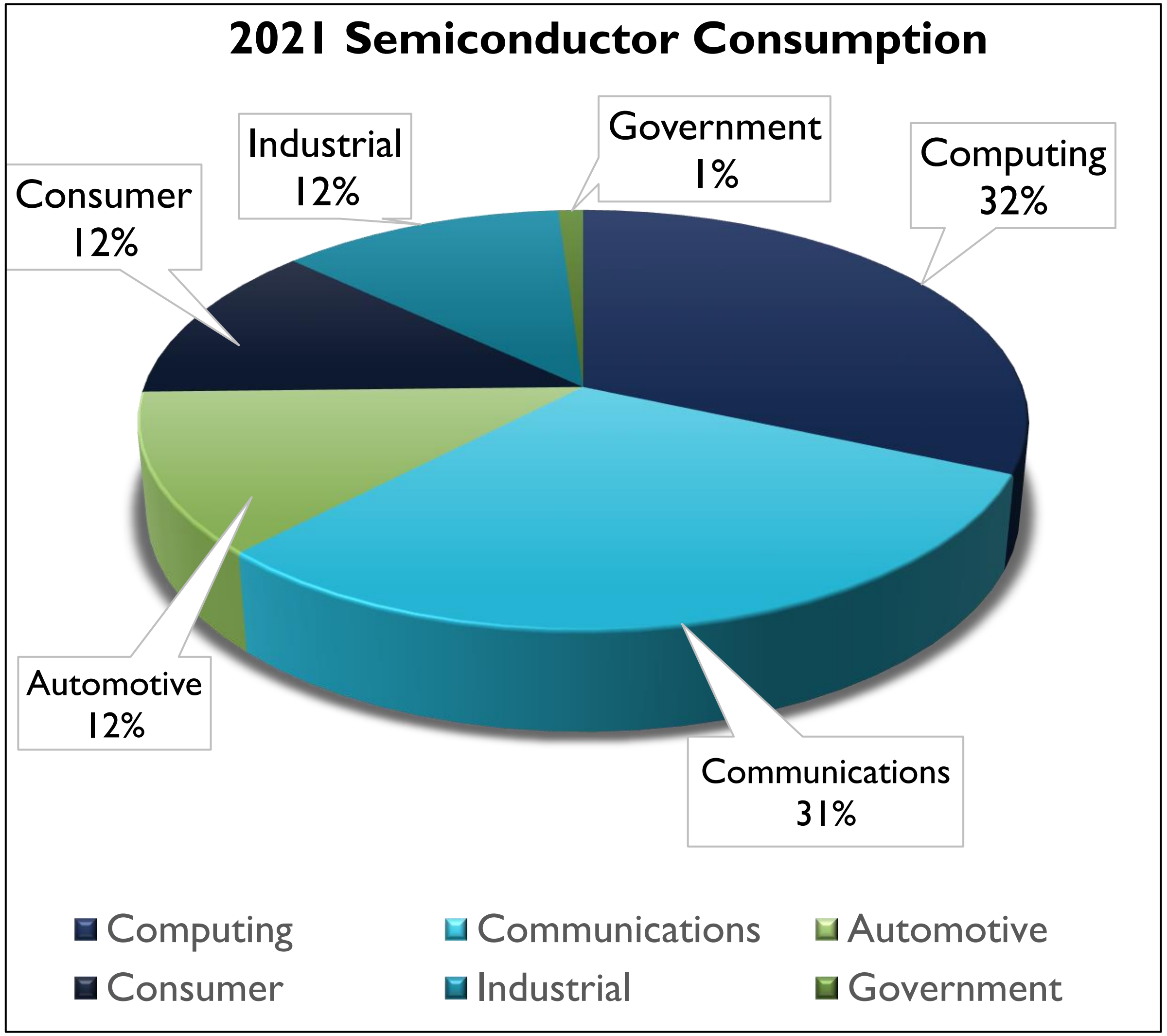
Global Electronic Systems Market – by Numbers



- Electronics is a major contributor to the world GDP, enabler of the modern 'digital economy'
- Annual growth rate of 6.67%, sales increased from $139.0 billion in 2001 to $574.0 billion in 2022
- Significant R&D investment, requires skilled labor and value creation through 'product differentiation'

Reference – Semiconductor Industry Association (SIA) Factbook 2023

End Markets for Electronic Systems



Reference – 2022 State of the US Semiconductor Industry Report - SIA

Electronic Systems – Application Areas:

- **Computing**
  - Cloud and distributed systems, PC, office equipment, tablet PCs etc.
- **Communications**
  - Wired/Wireless devices, routers, switches, networking, telecommunication infrastructure etc.
- **Automotive**
  - ECU, infotainment, power train, safety, ADAS etc.
- **Consumer**
  - All kinds of consumer appliances – smart TVs, digital cameras, smartphones, medical devices etc.
- **Industrial**
  - Industrial automation and robotics, manufacturing, monitoring, surveillance, etc.
- **Government**
  - Defense, aerospace etc.

Who are the Market Players?

- **Electronic Design Automation/IP Vendors -** software tools, virtual components, IP blocks etc.
- **Fabless companies -** design chips, outsource manufacturing/packaging/test to foundry/OSATs
- **Integrated Device Manufacturer (IDM)** – design and fabricate chips in house
- **Fablite companies -** outsource standard wafer processing, retain specialized manufacturing
- **Foundry -** wafer processing, chip fabrication, heterogenous integration etc.
- **Equipment** – design and manufacture technology for chip fabrication, packaging, test etc.
- **Outsourced Semiconductor Assembly and Test (OSAT) and Contract Manufacturers**
- **Materials Providers** – semiconductor materials, chemicals for chip fabrication, packaging etc.

More….

Integrated Circuits – Digital, Analog, and Mixed-Signal

Digital

- **<u>Logic</u>**
  - General purpose
  - Microprocessor (CPU)
  - Graphics Processing Unit (GPU)
  - Microcontroller (MCU)
  - Field Programmable Logic (FPL)
- **<u>Memory</u>**
  - Volatile – SRAM and DRAM
  - Non-volatile – ROM, Flash etc.

Mixed-Signal

- Analog to Digital Converters (ADC)
- Digital to Analog Converters (DAC)
- Digital Signal Processors (DSP)
- Delta-Sigma Modulators
- Error Detection/Correction ICs

Others

- Micro-Electro Mechanical Systems (MEMS)
  - Sensors
  - Actuators
- Photonic/Optoelectronic ICs
- Emerging Technologies

Analog

- Single-stage Amplifiers
- Differential Amplifiers
- Operational Amplifiers (OpAmps)
- Power Amplifiers
- Comparators
- Radio Frequency Integrated Circuits (RFIC)
- Phase Locked Loops (PLLs)

Digital ICs - operate on discrete logic high '1' or logic low '0' signal levels
Analog ICs – operate on continuous time varying signals between two levels
Mixed Signal ICs – operate on both digital and analog signals, often convert between the two (DAC, ADC)

Integrated Circuits – Market's Perspective

- **General Purpose IC**
  - Generic functionality, IC sold in huge quantities e.g., 74xx, 555, RAM, ROM etc.
- **Application Specific Integrated Circuit (ASIC)**
  - IC designed for a specific purpose/application area
  - **Application Specific Standard Product (ASSP)**
    - IC designed and optimized for a specific purpose, sold to numerous customers
    - E.g., Qualcomm sells Snapdragon processors to Samsung, MediaTek, Lenovo etc.
  - **User Specific Integrated Circuit (USIC)**
    - IC designed and optimized by a company for its specific use case, product differentiation
    - E.g., Apple designs 'Apple Silicon' for use ONLY in Apple products

What is a System on Chip (SoC)?

A SoC is an ASIC which integrates a complete computing system on a single chip (SiP – System on Package). Most often an SoC includes a Central Processing Unit (CPU), Graphics Processing Unit (GPU), embedded memory (SRAM), RF modem, sensors, I/O interfaces along side specialized hardware e.g., hardware accelerators on a single substrate or microchip.

***All SoCs are ASICs, but all ASICs need not be SoC's!**

System on Chip (SoC)

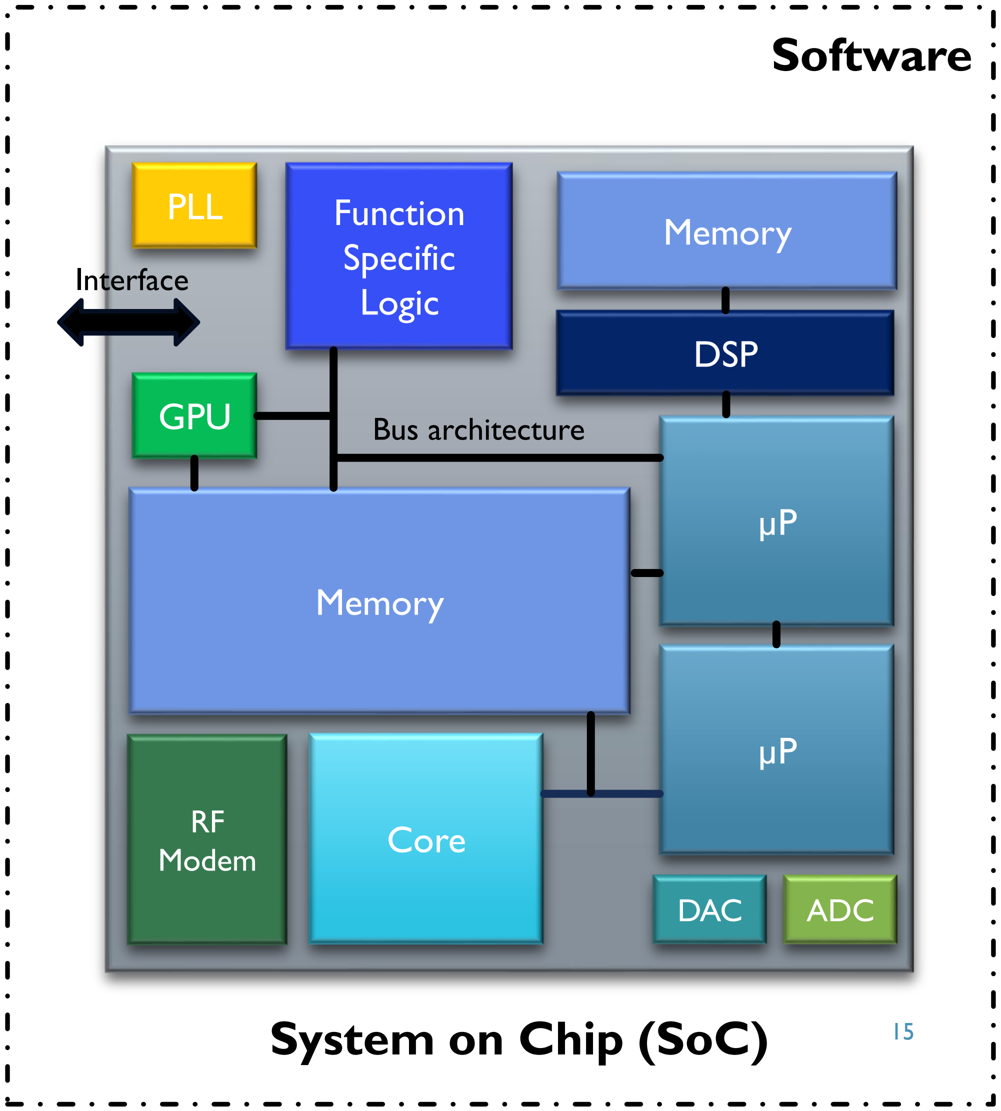
- A SoC combines multiple stand-alone circuits on a single silicon substrate – thereby creating a complete ‘System on Chip’
- ‘**Platform SoC**’ = ‘Software Stack’ + SoC
- **MpSoC =** Multiprocessor System on Chip, SoC with multiple processors

- **Benefits to the SoC approach:**
  - accommodate “increased functionality for the same price”
  - reduce the number of discrete components – hence the points of failure
  - faster, powerful systems at reduced price

- **What enabled the SoC revolution?**
  - Continued progress in IC fabrication technology
  - EDA tools, IP ecosystem and design methodologies
  - Market drivers – demand for more functionality

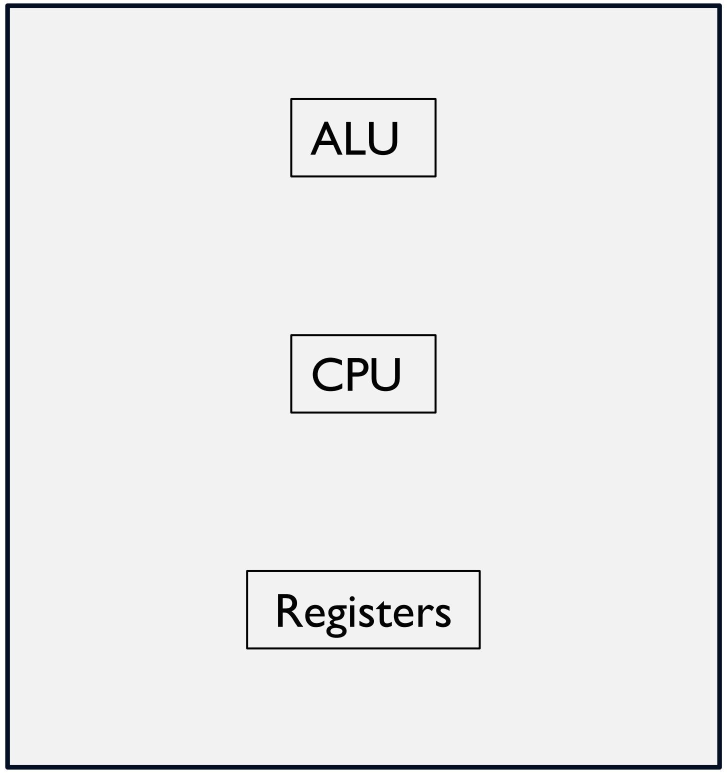
Smaller **Faster** **Cheaper** **Reliable** **Powerful**

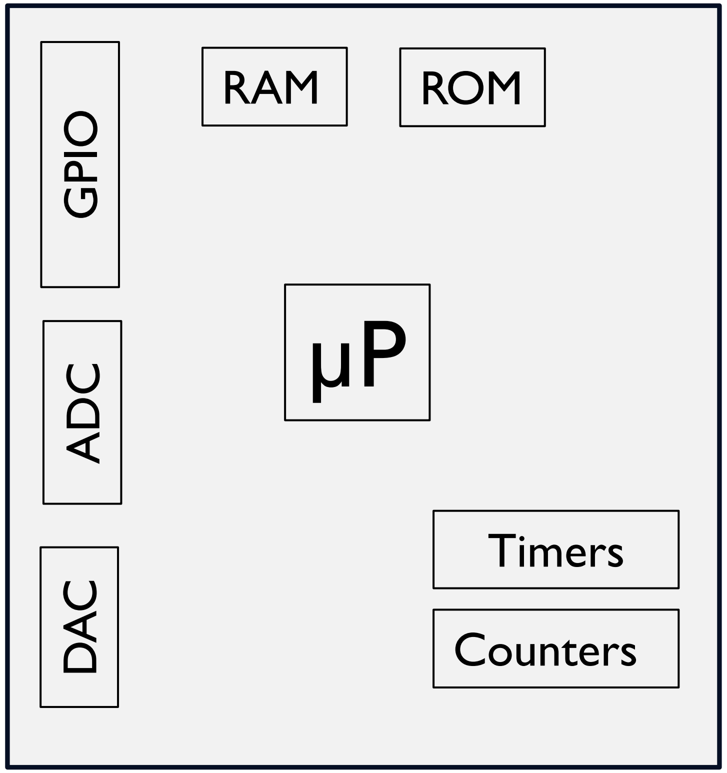
Platform SoC

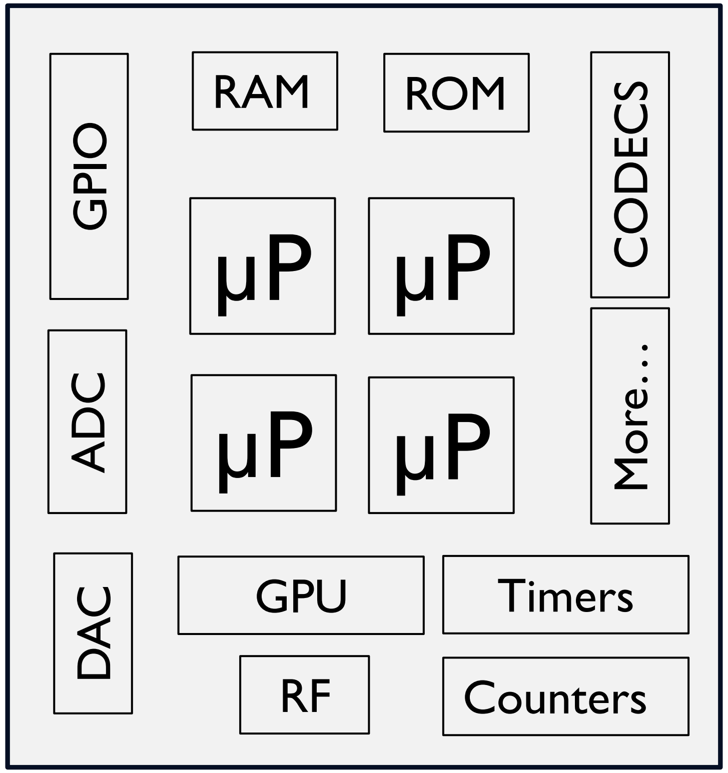


System on Chip (SoC)

Microprocessors (µP), Microcontrollers (µC), vs System on Chip (SoC)







Microprocessor:

- General purpose computing
- I/O devices and memory are interfaced externally

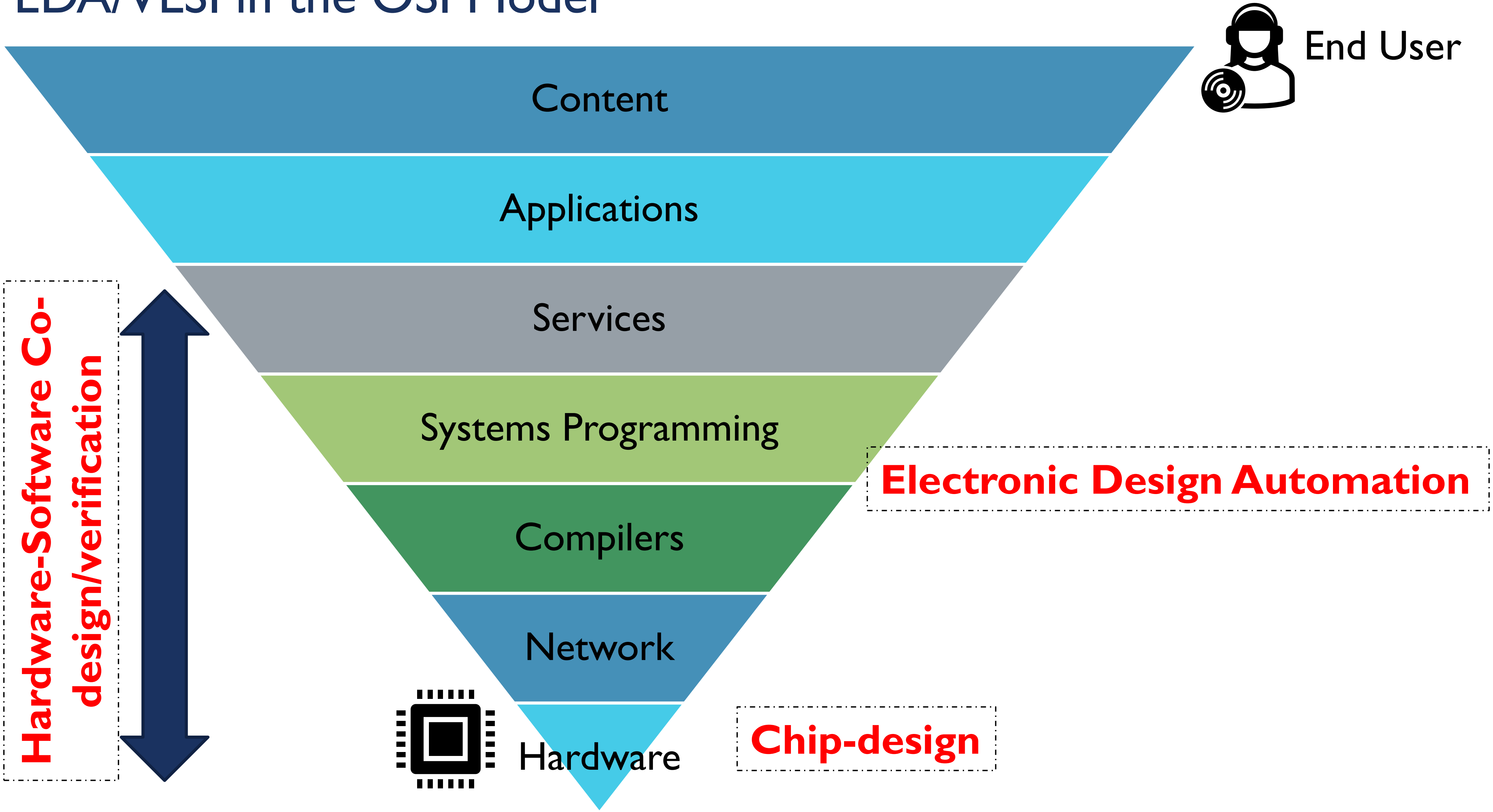
Microcontroller:

- Embedded computing
- µP + memory + I/O devices + other basic peripherals
- usually, single µP core

System on Chip (SoC):

- Complex applications
- multiple µP + memory + I/O devices + RF + GPU + ….
- Sophisticated computing system on a chip

EDA/VLSI in the OSI Model



End User

Content

Applications

Services

Systems Programming

Compilers

Network

Hardware

Hardware-Software Co-design/verification

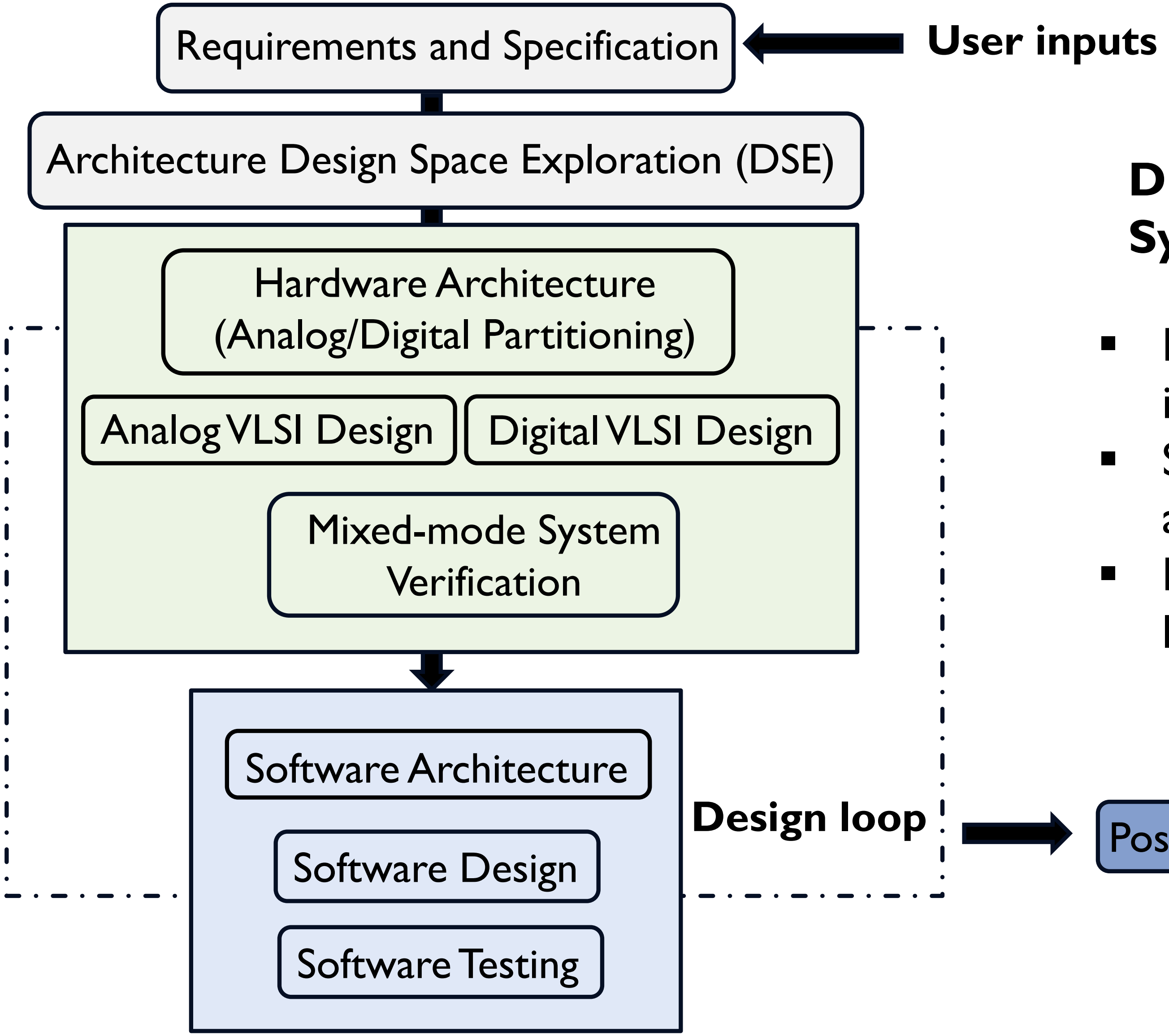
Electronic Design Automation

Chip-design

ELECTRONIC SYSTEMS DESIGN

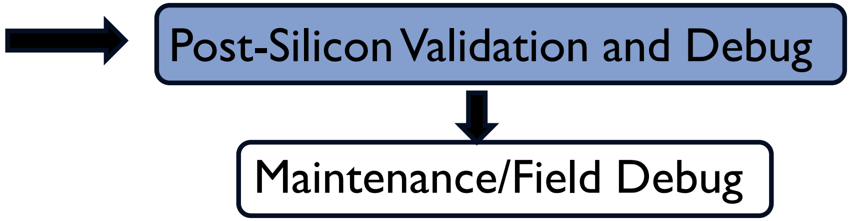
How to design Electronic Systems?

Electronic System Design – Traditional Design Methodology

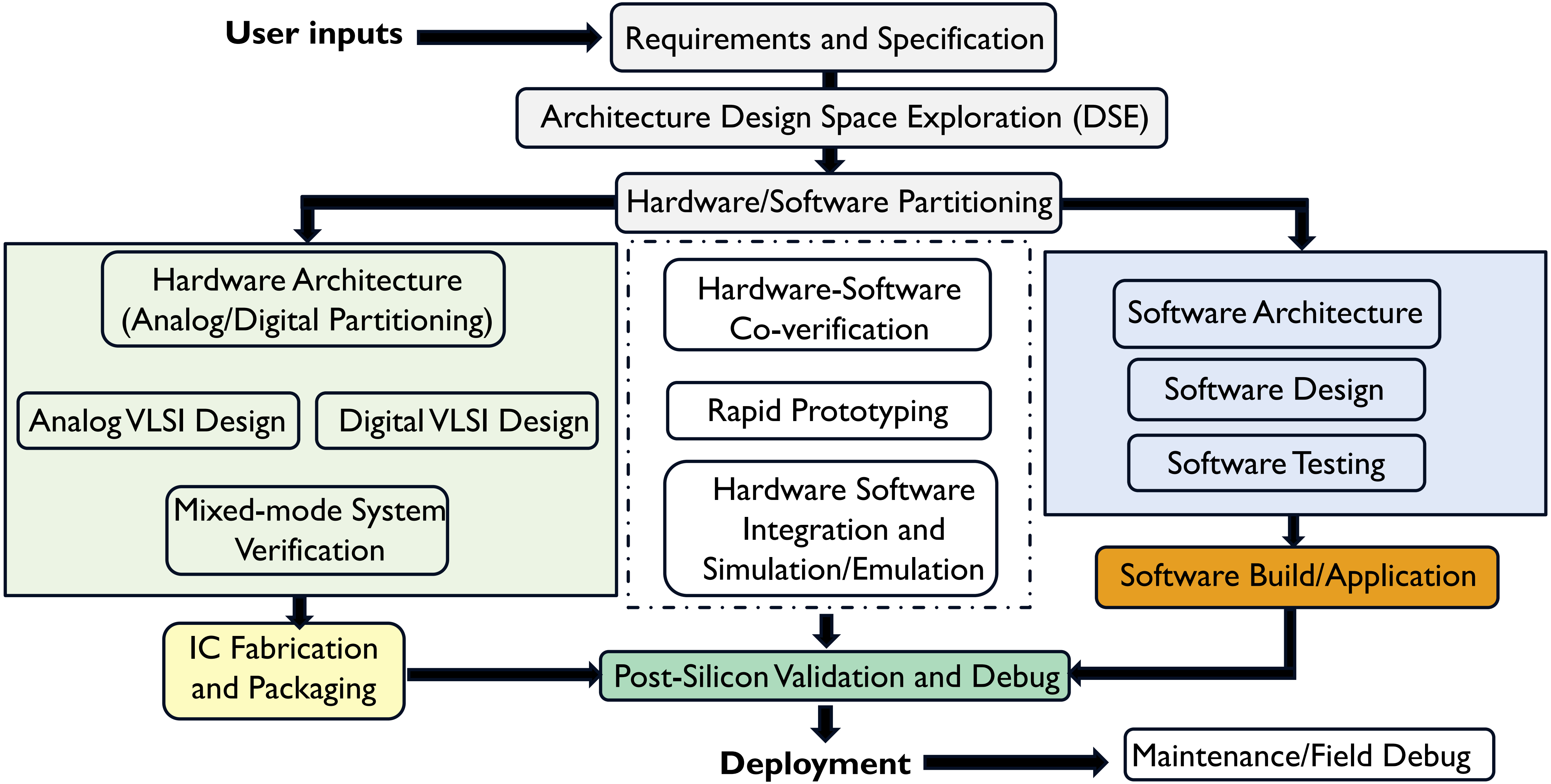

Requirements and Specification
User inputs
Architecture Design Space Exploration (DSE)
Hardware Architecture (Analog/Digital Partitioning)
Analog VLSI Design
Digital VLSI Design
Mixed-mode System Verification
Software Architecture
Software Design
Software Testing
Design loop

Drawbacks of Traditional Electronic System Design Flow:

- Hardware architecture is designed and implemented first
- Software development can only start after prototype hardware is available
- Longer TTM – hardware-software design loop

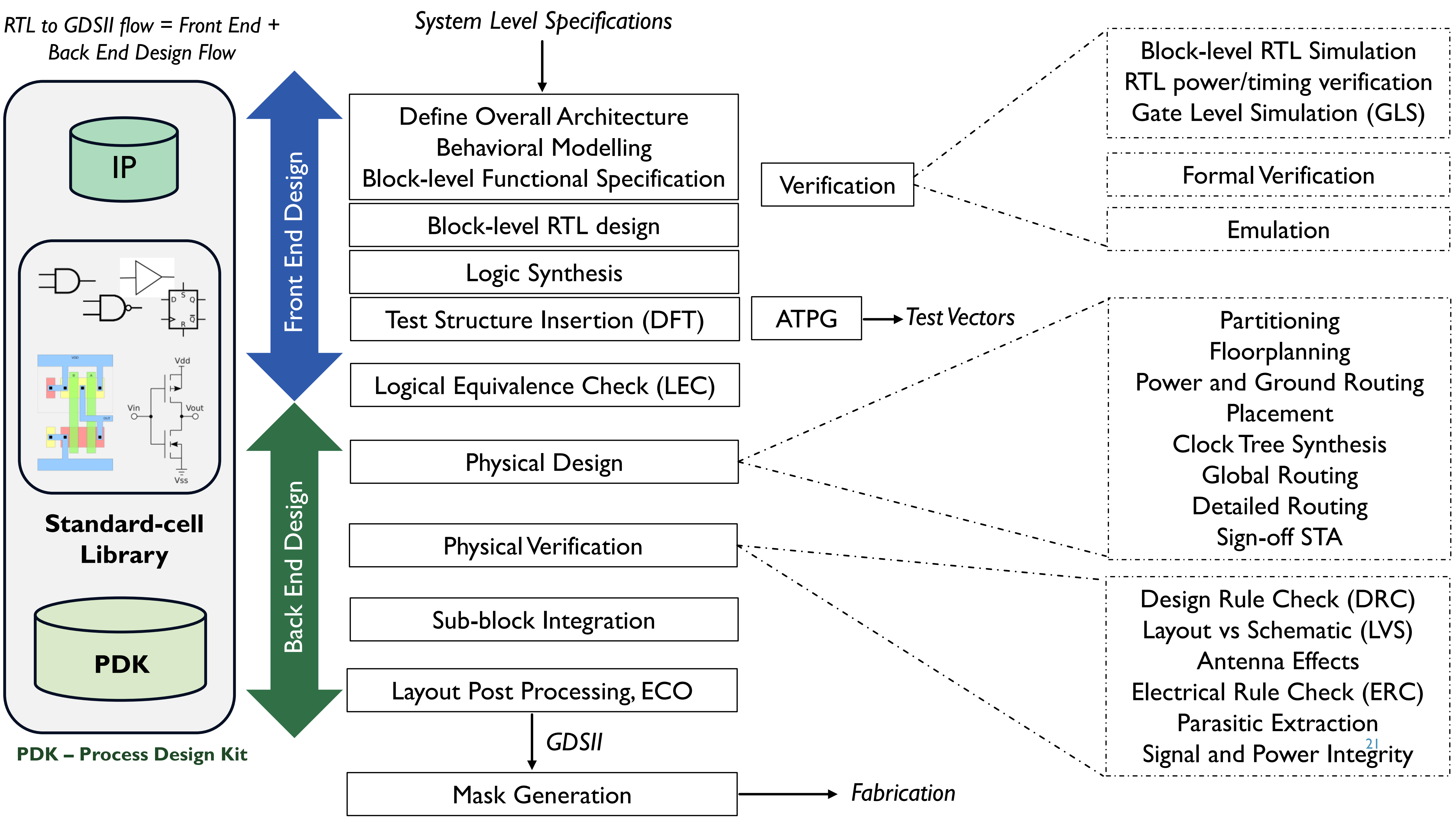


Electronic Systems Design – Hardware/Software Co-design Methodology


User inputs
Requirements and Specification
Architecture Design Space Exploration (DSE)
Hardware/Software Partitioning
Hardware Architecture (Analog/Digital Partitioning)
Analog VLSI Design
Digital VLSI Design
Mixed-mode System Verification
Hardware-Software Co-verification
Rapid Prototyping
Hardware Software Integration and Simulation/Emulation
Software Architecture
Software Design
Software Testing
Software Build/Application
IC Fabrication and Packaging
Post-Silicon Validation and Debug
Deployment
Maintenance/Field Debug

Standard Cell-Based ASIC Design

RTL to GDSII flow = Front End + Back End Design Flow



IP

Standard-cell Library

PDK

PDK – Process Design Kit

Front End Design

Back End Design

System Level Specifications

- Define Overall Architecture
  Behavioral Modelling
  Block-level Functional Specification
- Block-level RTL design
- Logic Synthesis
- Test Structure Insertion (DFT)
- Logical Equivalence Check (LEC)
- Physical Design
- Physical Verification
- Sub-block Integration
- Layout Post Processing, ECO

GDSII

- Mask Generation → *Fabrication*

Verification:

- Block-level RTL Simulation
  RTL power/timing verification
  Gate Level Simulation (GLS)
- Formal Verification
- Emulation

ATPG → *Test Vectors*

Physical Design:

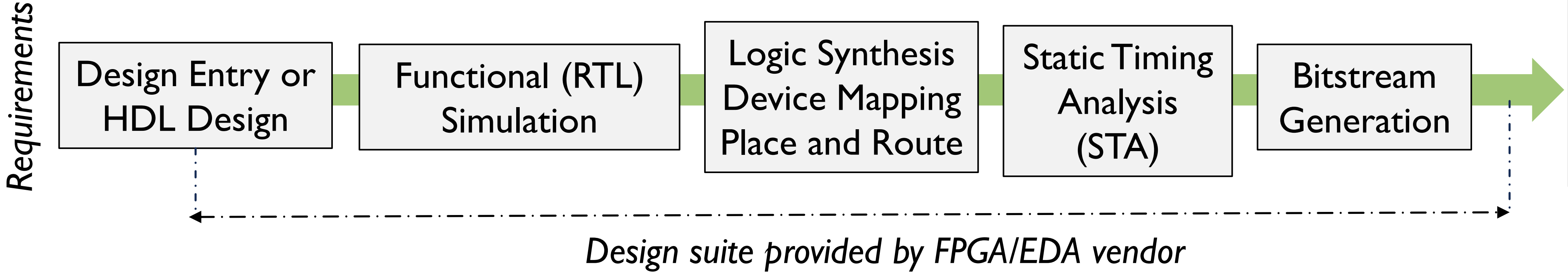
- Partitioning
- Floorplanning
- Power and Ground Routing
- Placement
- Clock Tree Synthesis
- Global Routing
- Detailed Routing
- Sign-off STA

Physical Verification:

- Design Rule Check (DRC)
- Layout vs Schematic (LVS)
- Antenna Effects
- Electrical Rule Check (ERC)
- Parasitic Extraction
- Signal and Power Integrity

FPGA based Design

- **Field Programmable Gate Array (FPGA)**
  - pre-defined 'Look Up Tables' (LUTs), hardware resources which can be programmed using a 'bit stream'
- **FPGAs vs ASICs**
  - **Design Methodology**
    - ASIC design methodology = RTL to GDSII flow (Front End + Back End Design)
    - FPGA design methodology = digital design → generate bit-stream → program FPGA device → deploy
  - **'Reprogrammable'**
    - ASIC is 'cast' in Silicon → cannot be modified unless there is a 'respin' or new design, system level changes only possible through software programming if ASIC embeds a CPU/GPU/DSP etc.
    - FPGA is 'Field Programmable' i.e., a new digital design can be remapped to hardware and redeployed on the fly. Modern FPGAs embedded with a CPU are called **'Embedded FPGA Processors'** or **'SoC FPGAs'** or **'FPGA SoCs'**
  - **Others**
    - FPGA based systems often have high production cost (this is changing overtime), mostly suited for low volume
    - ASIC performance is generally better than FPGAs
    - FPGAs are a very good prototyping vehicle or for products requiring constant 'reconfigurability'.



Requirements

Design Entry or HDL Design

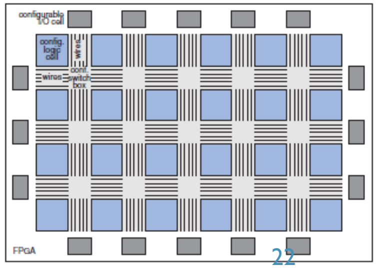
Functional (RTL) Simulation

Logic Synthesis Device Mapping Place and Route

Static Timing Analysis (STA)

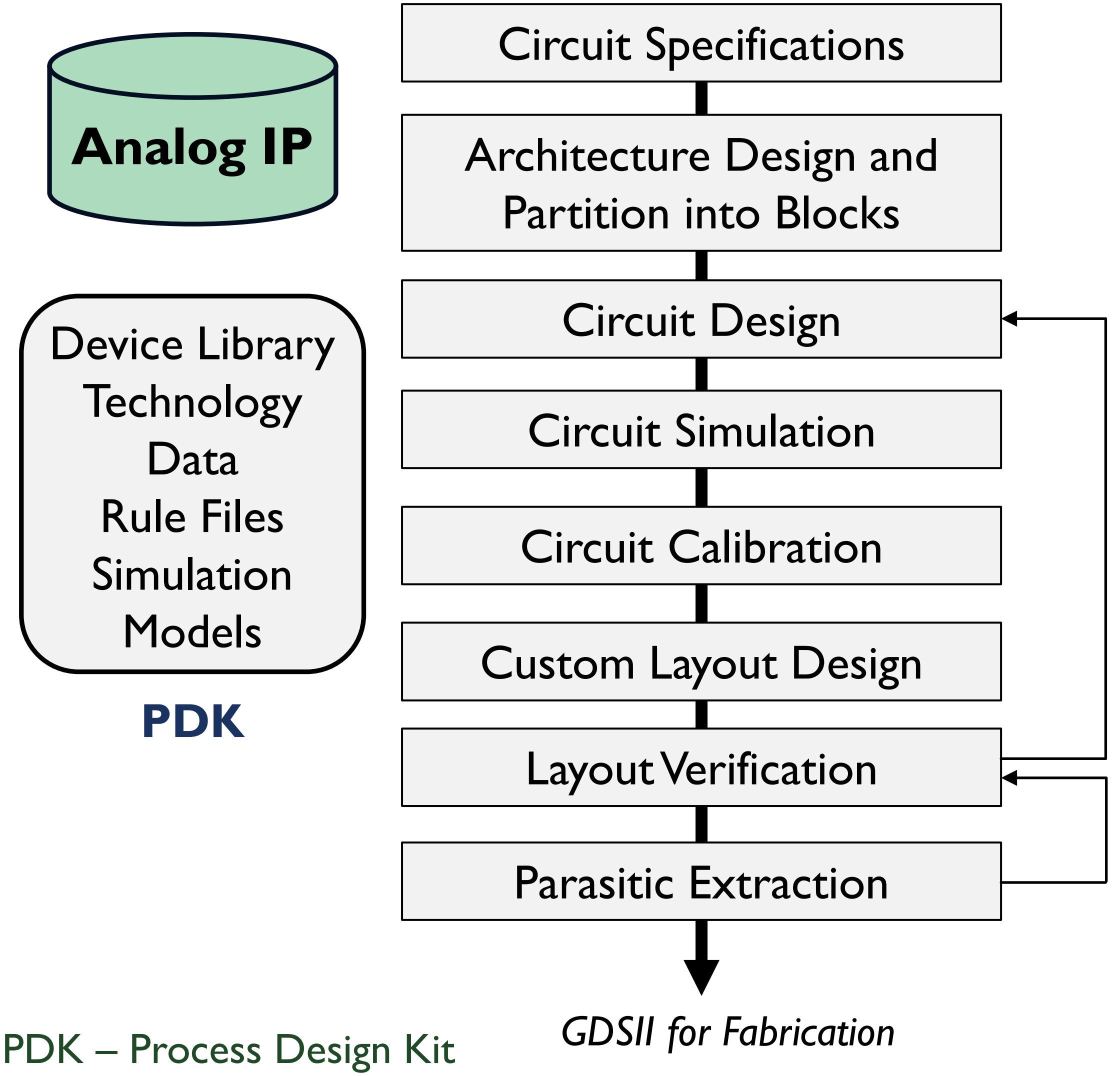
Bitstream Generation

Design suite provided by FPGA/EDA vendor



FPGA programming

Analog VLSI Design



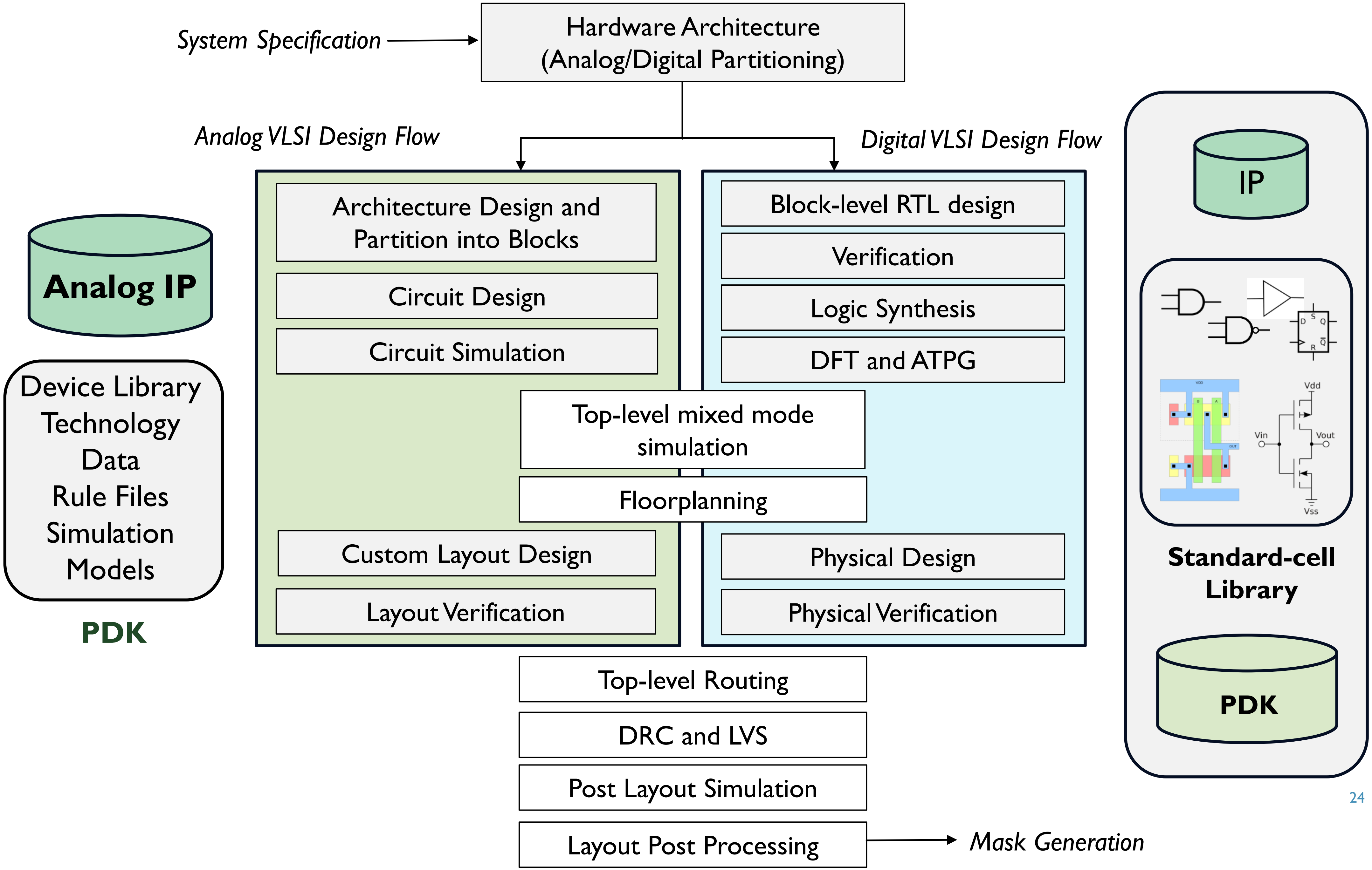
PDK – Process Design Kit

Analog IC Design:

- Analog circuits interface and manipulate continuous time arbitrary electrical signals
- Analog circuit design is *considered a* **'Black Art'**
- Analog design automation still today mostly remains an 'elusive research topic'
- Analog circuits WILL always remain relevant and can play a significant role in the AI chip revolution

"Analog Circuit designers are scarce, VERY WELL compensated, gain lots of respect, and often regarded as Artists" - a well known adage in the VLSI industry!

Mixed-Signal Design



System Specification

Hardware Architecture (Analog/Digital Partitioning)

Analog VLSI Design Flow

Digital VLSI Design Flow

Analog IP

Device Library
Technology
Data
Rule Files
Simulation
Models

PDK

Architecture Design and Partition into Blocks

Circuit Design

Circuit Simulation

Custom Layout Design

Layout Verification

Block-level RTL design

Verification

Logic Synthesis

DFT and ATPG

Physical Design

Physical Verification

Top-level mixed mode simulation

Floorplanning

Top-level Routing

DRC and LVS

Post Layout Simulation

Layout Post Processing

Mask Generation

IP

Standard-cell Library

PDK

EXECUTING DESIGN METHODOLOGIES USING EDA TOOLS

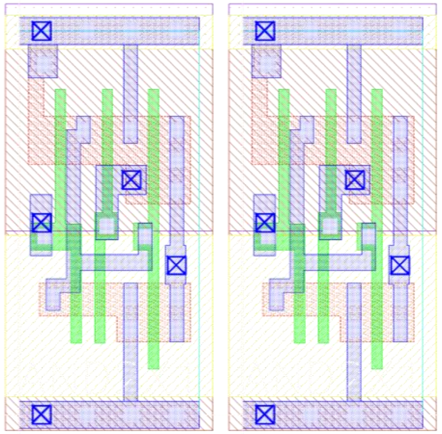
EDA Methodology in Practice

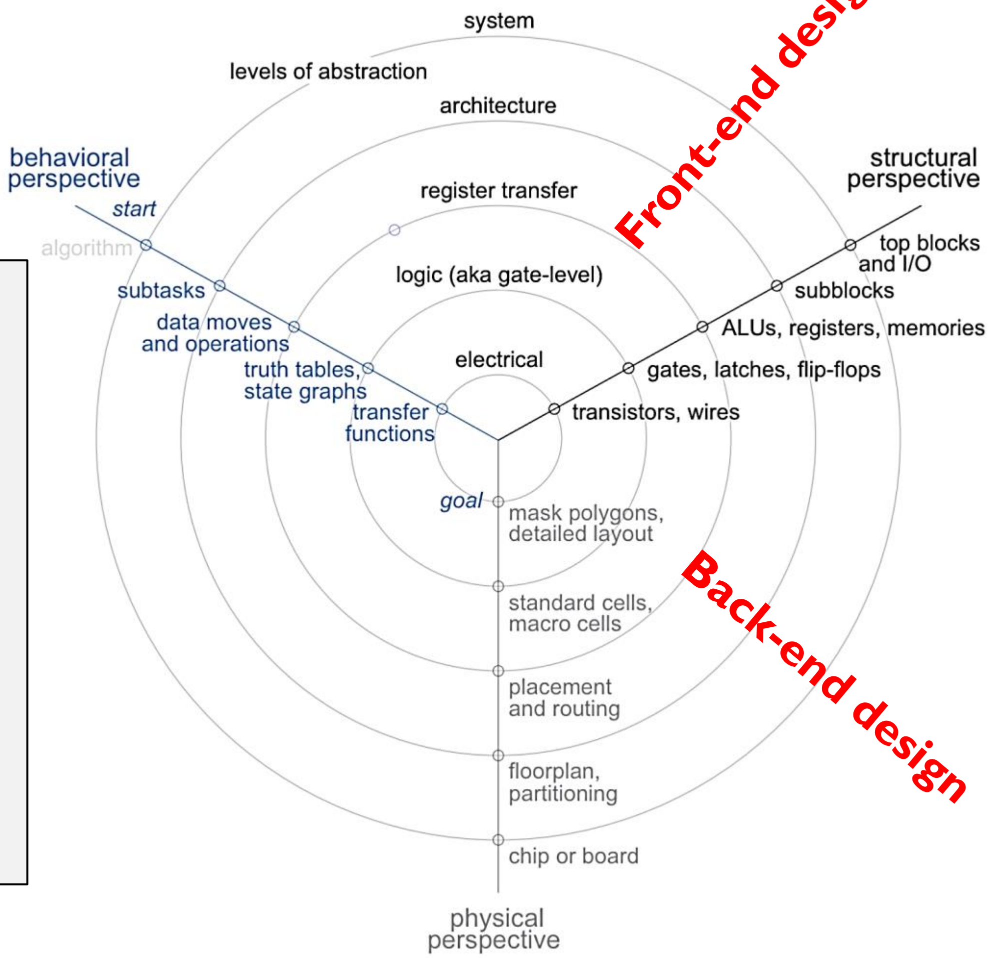
Design Methodologies – ‘Top-Down Approach’

```
module dff (CK,Q,D);
input CK,D;
output Q;
reg Q;
always @ (posedge CK)
  Q <= D;
endmodule

module
And_DFF_Or(in1,in2,in3,CK,out1);
input in1,in2,in3,CK;
output out1;
wire w1,w2;
and AND1(w1,in1,in2);
dff DFF1(CK,w2,w1);
or OR1(out1,w2,in3);
endmodule
```

```
module And_DFF_Or(in1, in2, in3, CK,
out1, SCAN_EN, SCAN_IN, SCAN_OUT);
  input in1, in2, in3, CK, SCAN_EN,
SCAN_IN;
  output out1, SCAN_OUT;
  wire in1, in2, in3, CK, SCAN_EN,
SCAN_IN;
  wire out1, SCAN_OUT;
  wire n_0;
  OR2X1 g11__2398(.A (in3), .B
(SCAN_OUT), .Y (out1));
  SDFFQX1 DFF1_Q_reg(.CK (CK), .D
(n_0), .SI (SCAN_IN), .SE (SCAN_EN),
        .Q (SCAN_OUT));
  AND2X1 g13__5107(.A (in1), .B (in2),
.Y (n_0));
endmodule
```



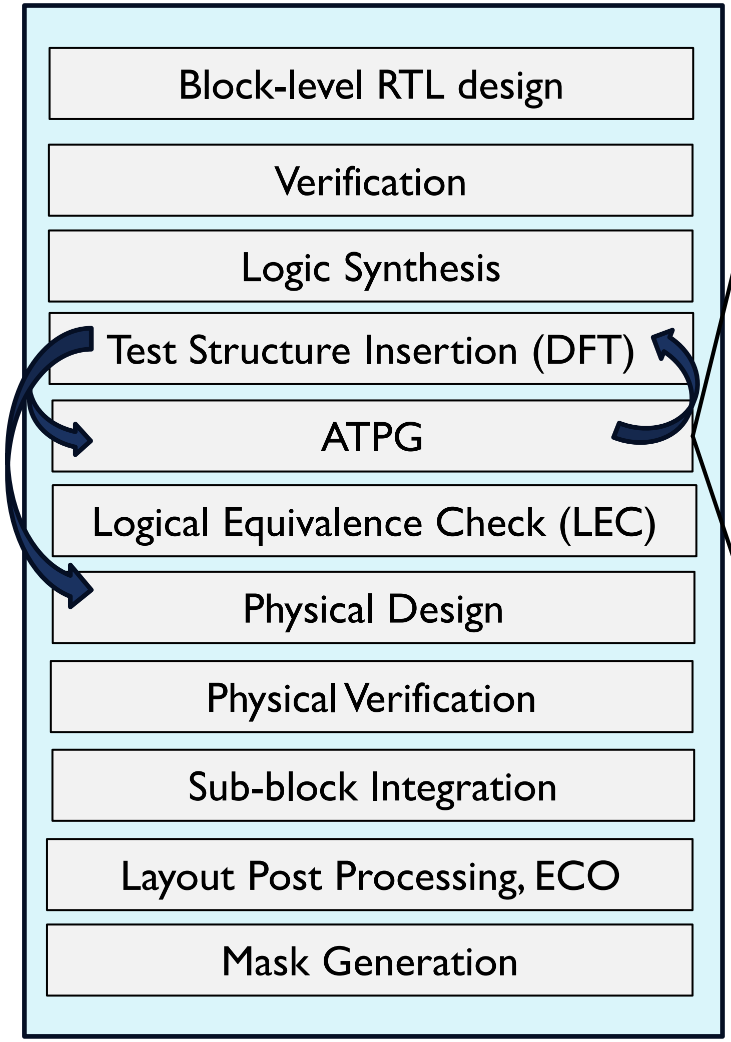


Daniel-Gajski Y Chart

Reference – Top-Down VLSI Design, Figure 1.10

‘Design Methodology’ and ‘Tool Flow’ Example

DFT methodology influences the **ATPG** methodology or the other way round. It also impacts the **Physical Design** flow.



Overall Design Goals:
Achieve high (99%) stuck-at and 85% transition fault coverage.

Test Methodology:

- DC pin test parametric test
- Test logic verification
- DC logic stuck-at
- At-speed logic delay
- Memory testing
- Iddq testing
- Specialty testing
  - Analog
- Burn-in testing

‘Tool Flow’ helps execute a ‘Test Methodology’ to achieve the overall ‘Design Goal’.

ATPG Tool Flow

1. Setup tool environment
2. Read design netlist
3. Read standard-cell Verilog
4. Build ATPG design model
5. Setup constraints
6. Build chip testmodes
7. DRC checks
8. Enumerate list of faults
9. Generate scan-chain tests
10. Generate stuck-at ATPG patterns, simulate to calculate FC
11. Generate two-cycle transition ATPG patterns, simulate to calculate FC
12. Compact and write out Vector set in required format

SUMMARY

Future Developments in 'Design Methodologies'

Design methodologies will continue to address the following challenges:

- **Hardware-software co-design and verification**
  - SoC's increasingly involve multiple IPs (often from multiple vendors) connected by a *bus architecture*
  - Verification complexity increases proportionately with design complexity

- **Design for Testability**
  - dealing with newer defect mechanisms in advanced nodes, low pin count testing and cost of testing

- **Design Integrity**
  - cross talk, IR drop, timing closure, Electromigration (EM), Electrostatic Discharge (ESD)

- **3DIC Integration and Packaging**
  - Integrating heterogenous cores (digital, analog, memory etc.) and 3D IC integration

- **Power, Thermal and Timing**
  - Timing closure – ensuring design meets the timing checks and requirements
  - Higher power = hotter chips!

Summary

- Electronic systems form the backbone of the modern digital economy
- Multiple players – EDA/IP, fabless, foundry, IDM's etc. make up the semiconductor value chain and ecosystem
- A **System on Chip (SoC)** - integrates a complete computing system on a single chip
- Today's electronic systems are mostly developed using 'Hardware-Software Co-design Methodology'
- **Design methodology -** sequence of steps needed to convert a 'concept' into 'working silicon'
- Successful design methodology development is a multi-disciplinary endeavor
  - Hardware architecture and systems engineering
  - Circuit Design (digital and analog)
  - Verification technologies
  - Test Engineering
  - Fabrication, physical layout design and semiconductor physics
  - Expertise in use of advanced EDA tools, tool scripting etc.
  - Optimum use of compute resources
  - Design Data Management

> *"There is no single right way to design a chip, but sure there are many wrong ways."*
> *-Anonymous*

References

- Kaeslin, Hubert. Top-down digital VLSI design: from architectures to gate-level circuits and FPGAs. Morgan Kaufmann, 2014.
- Dillinger, Thomas. VLSI Design Methodology Development. Prentice Hall, 2019.