LSI Design Flow

BG001

Tsuneo Funabashi

(Hai Pham reviewed and modified)

Renesas Electronics Corporation

Rev. 5.4

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

Preface

1. Intended audience

H/W and S/W engineers who have studied BF001

2. Purpose of this course

* to understand how to design LSIs
(from front end to back end)

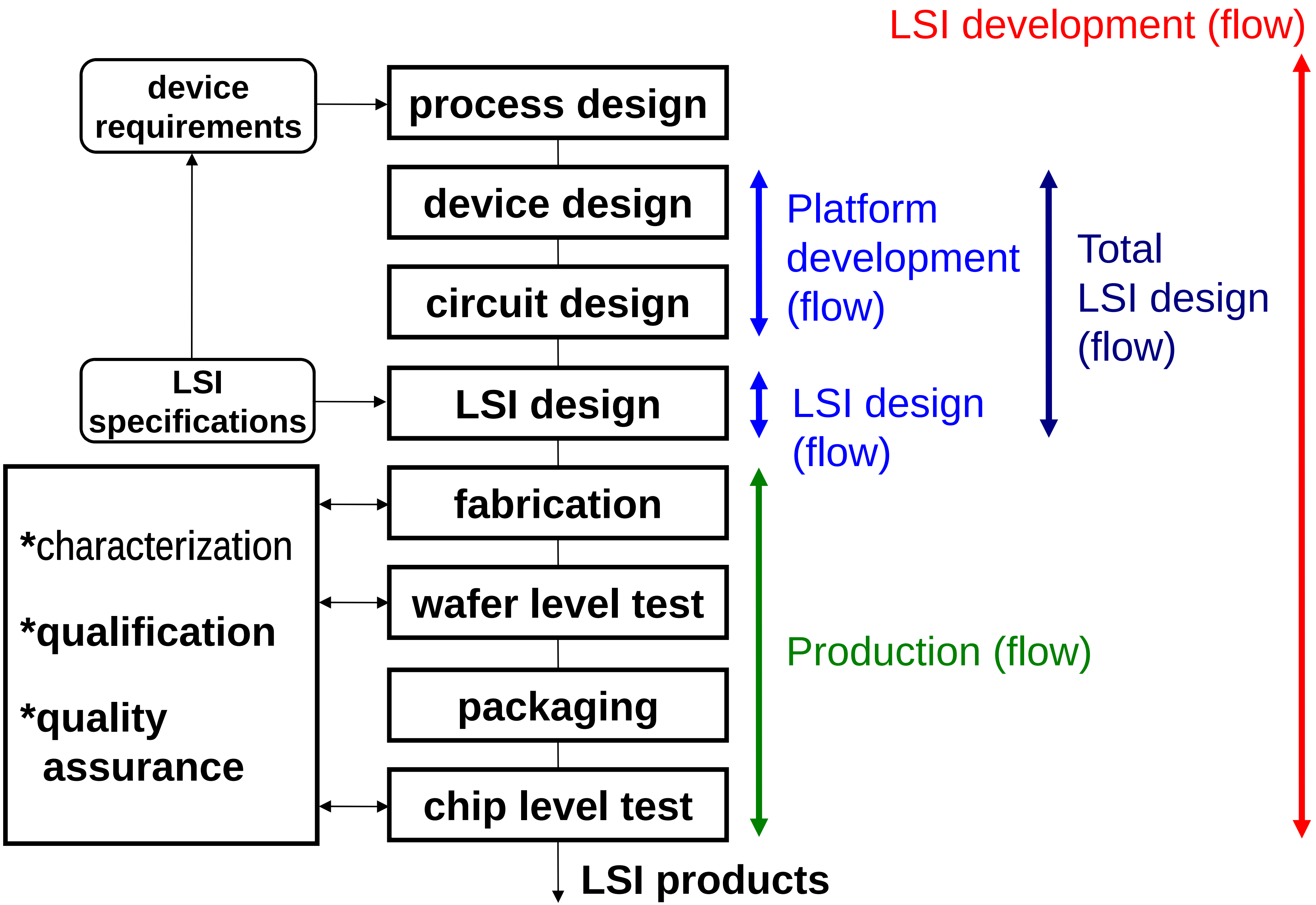
* to fill knowledge gap between automated designs
and actual device behavior

3. Recommendations

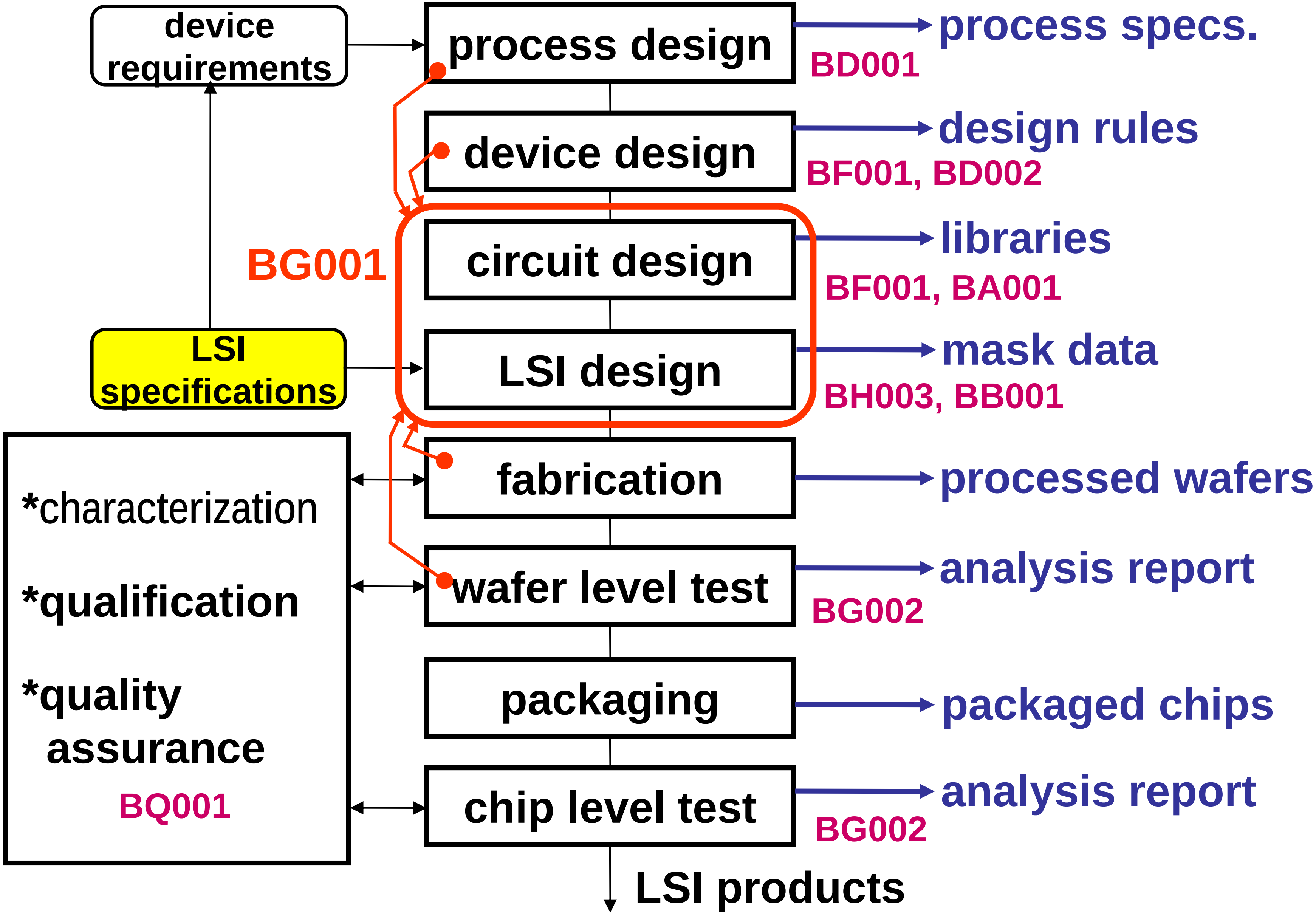
* to memorize every info in this course is not necessary

* but to recall it when you understand other groups' activities

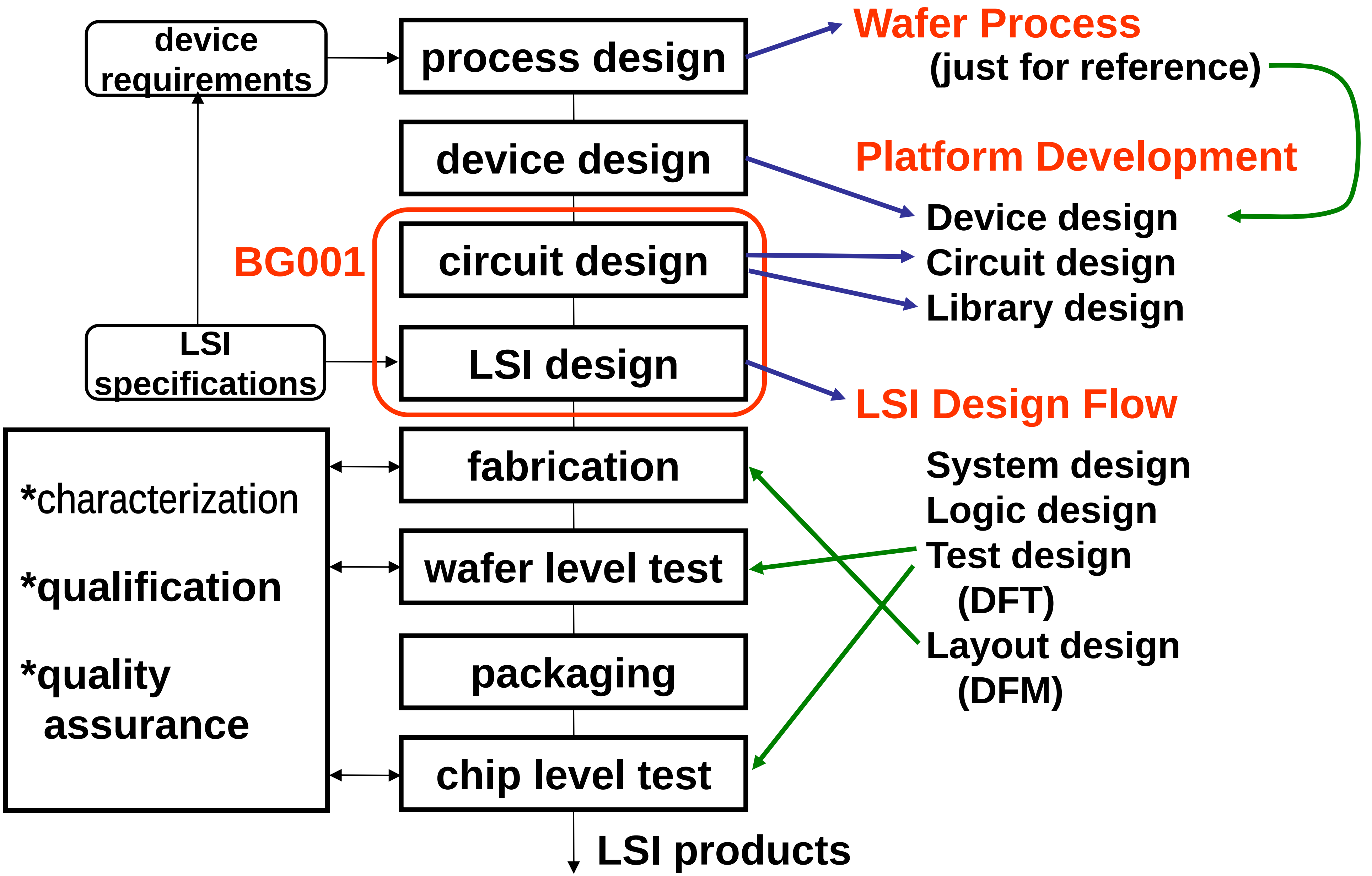
Terminology



Training Courses in LSI Development



Formation of BG001 in LSI Development



device requirements

process design

device design

BG001

circuit design

LSI specifications

LSI design

*characterization

*qualification

*quality assurance

fabrication

wafer level test

packaging

chip level test

LSI products

Wafer Process

(just for reference)

Platform Development

Device design

Circuit design

Library design

LSI Design Flow

System design

Logic design

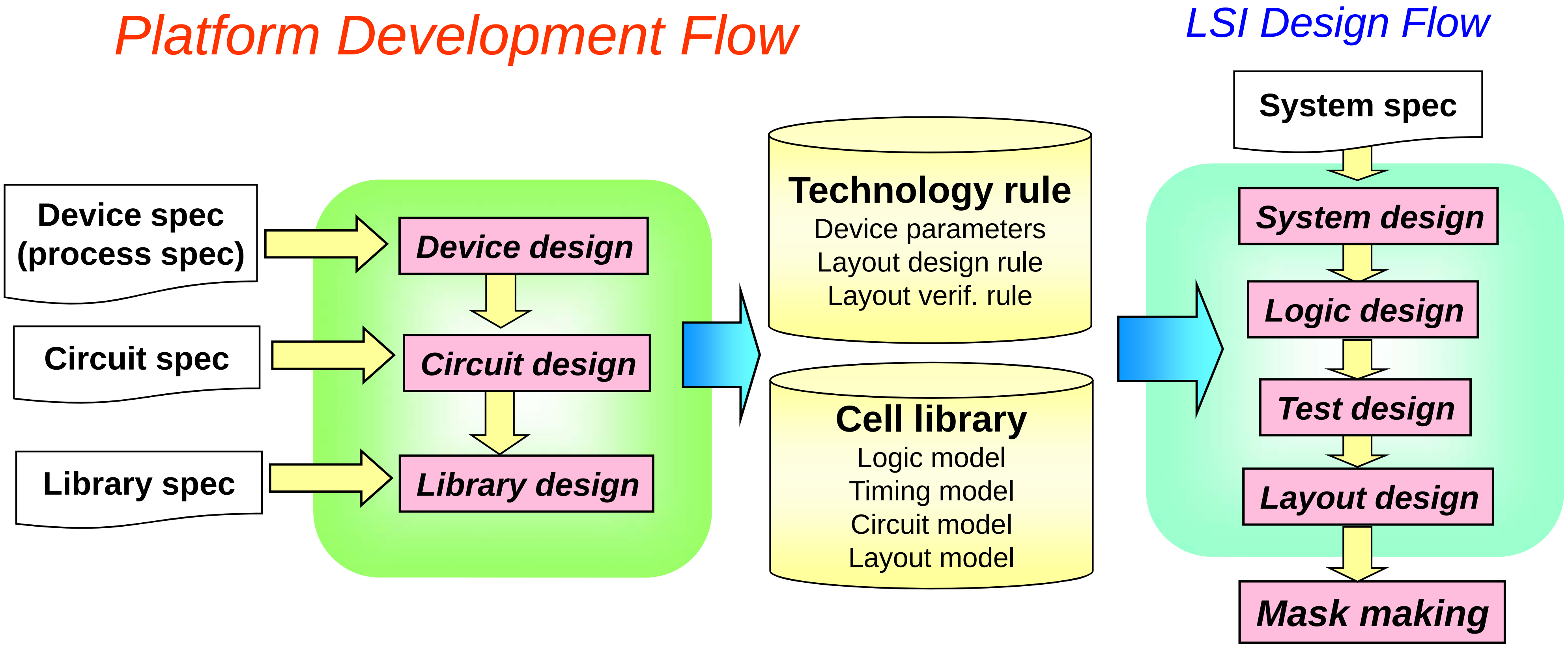
Test design (DFT)

Layout design (DFM)

Total LSI Design Flow

To design LSIs using design environment established by platform development

- Quality of the platform determines quality of all products



Product Categories

GPSP: General Purpose Standard Product

MCU: Micro Controlling Unit (CPU + Peripherals + Memory)

MPU: Micro Processing Unit (High-performance CPU)

Discretes

Standard IC

Memory

ASSP: Application Specific Standard Product

(see next slide)

ASCP: Application Specific Custom Product

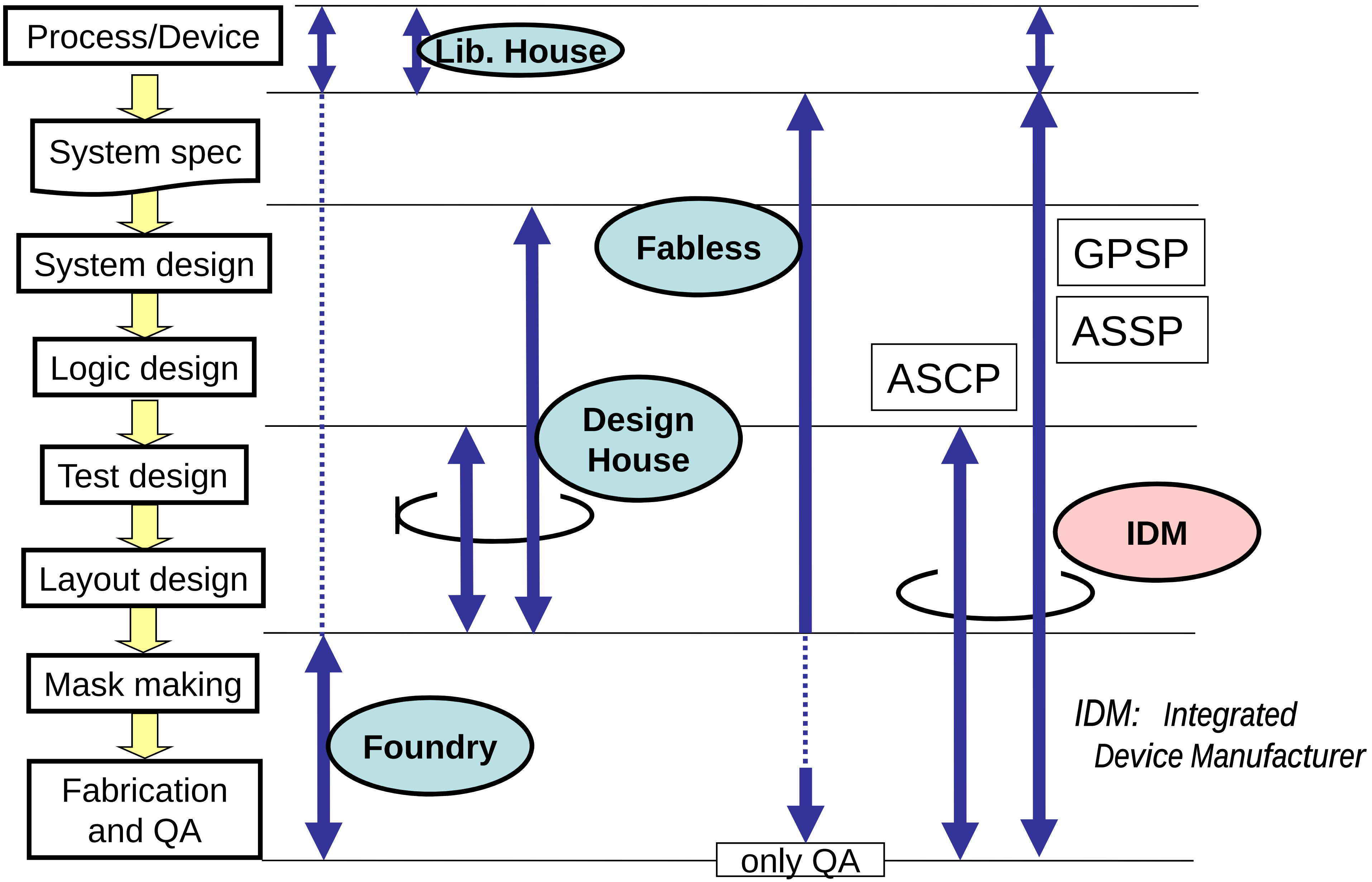
ASIC: Application Specific Integrated Circuit, or

SoC: System on Chip

SiP: System in Package

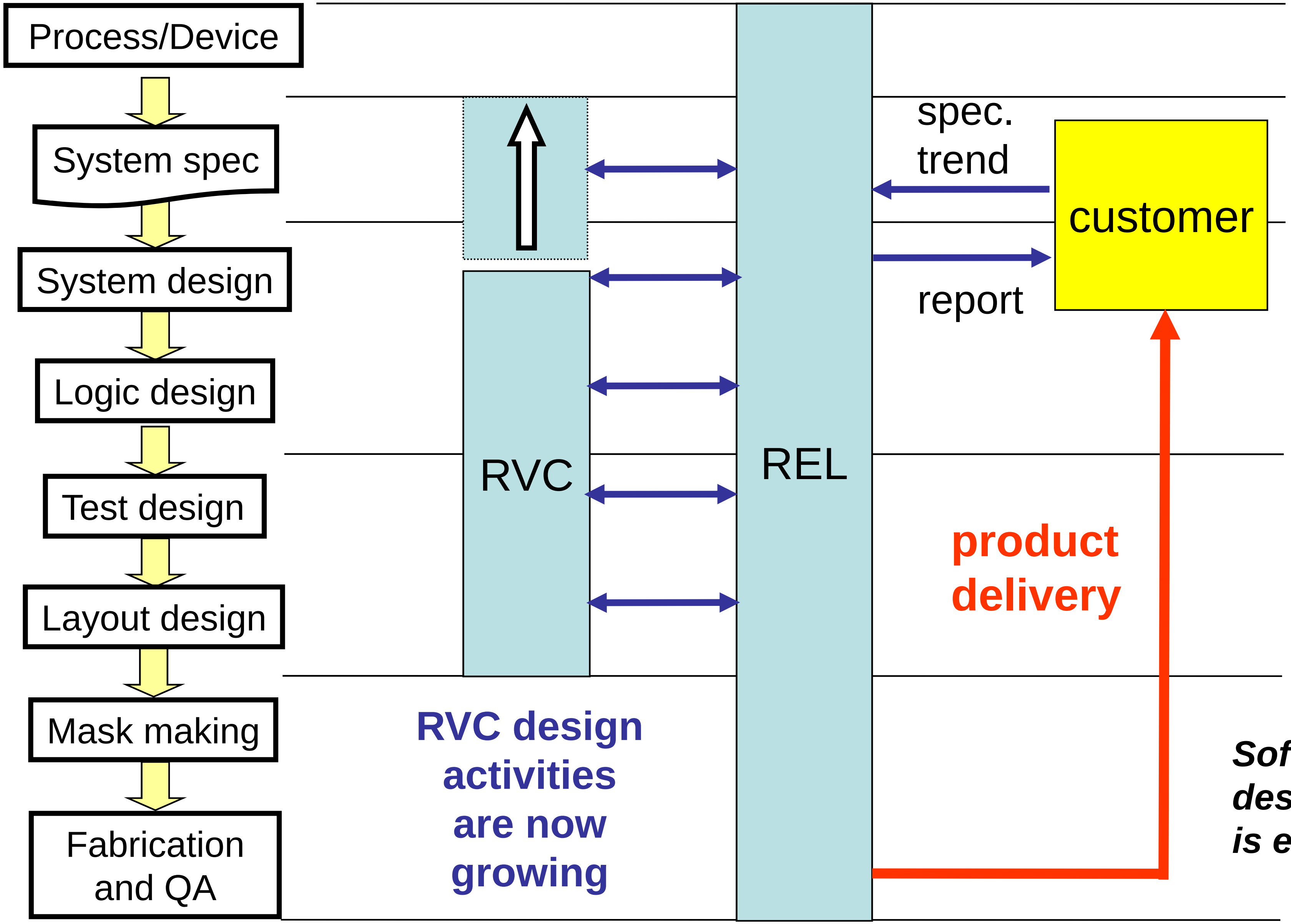
CPU: Central Processing Unit

LSI Development and Business Models



IDM: Integrated Device Manufacturer

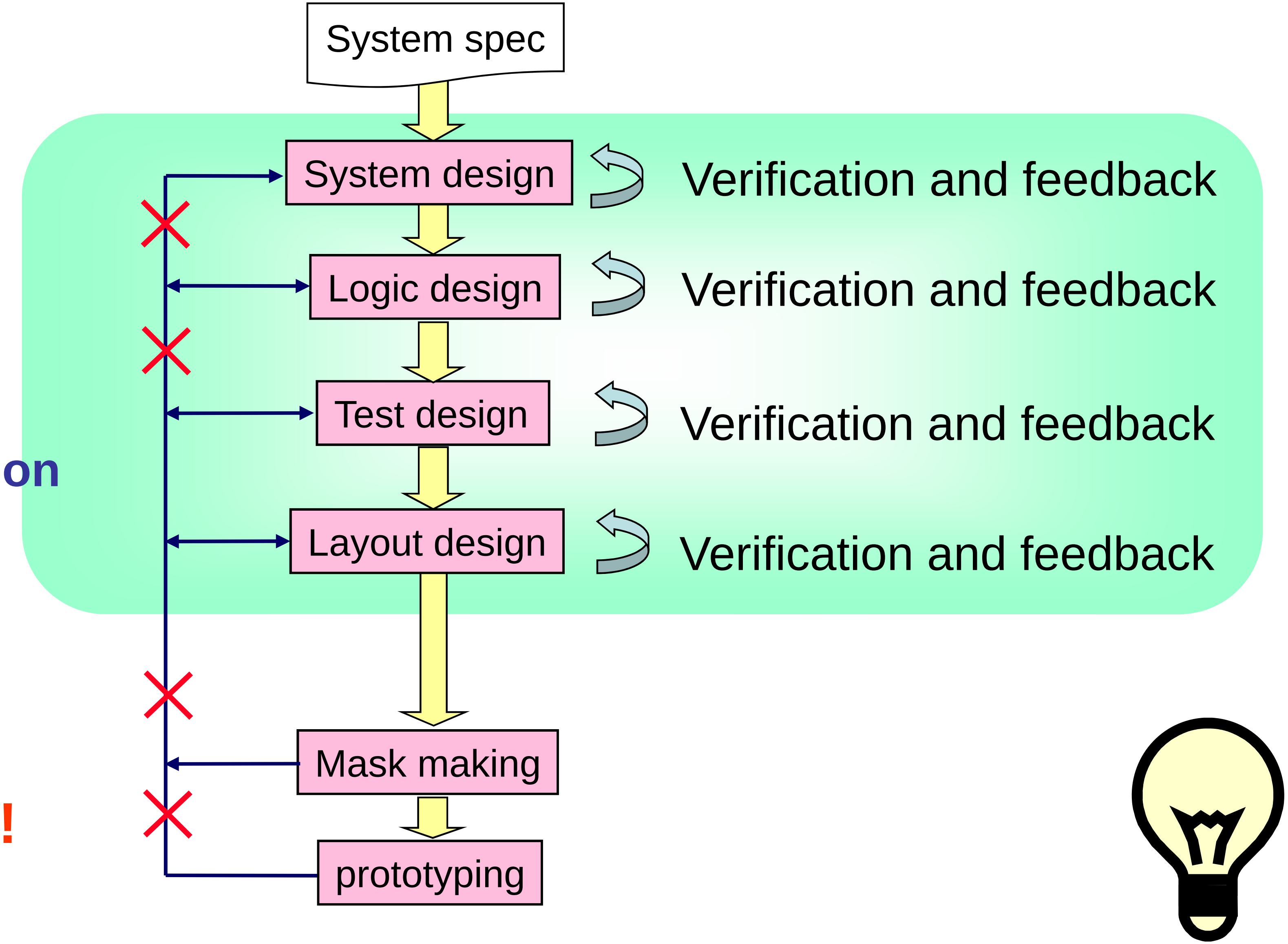
LSI Development in Renesas Group



RVC design activities are now growing

Software design flow is excluded

Self-Organized Design Steps



System spec
System design
Verification and feedback
Logic design
Verification and feedback
Test design
Verification and feedback
Layout design
Verification and feedback
Mask making
prototyping

Minimize retrogression

No spin!

Design Input and Design Output

- To minimize retrogression, Input and Output of each design step must be clearly defined: key in ISO9001
- Input and Output are design interfaces, and must be well documented
- If Input is unclear, confirm with person in charge of upper stream design step
- Before releasing Output verify it and investigate it by design review
- Modification or addition to Input must be clearly stated in Output with reason for existence
- Modifications to existing design and newly added design must be clearly designated in Output with reason for existence

Design Output

= documentation first + design database later

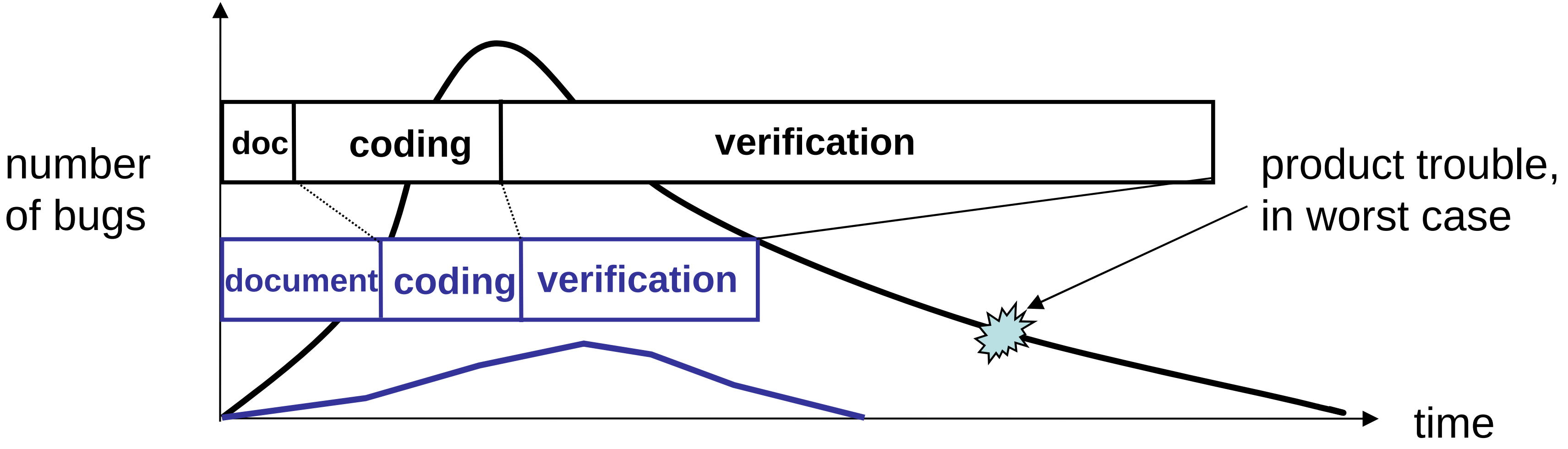
- **becomes Input of succeeding design step**
- **determines design quality**
- **becomes fundamental document for future product maintenance**

<example of system design>

Start coding for the moment, and create document later in my own time: **putting the cart before the horse**.

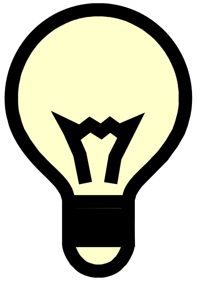
Sometimes, system designers are also logic designers.

But close system design by outputting system specification as Output.



Well Organized Document

- To be fully conscious of readers: like instructing them
- To make design interfaces clear: design input, and design output
- To make 5W1H clear (who, what, when, where, why, how)
- To make document systematic, and avoid distributed description by organizing your thoughts
- To avoid ambiguous expression: no room for arbitrary interpretation by readers
- Not to abuse alteration, and to make revision history and distribution history clear



Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

Photolithography and Mask Set (1)

<simplified in an easy-to-understand way>

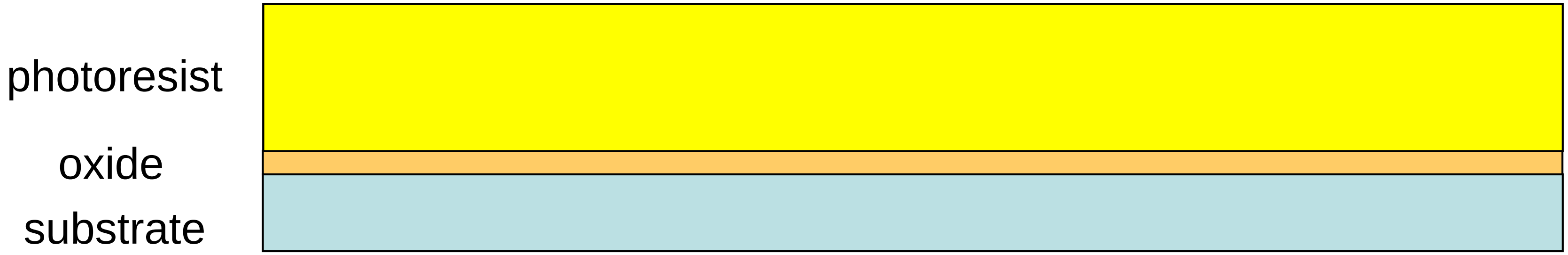
Photolithography and Mask Set (1)

<simplified in an easy-to-understand way>

oxide
substrate

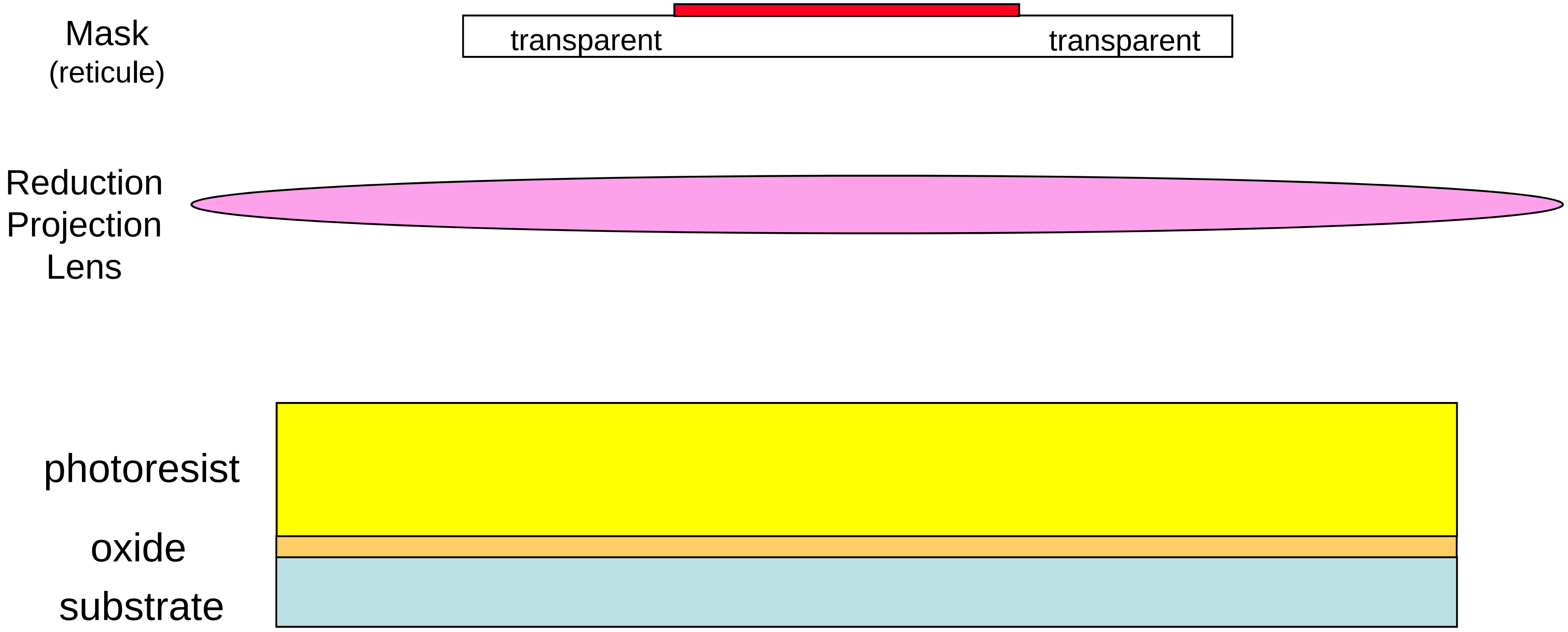
Photolithography and Mask Set (1)

<simplified in an easy-to-understand way>



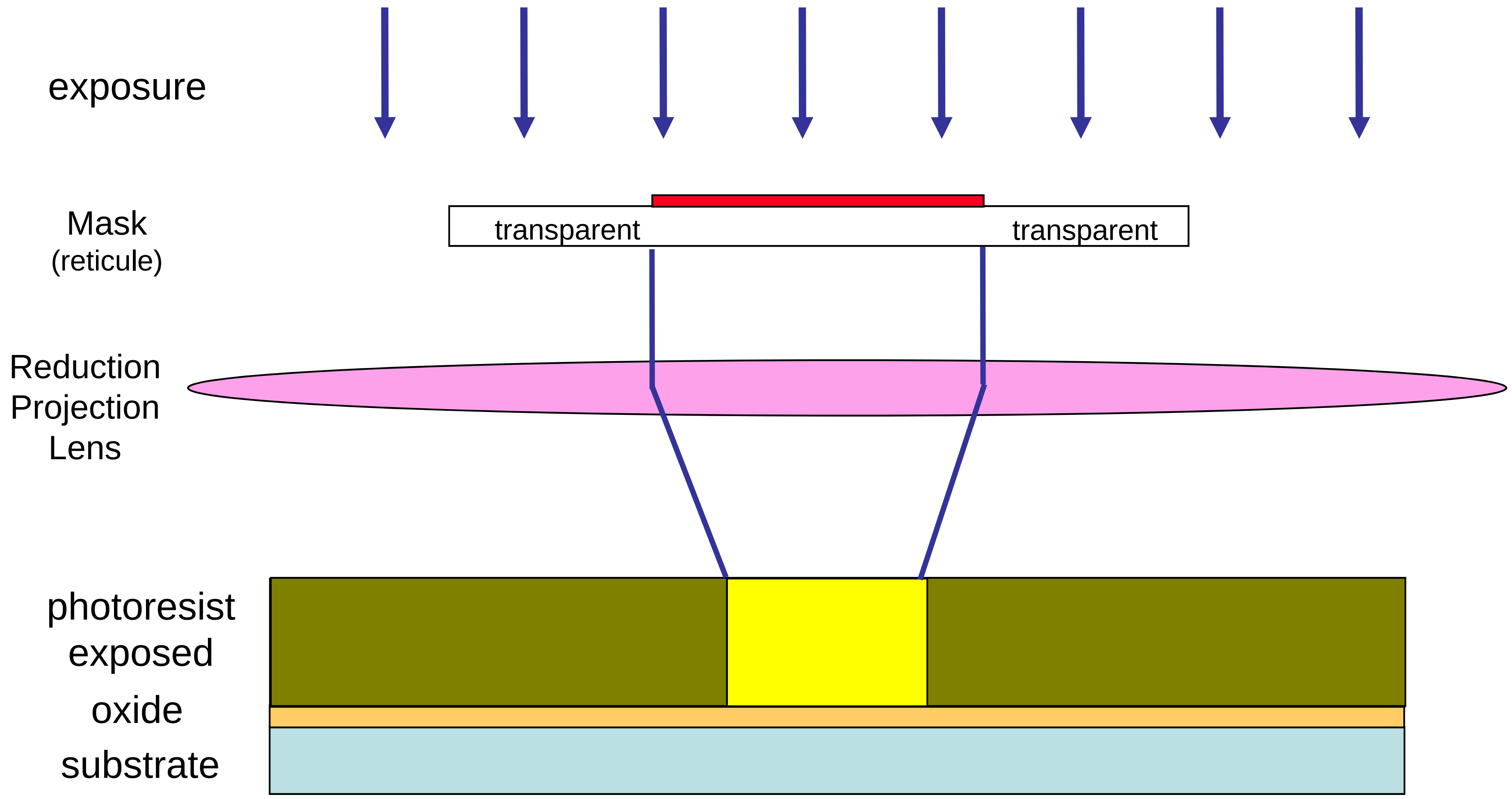
Photolithography and Mask Set (1)

<simplified in an easy-to-understand way>



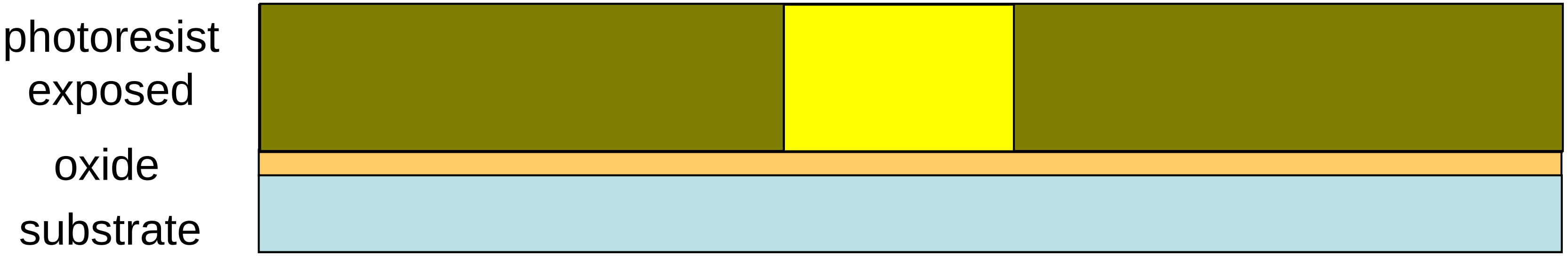
Photolithography and Mask Set (1)

<simplified in an easy-to-understand way>



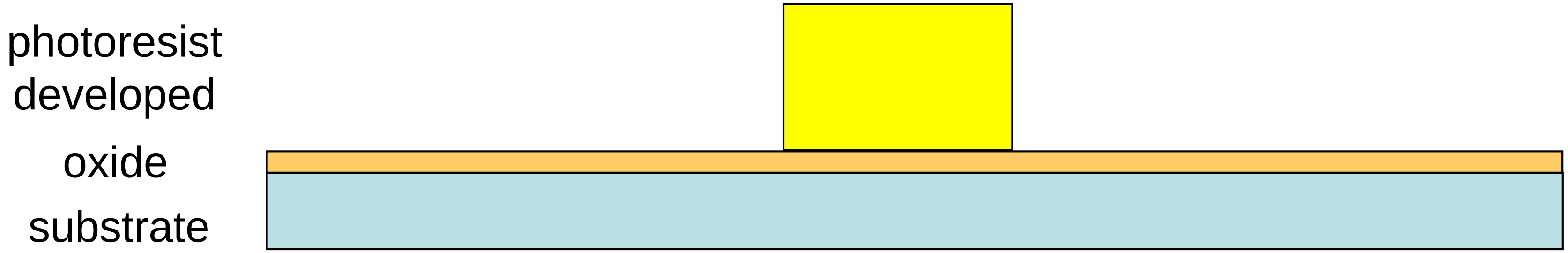
Photolithography and Mask Set (1)

<simplified in an easy-to-understand way>



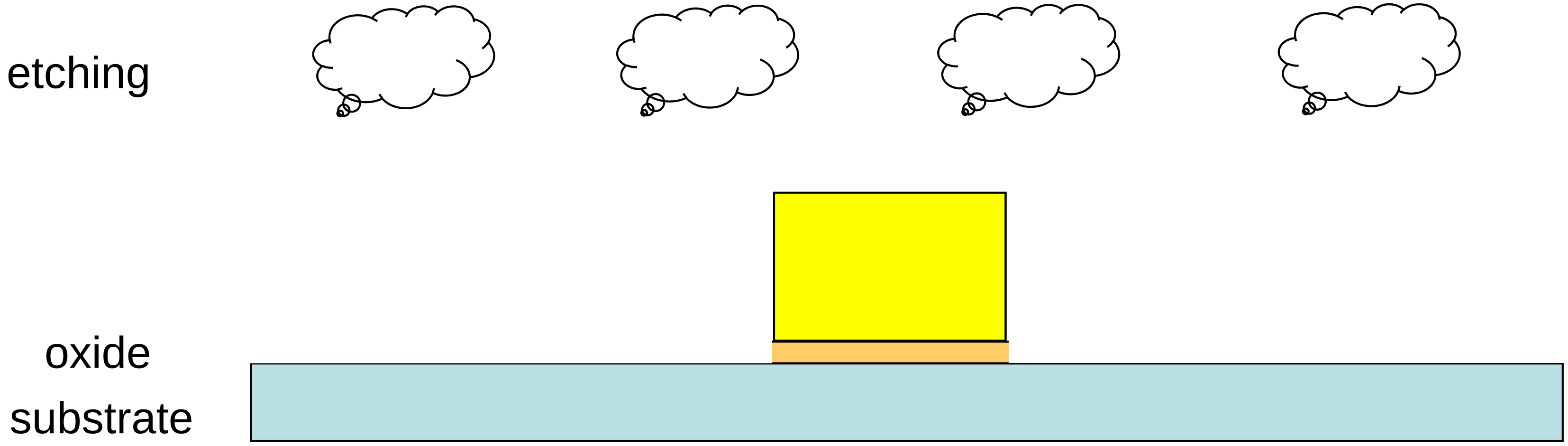
Photolithography and Mask Set (2)

<simplified in an easy-to-understand way >



Photolithography and Mask Set (2)

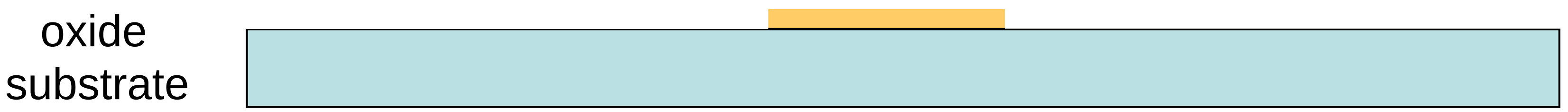
<simplified in an easy-to-understand way >



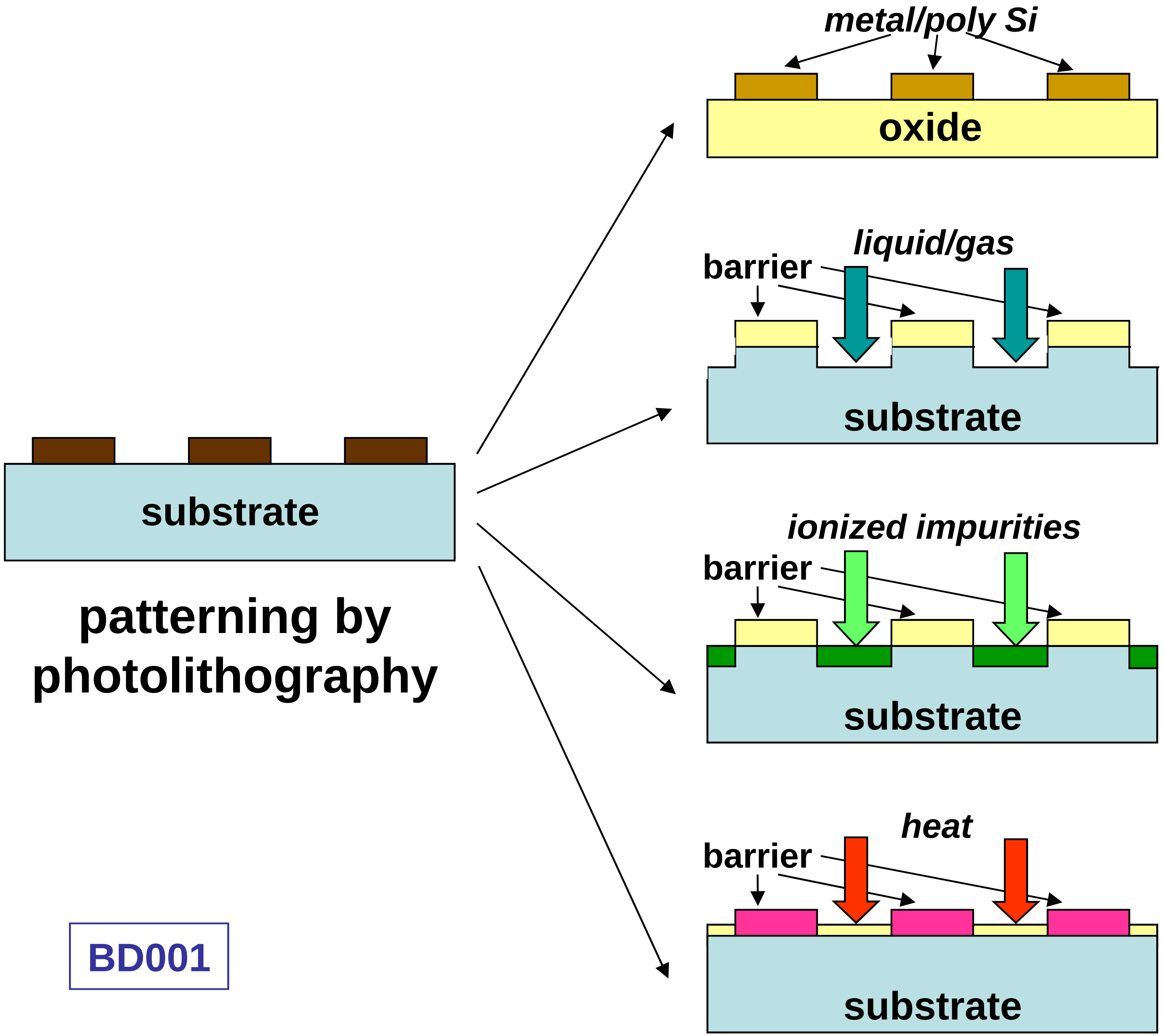
Photolithography and Mask Set (2)

<simplified in an easy-to-understand way >

Photolithography: these steps are used to shape some specific patterns on silicon substrate surface by using mask set



Process Steps



patterning by photolithography

wiring

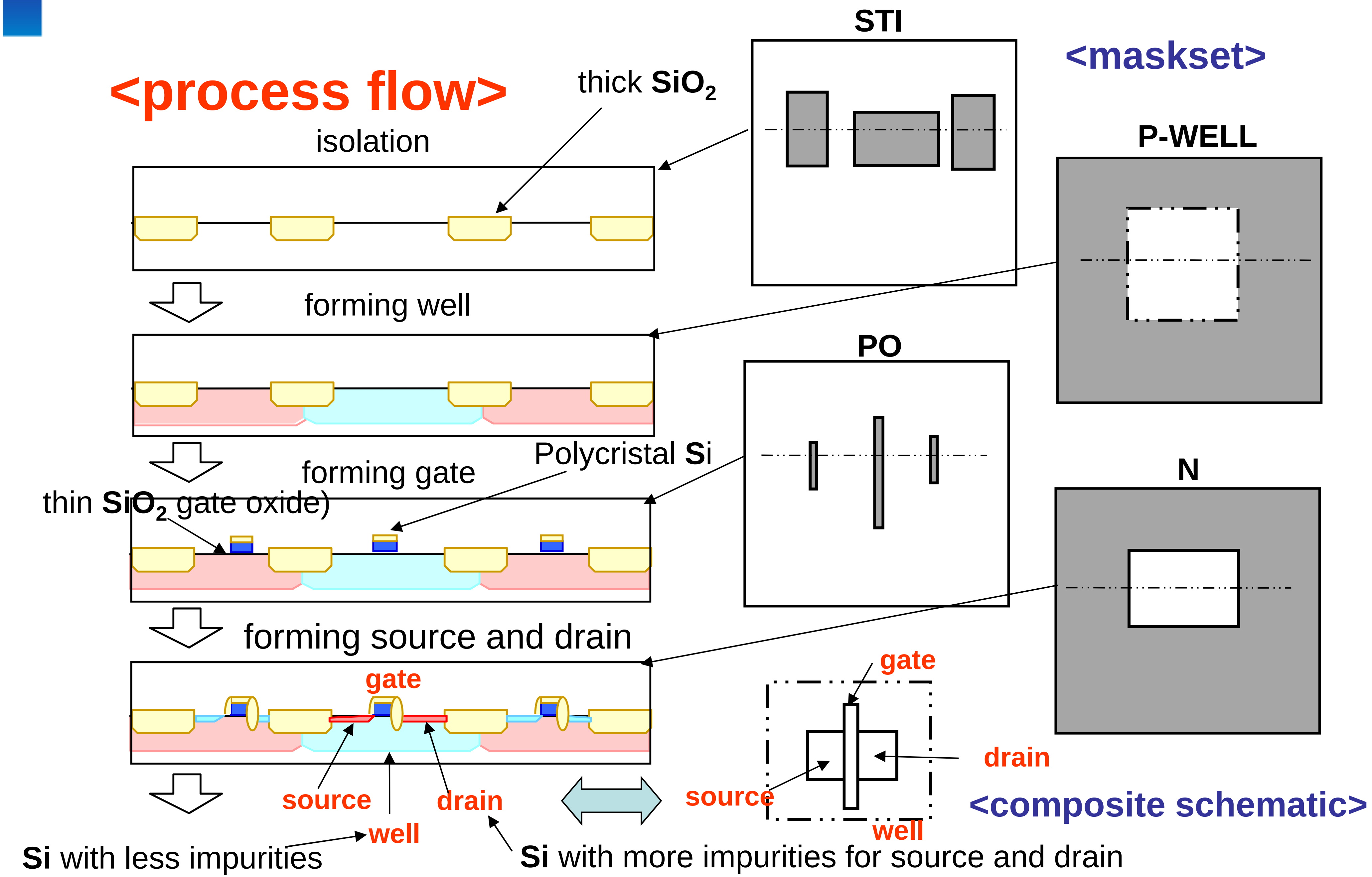
etching

implantation

oxidation

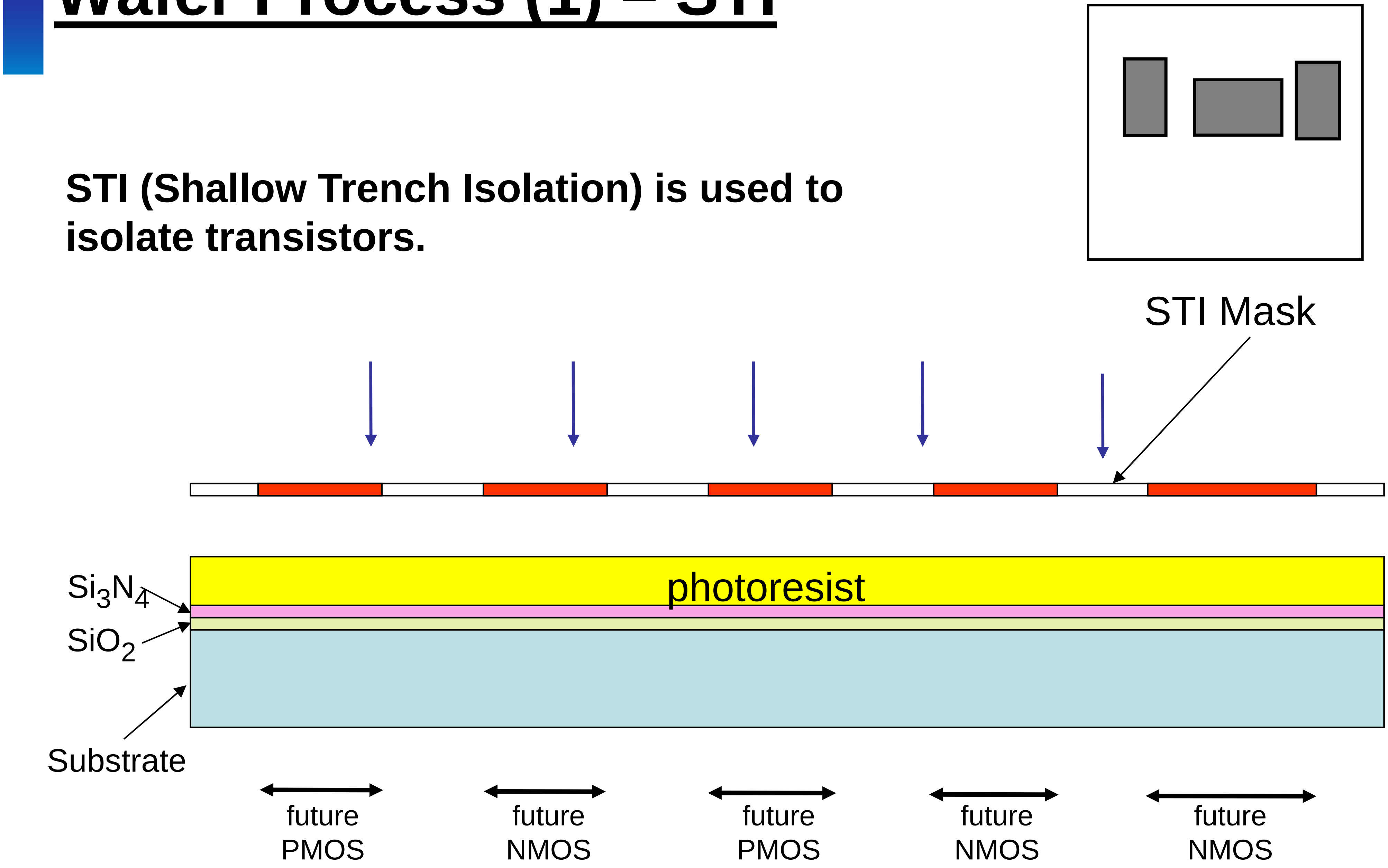
BD001

Shaping Transistor



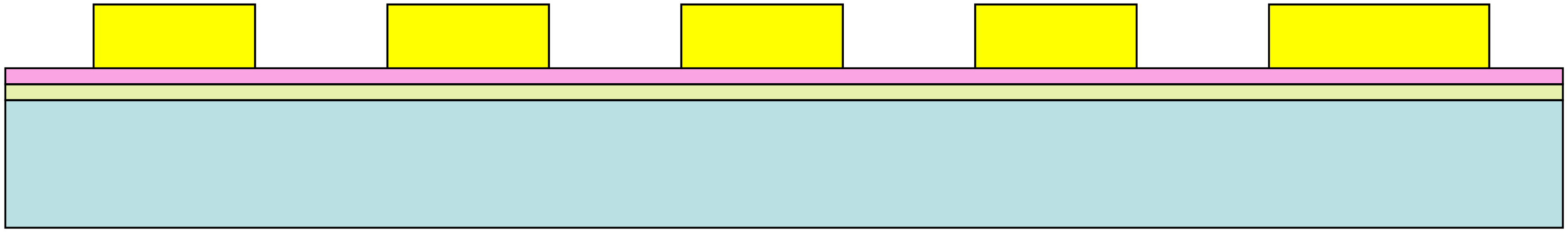
Wafer Process (1) – STI

STI (Shallow Trench Isolation) is used to isolate transistors.

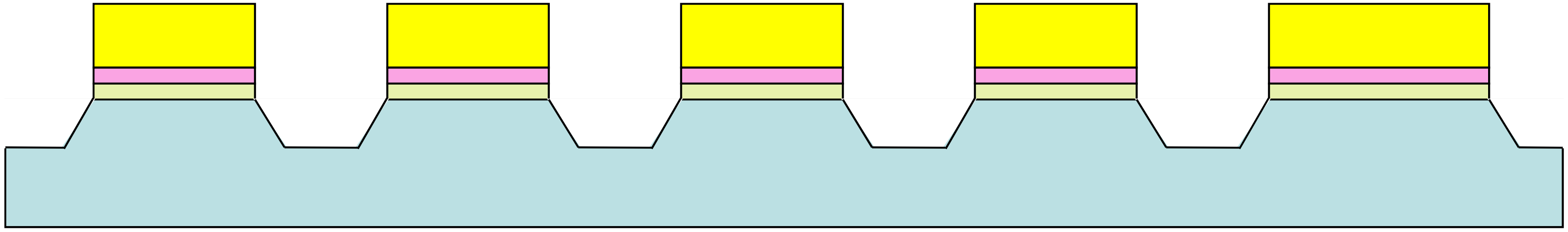


Wafer Process (2) – STI

Photoresist developed

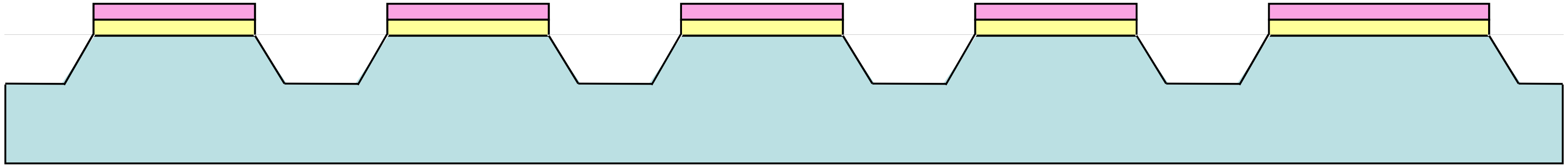


Substrate etched by plasma

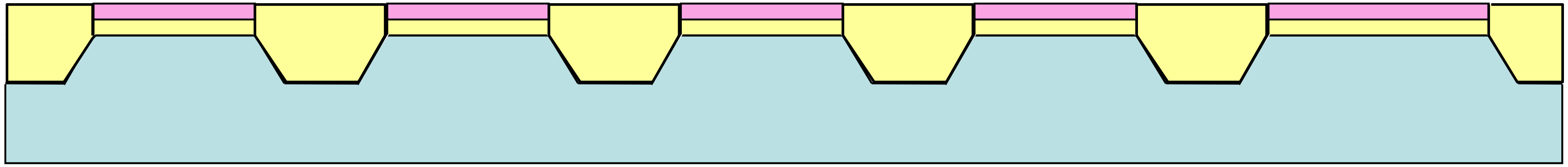


Wafer Process (3) – STI

Photoresist removed

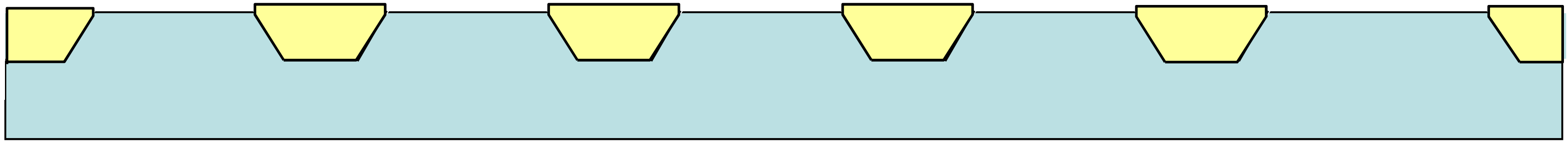


Oxidation and CMP (Chemical Mechanical Polishing)



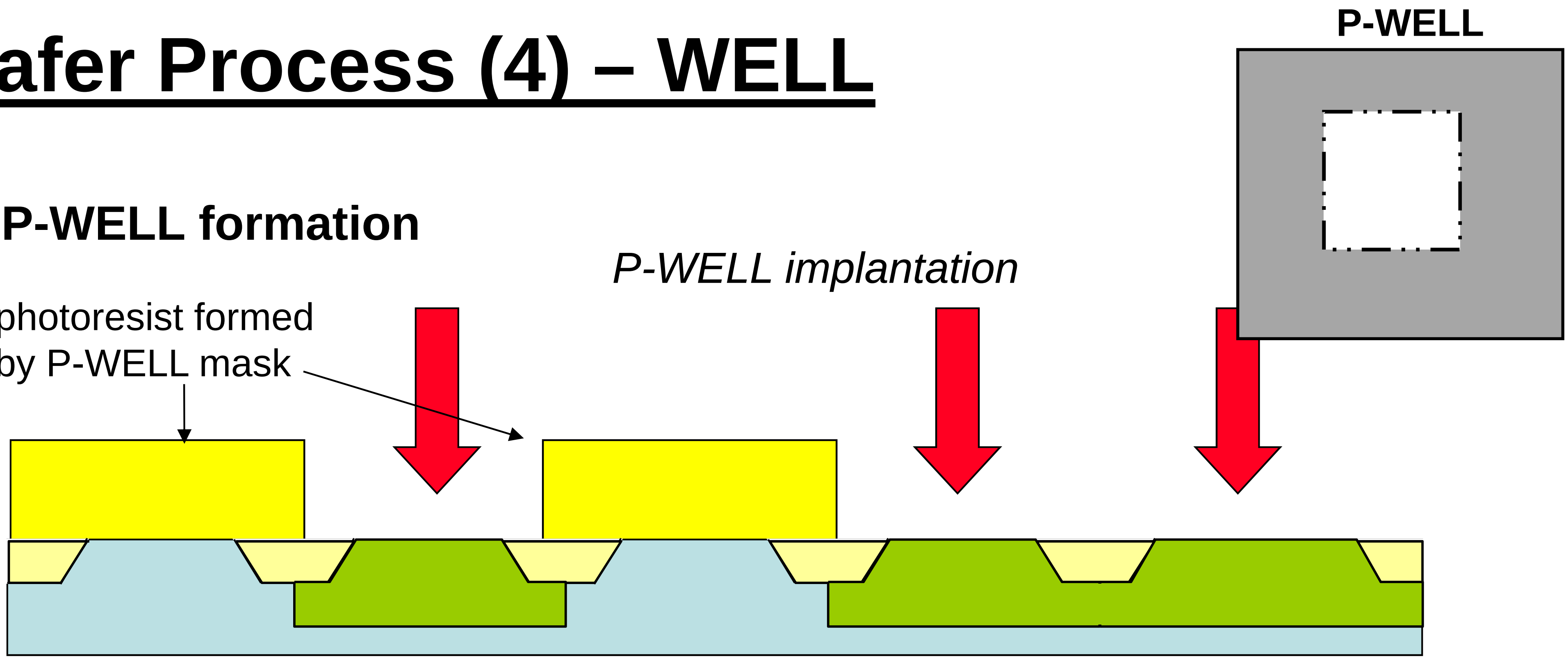
Si_3N_4 (P-SiN)is used as a barrier to oxidation.
CMP is used for planarization.

STI etch back and P-SiN etching



Wafer Process (4) – WELL

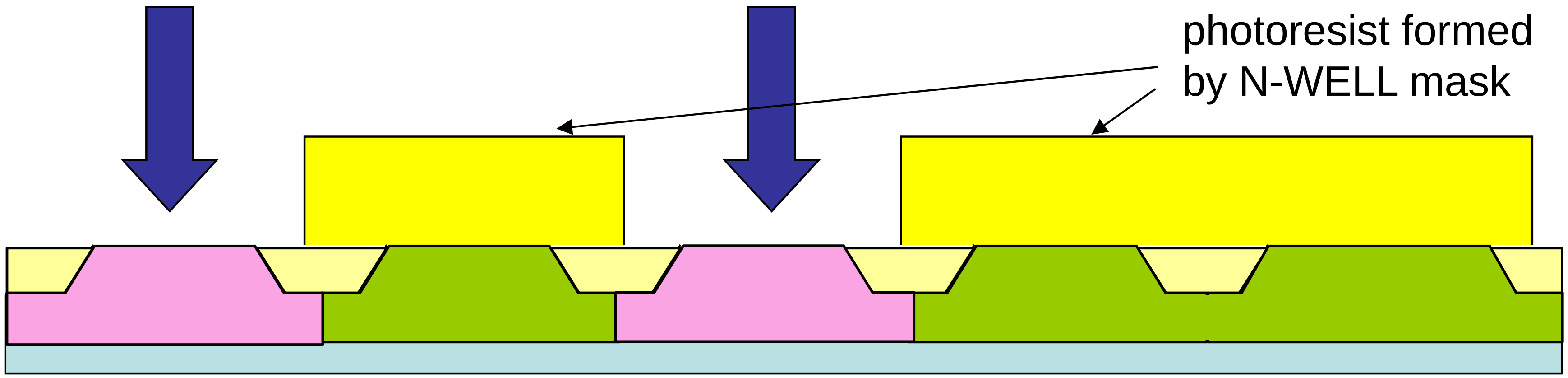
P-WELL formation


P-WELL

P-WELL implantation

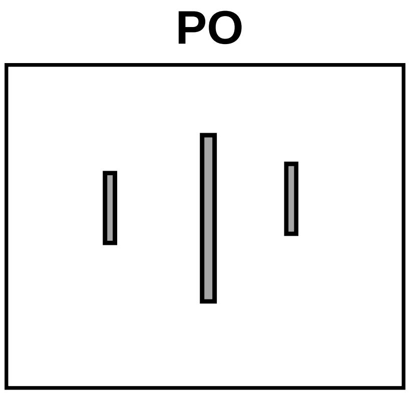
photoresist formed by P-WELL mask

N-WELL formation

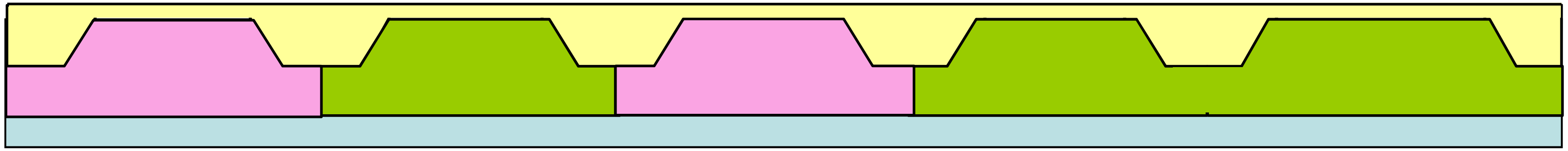

N-WELL implantation

photoresist formed by N-WELL mask

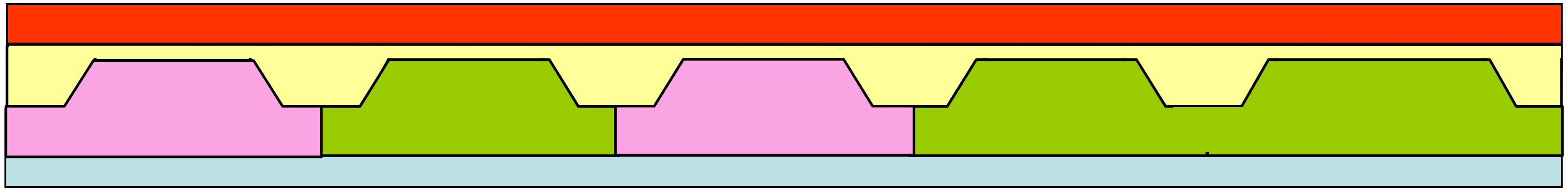
Wafer Process (5) – Gate



Gate Oxidation

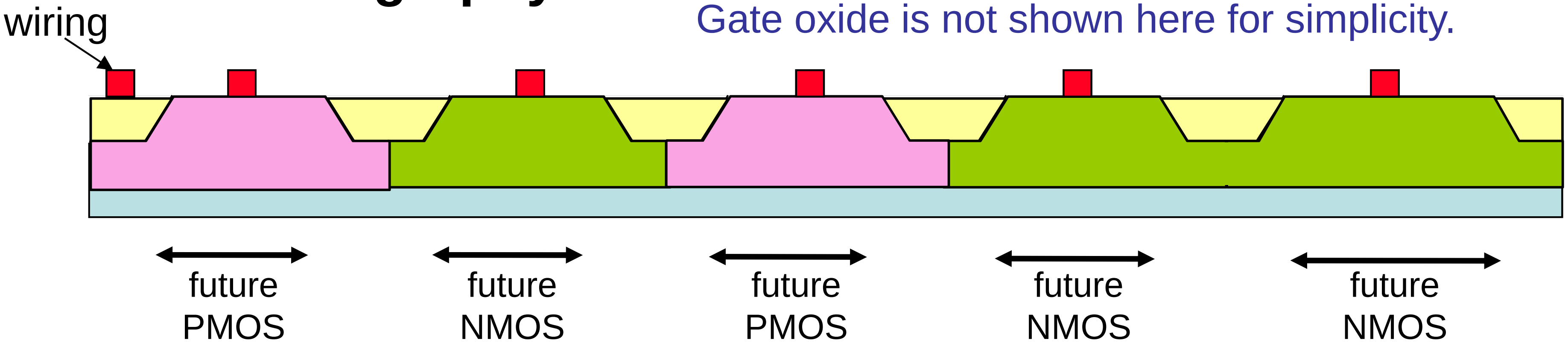


Poly silicon deposition

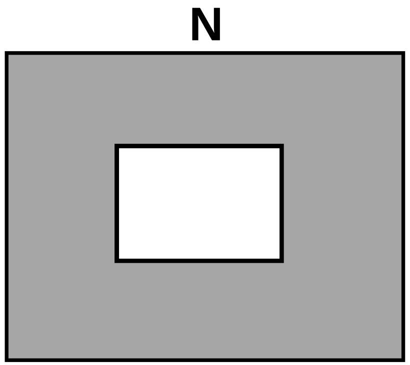


Photolithography

Gate oxide is not shown here for simplicity.

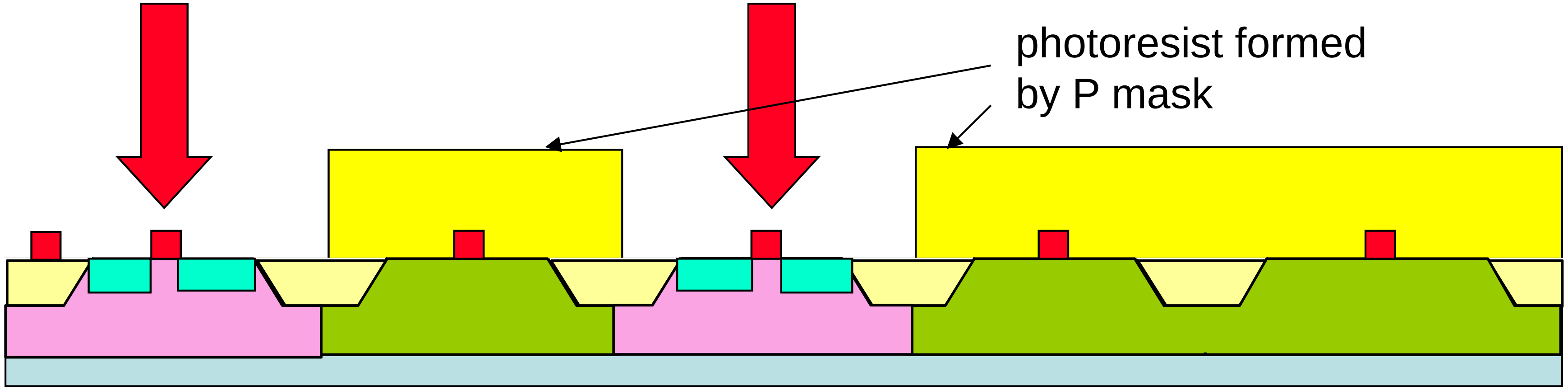


Wafer Process (6) – Source/Drain



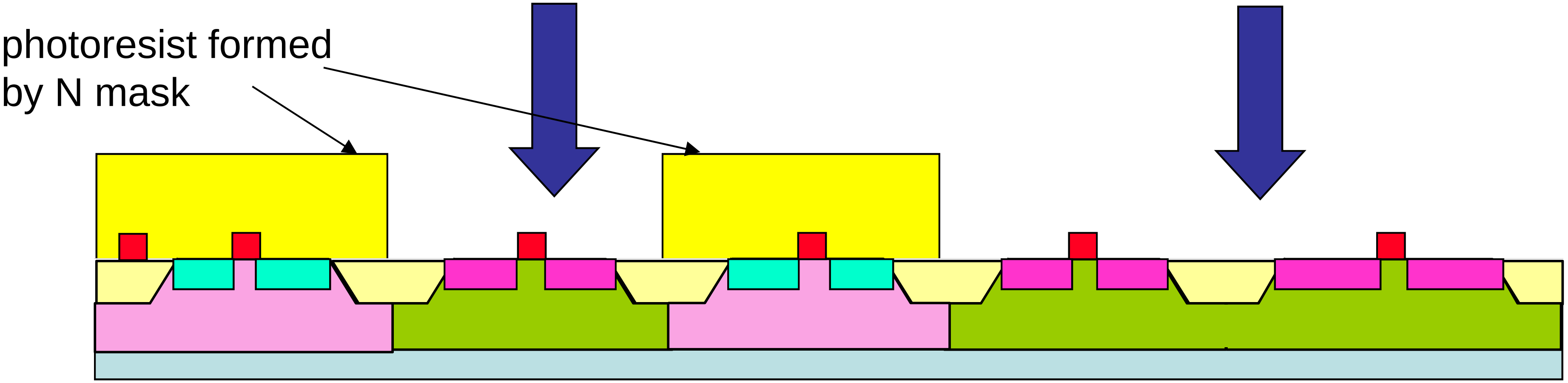
P+ Diffusion

high concentration ion implantation (acceptor)



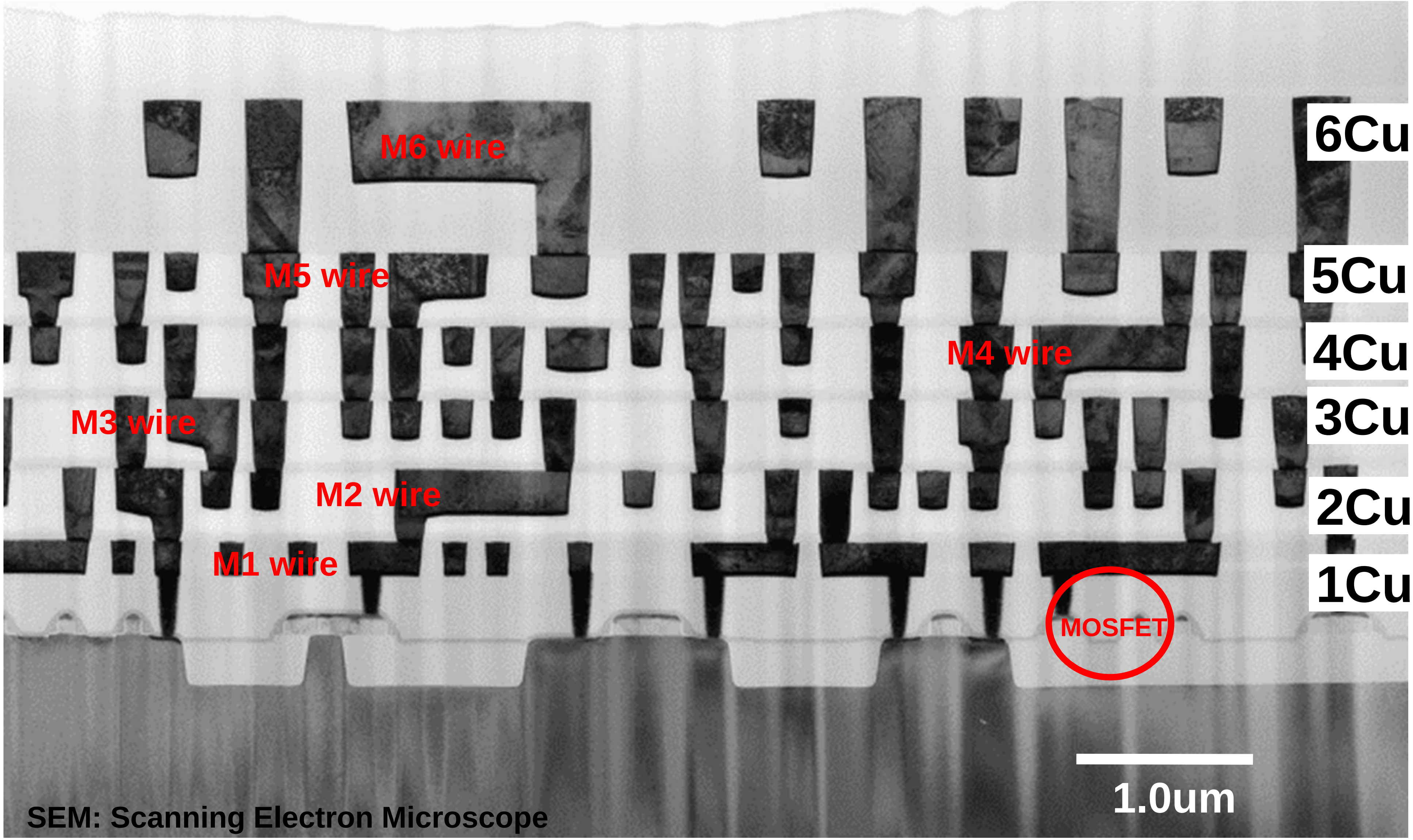
N+ Diffusion

high concentration ion implantation (donor)



Final Cross-section

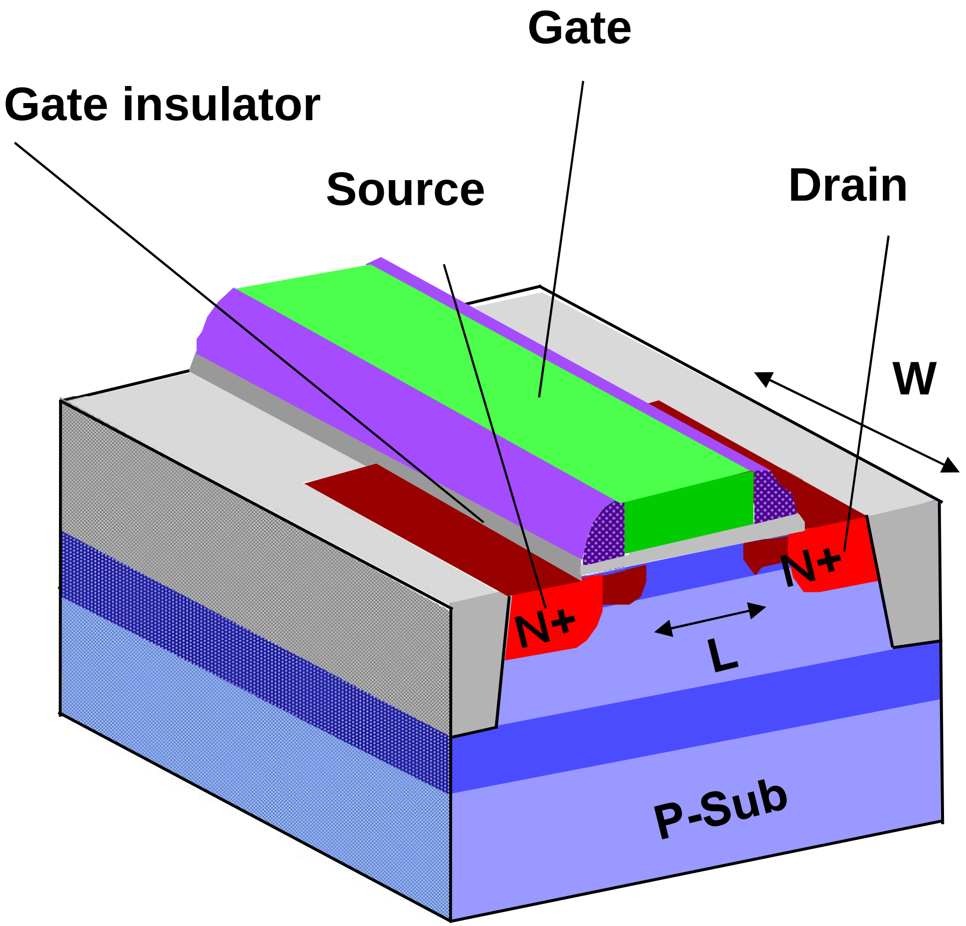
Cross-section SEM photograph (6 layers Cu metal wire)



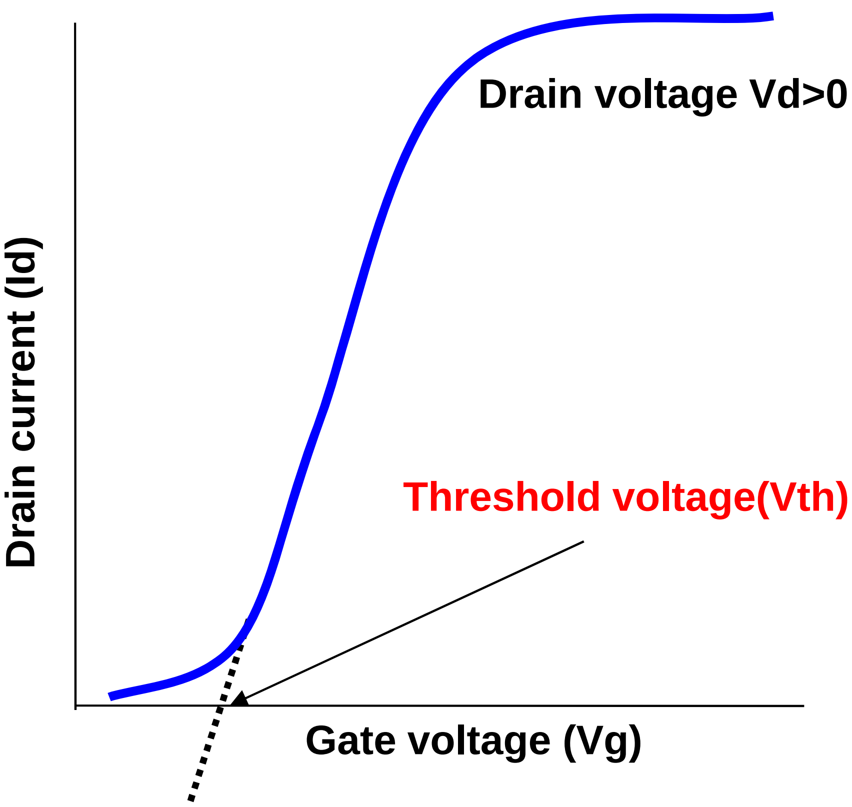
SEM: Scanning Electron Microscope

MOSFET

*** focusing on n-channel transistor ***



L: Channel length
W: Channel width

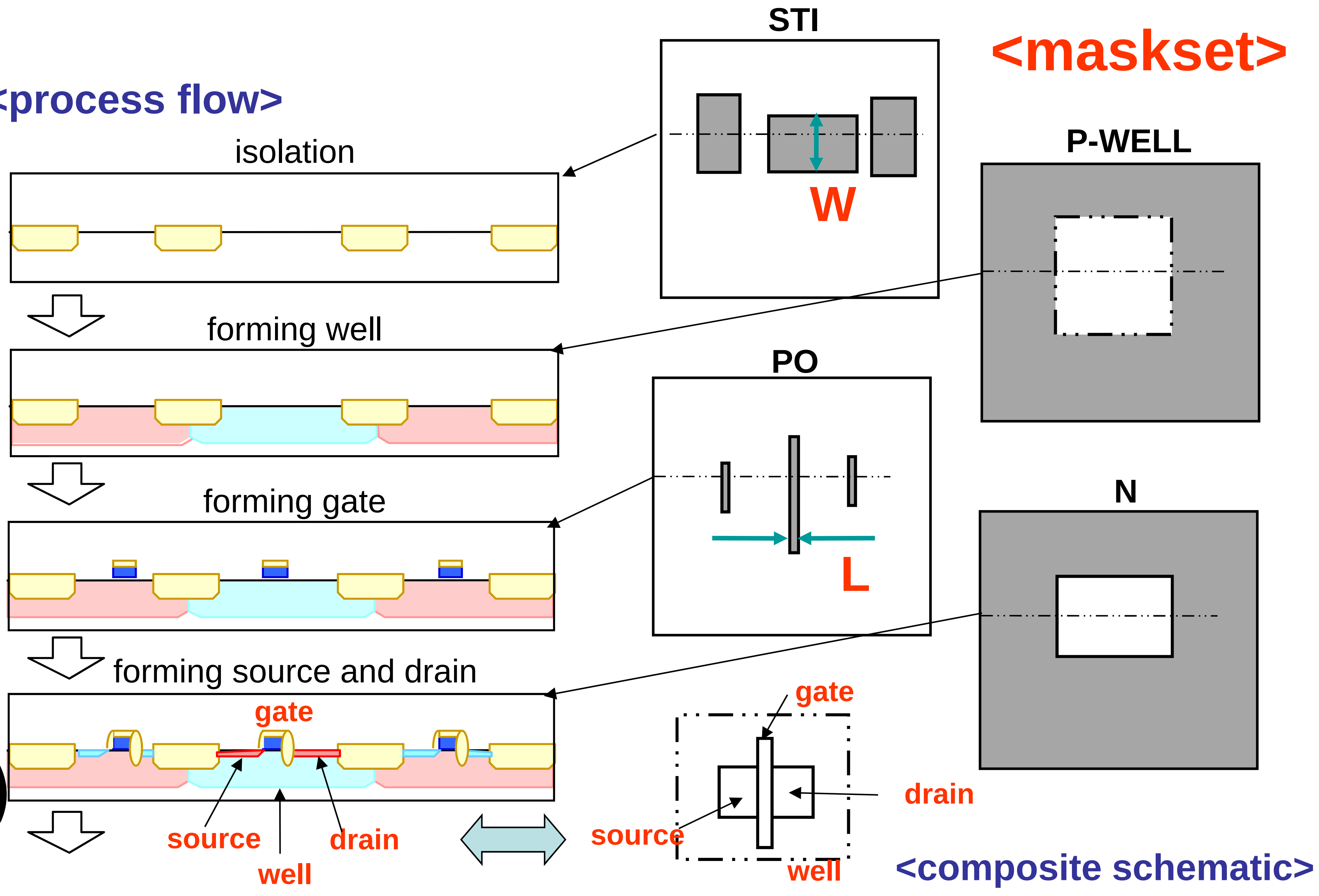


FET: Field Effect Transistor

Shaping Transistor

<process flow>

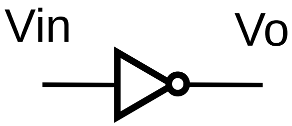
<maskset>



<composite schematic>

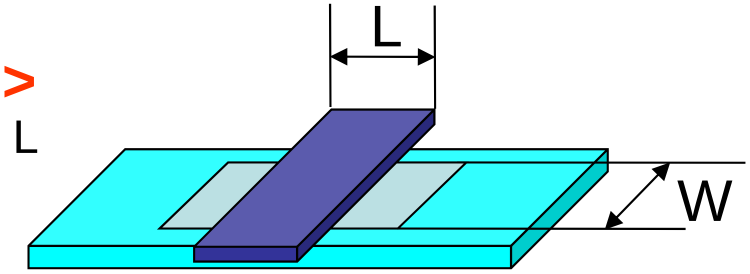
Inverter Design

<cell symbol>

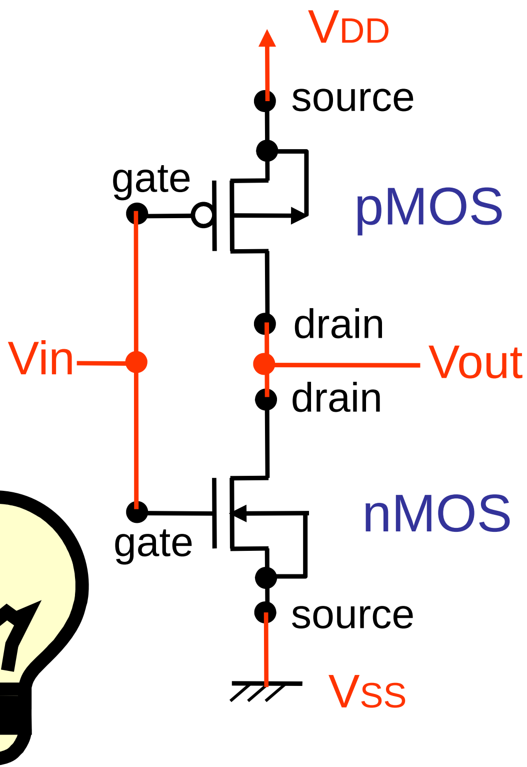


<circuit design>

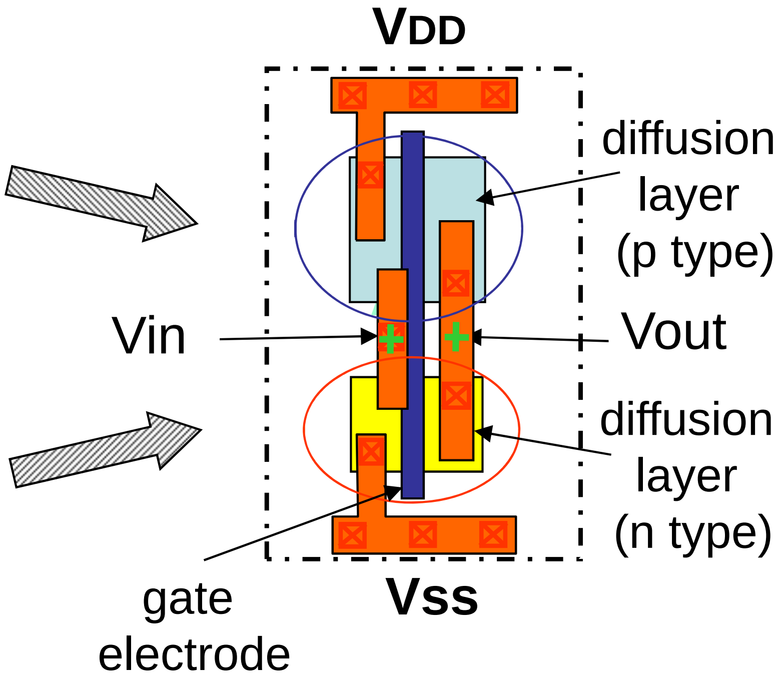
Define W and L depending on speed and driveabilty



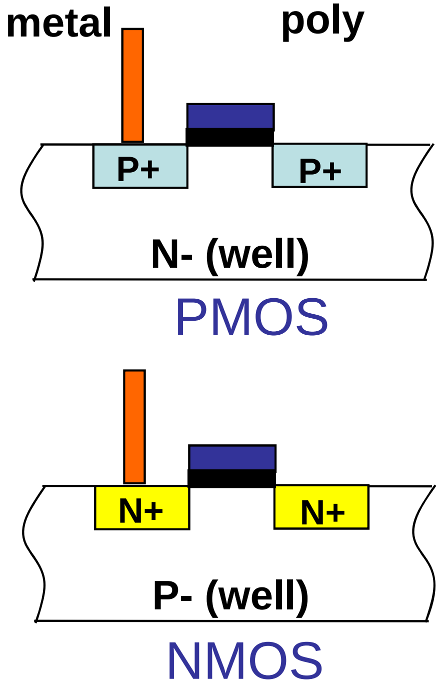
<circuit schematics>



<composit schematics>



<cross section>



-well connection not shown for simplicity-

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

    Platform Development

        Device Design

        Circuit Design

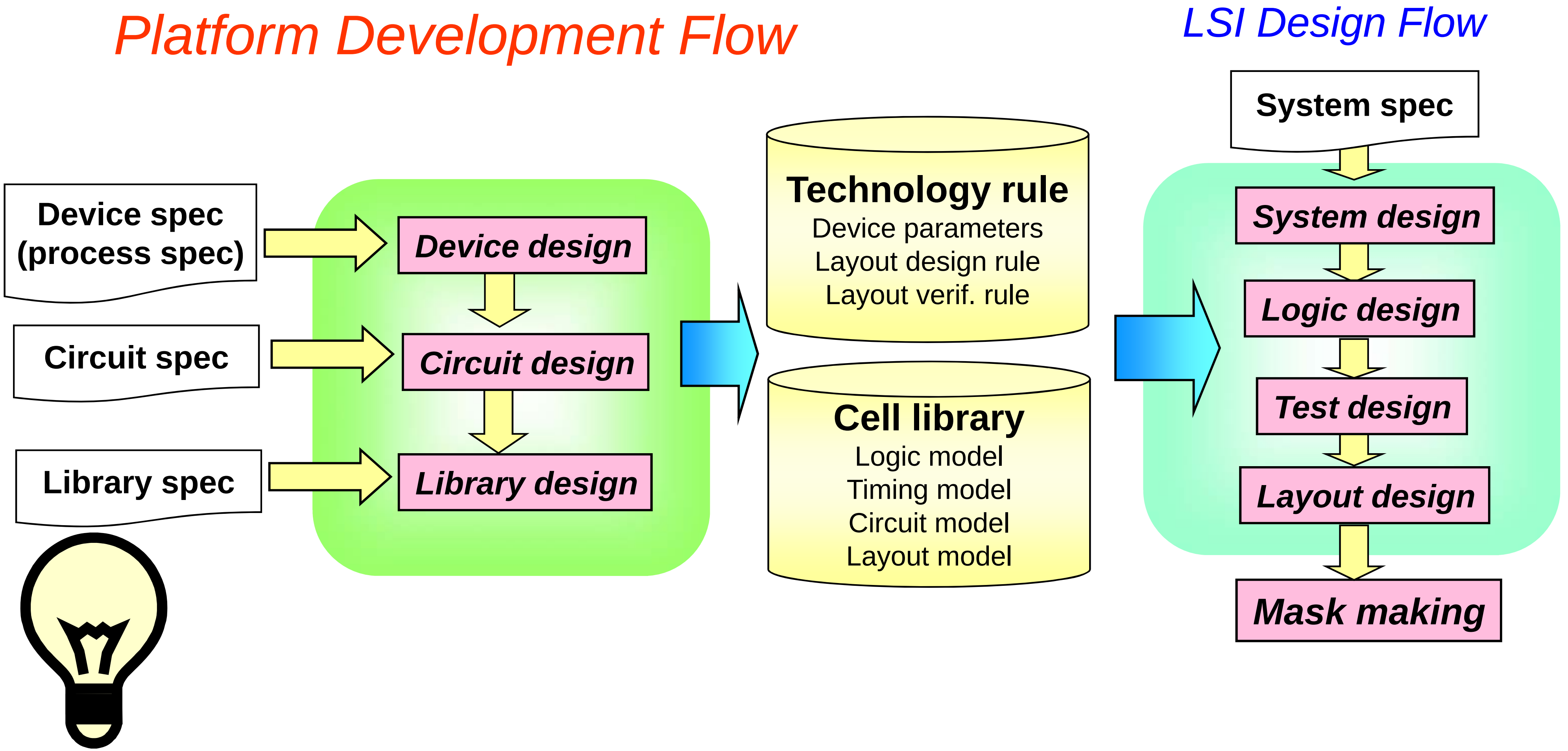
        Library Design

    LSI Design Flow

Total LSI Design Flow

To design LSIs using design environment established by platform development

- Quality of the platform determines quality of all products



Platform Development Flow

LSI Design Flow

Device spec (process spec)

Circuit spec

Library spec

Device design

Circuit design

Library design

Technology rule

Device parameters

Layout design rule

Layout verif. rule

Cell library

Logic model

Timing model

Circuit model

Layout model

System spec

System design

Logic design

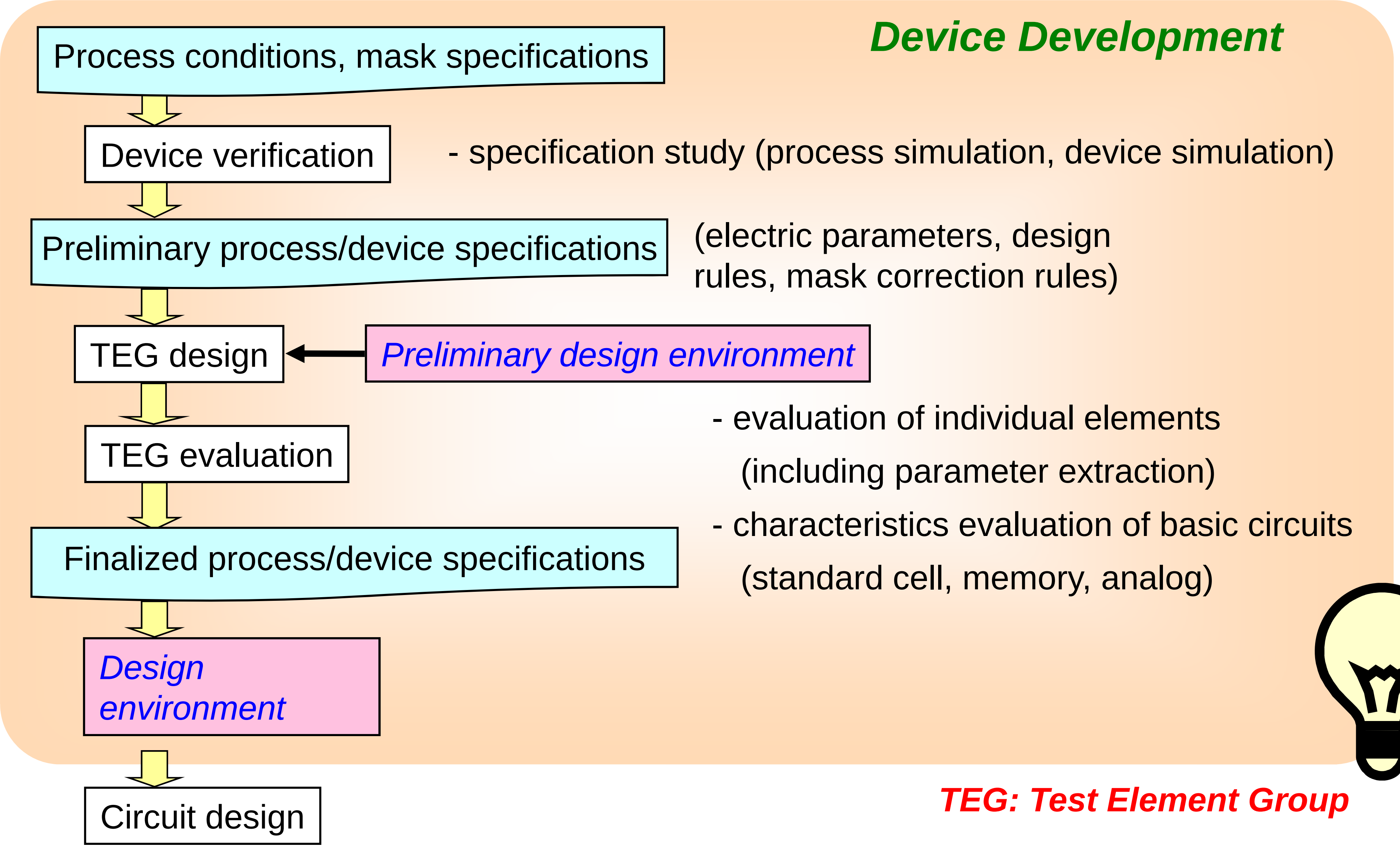
Test design

Layout design

Mask making

Device Design

Determine transistor characteristics and parasitic device parameters



Device Development

Process conditions, mask specifications

Device verification

- specification study (process simulation, device simulation)

Preliminary process/device specifications

(electric parameters, design rules, mask correction rules)

TEG design

Preliminary design environment

TEG evaluation

- evaluation of individual elements (including parameter extraction)
- characteristics evaluation of basic circuits (standard cell, memory, analog)

Finalized process/device specifications

Design environment

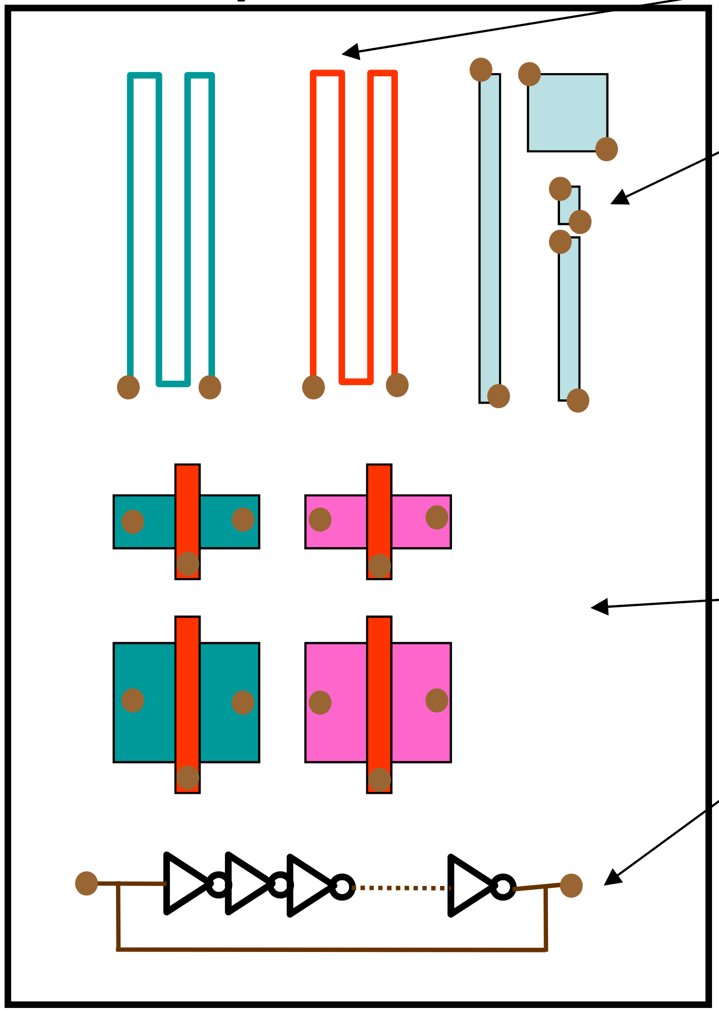
Circuit design

TEG: Test Element Group

TEG

Test Element Group = test chip

TEG top view

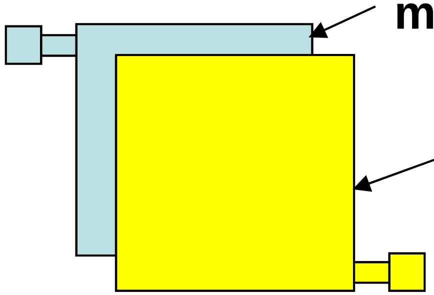


● Probing pad

Resistor strings

poly, N^+, P^+, WELL, metals

Capacitors with different shape



Transistors with different shape

Ring oscillators

(odd-stage inverter chain)

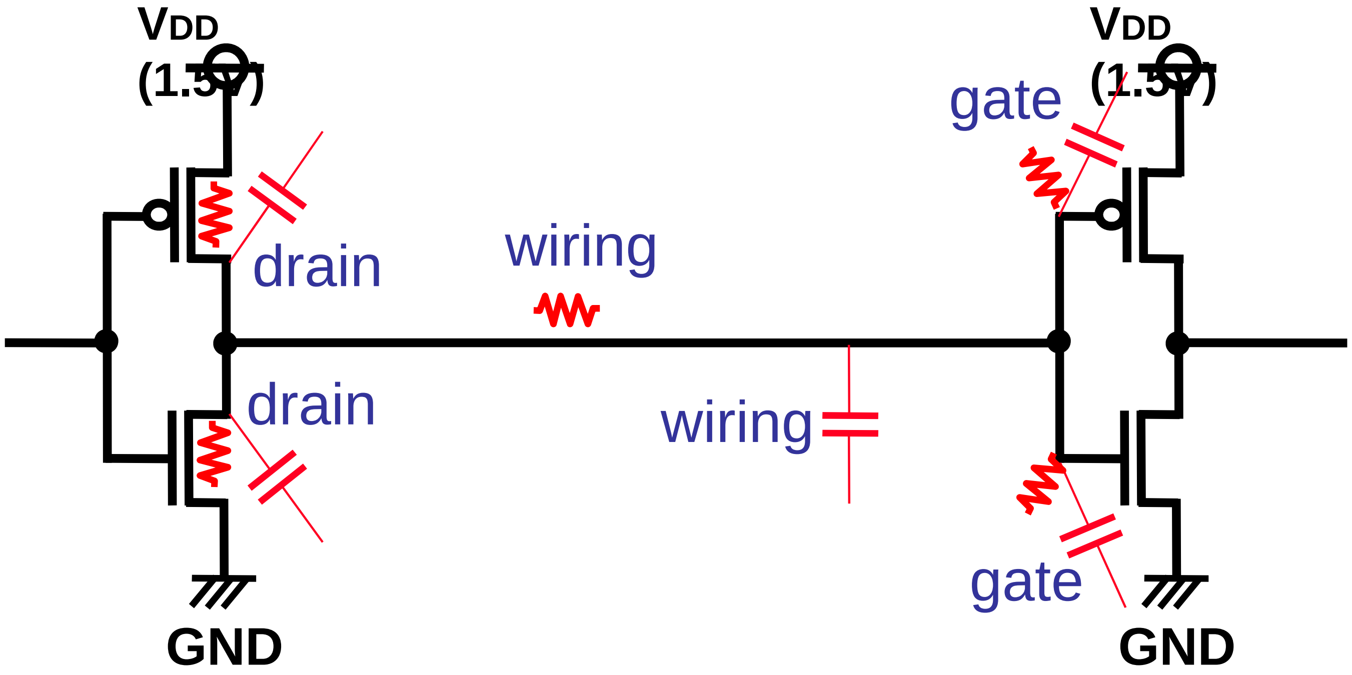
Various parameters required for circuit modeling are extracted by evaluating TEG

Circuit Model

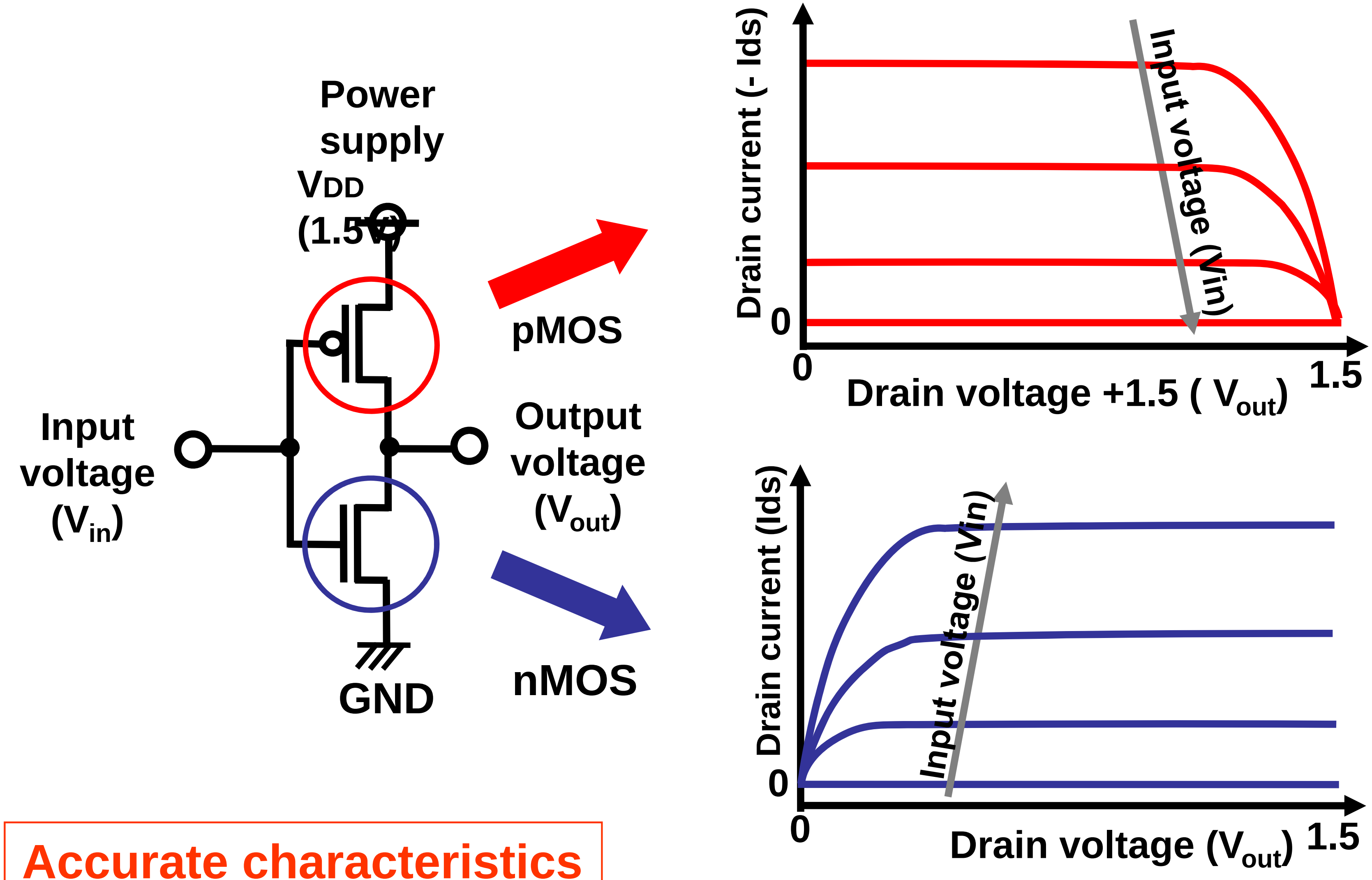
- For estimating delay time and power consumption -

An inverter chain with parasitic devices will be investigated.

Accurate parameters are required

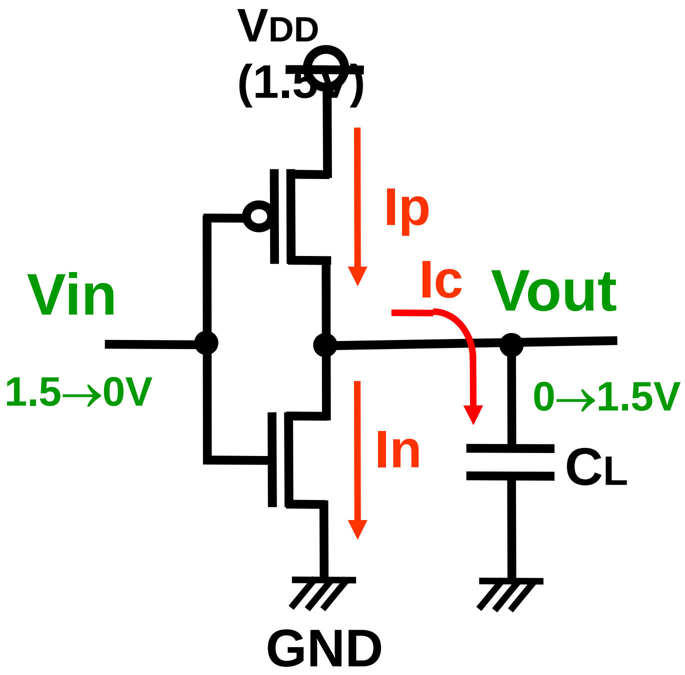


Current Characteristics



Accurate characteristics are required

Actual Circuit Behavior



$Ip = Ic + In$

$Ip = fp(Vin, Vout, V_{DD}, V_{thp})$

$In = fn(Vin, Vout, V_{DD}, V_{thn})$

$$Ic = C_L \frac{dVout}{dt}$$

$$\boxed{\frac{C_L \cdot V_{DD}}{\beta p \cdot (V_{DD} - |V_{thp}|)^2}}$$

was just a simplified equation. Actual behavior is obtained by circuit simulation based on differential equations: SPICE.

Device Development and EDA Tools

TEG evaluation facilitates to determine:

1) Characteristics of transistors

- SPICE parameters (BSIM3/BSIM4)
- parameters for circuit design tools

2) Wiring structure

- capacitor parameters (voltage dependent, temperature dependent)
- resistor parameters (temperature dependent)

3) Design rules (rules for mask patterns)

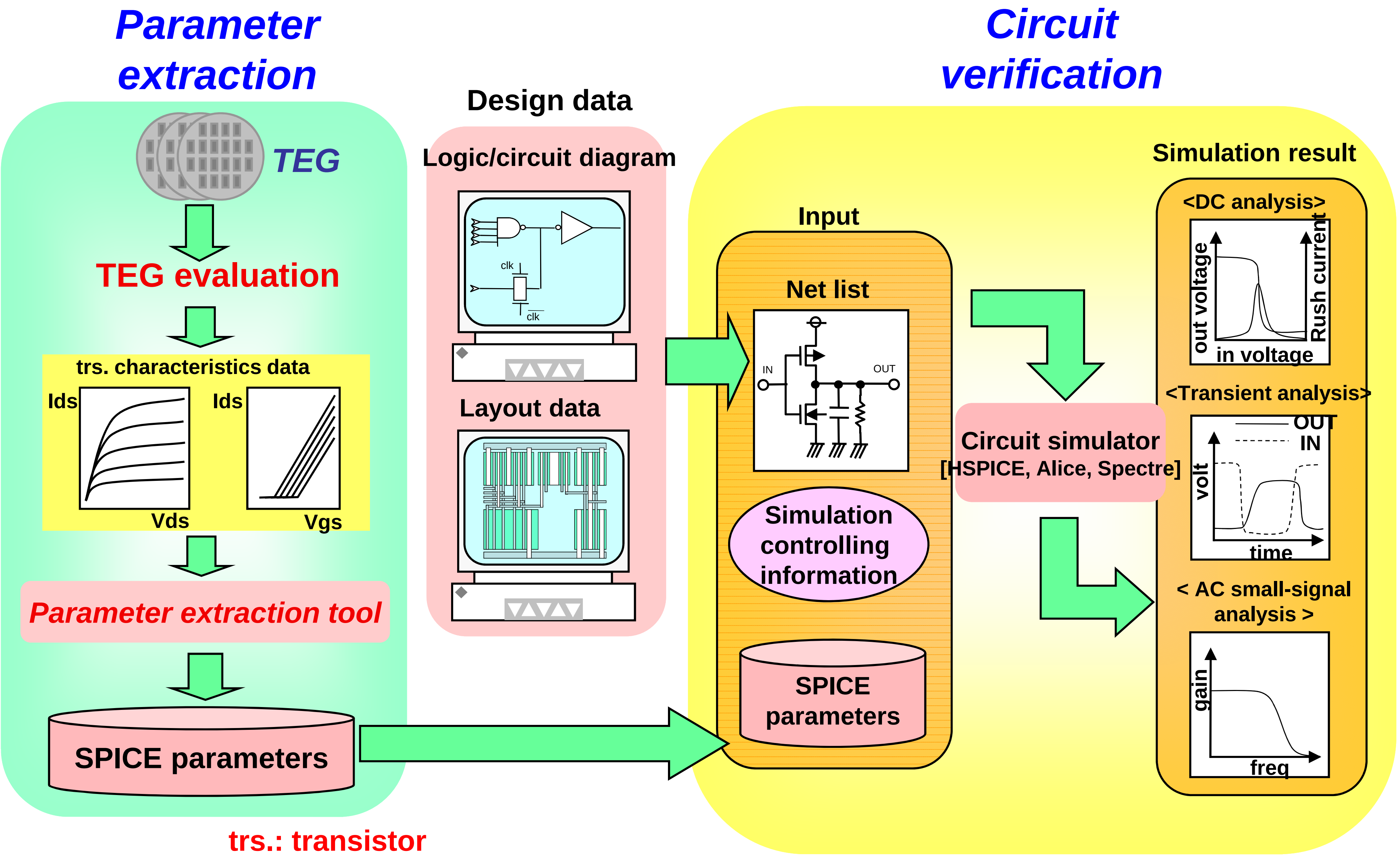
- parameters for physical data checkers (DRC/LVS/ERC)
- data for parameter extraction tools (LPE)
- parameters of OPC for mask data modification

EDA: Electronic Design Automation

LPE: Layout Parasitic Extraction

OPC: Optical Proximity Correction

SPICE



Parameter extraction

TEG

TEG evaluation

trs. characteristics data

Parameter extraction tool

SPICE parameters

Design data

Logic/circuit diagram

Layout data

Circuit verification

Input

Net list

Simulation controlling information

SPICE parameters

Circuit simulator [HSPICE, Alice, Spectre]

Simulation result

<DC analysis>

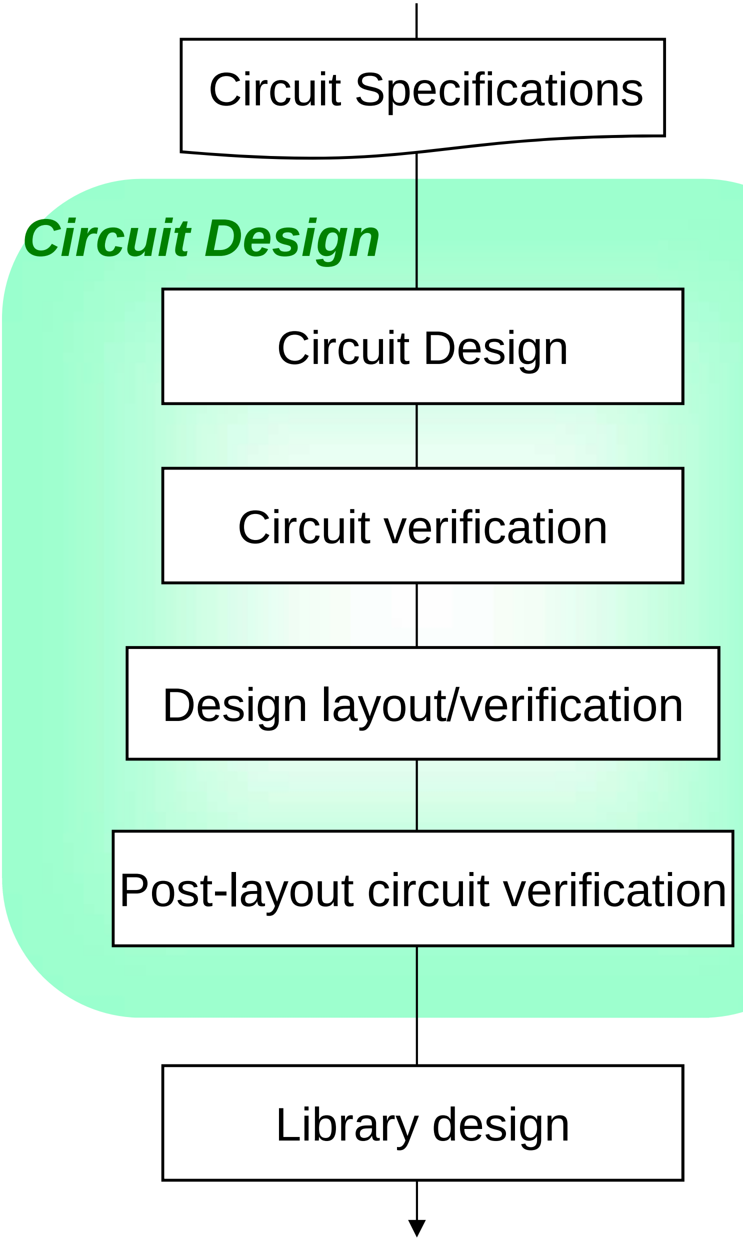
<Transient analysis>

< AC small-signal analysis >

trs.: transistor

Circuit Design

Determine circuits of basic circuits (libraries)



Schematic entry using device symbols (trs. C. R.) [CADENCE dfII, Composer]

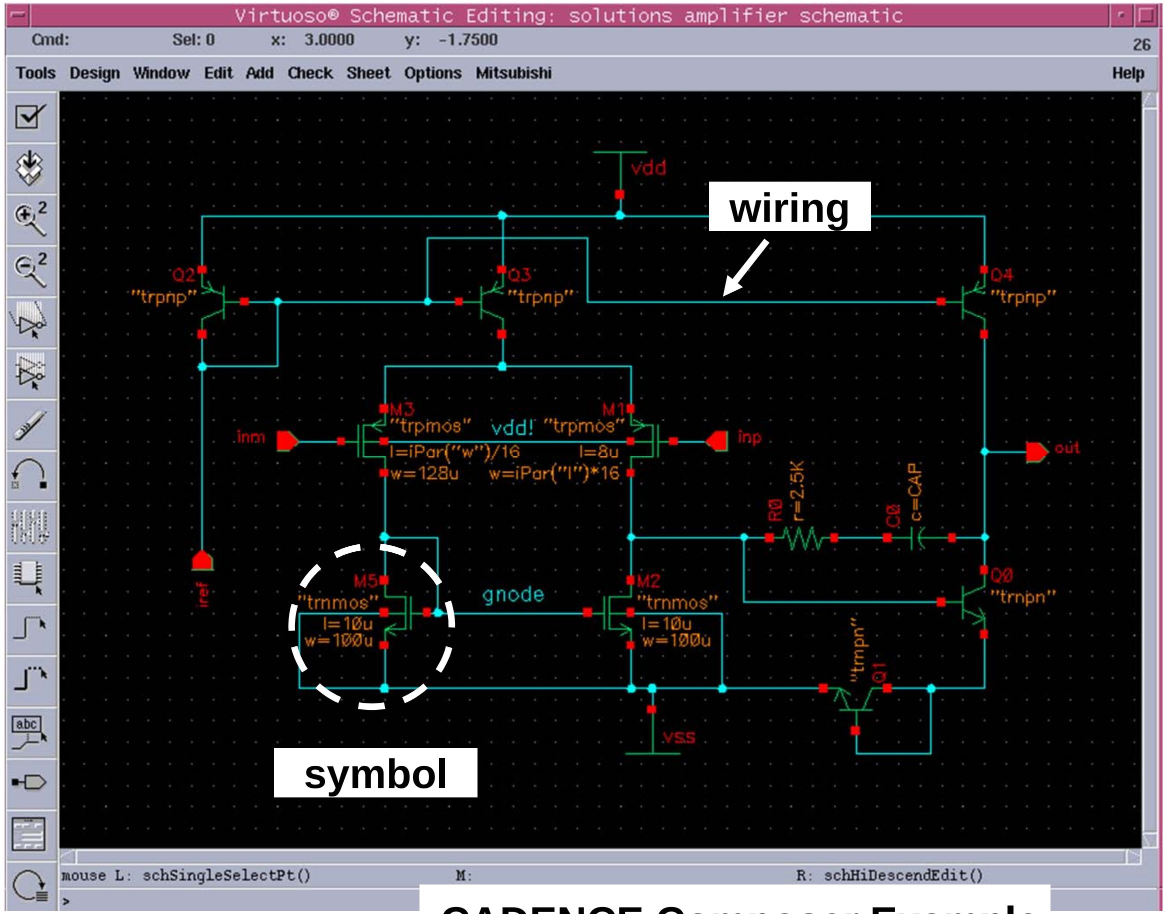
Confirm whether circuit specifications (DC/AC characteristics, functions) are satisfied [Spectre/ALICE/HSPICE, Verilog-Spectre etc.]

Confirm whether it satisfies the process rules by editing layout data Virtuoso/Virtuoso-XL, Calibre/Assura/Diva etc.]

Reconfirm circuit specifications with post-layout load conditions (actual shapes of trs. and wiring) [Columbus-AMS/Assura-RCX etc]

Schematic Entry

◆ Electric circuits are inputted with circuit schematic editor interactively, where pre-determined device symbols, such as transistors, capacitors, and resistors, are used.



Virtuoso® Schematic Editing: solutions amplifier schematic

Cmd: Sel: 0 x: 3.0000 y: -1.7500 26

mouse L: schSingleSelectPt() M: R: schHiDescendEdit()

wiring

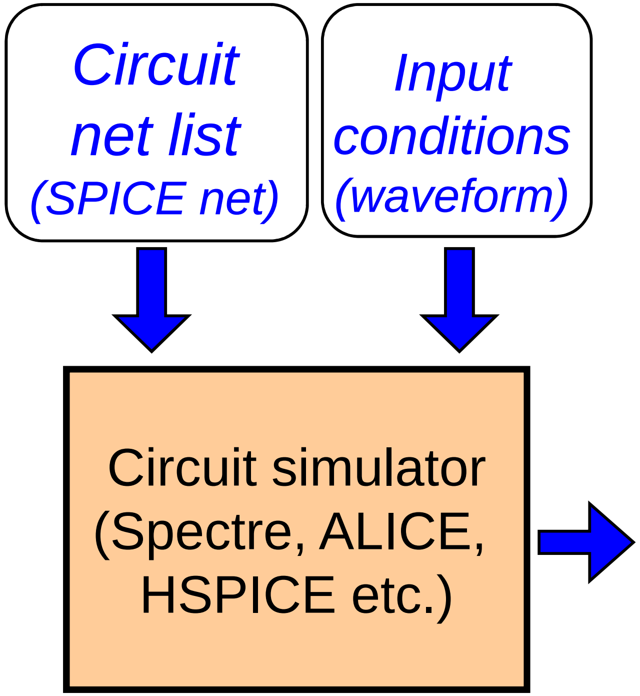
symbol

CADENCE Composer Example

Verification

- Confirm desired output is obtained from circuit net list under certain input conditions and with electrical characteristic parameters:
  DC analysis, transient analysis and AC analysis.

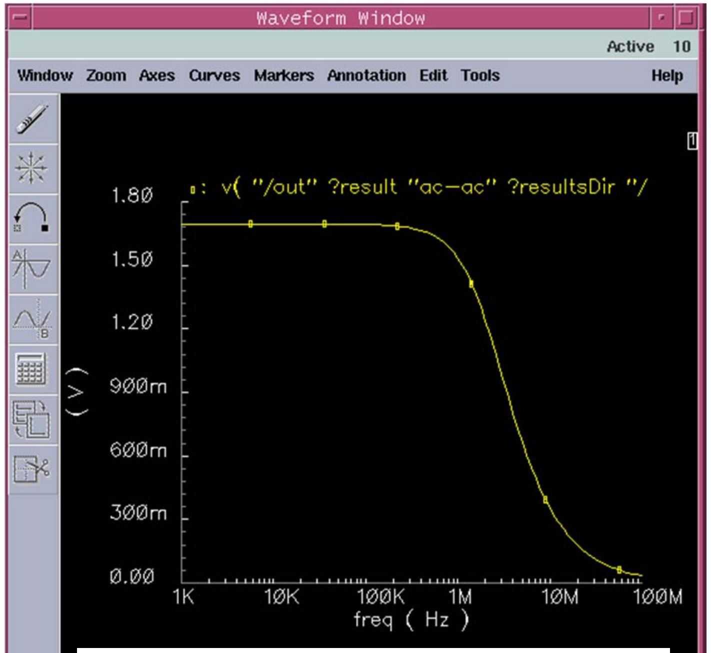
by visual confirmation



Circuit net list (SPICE net)

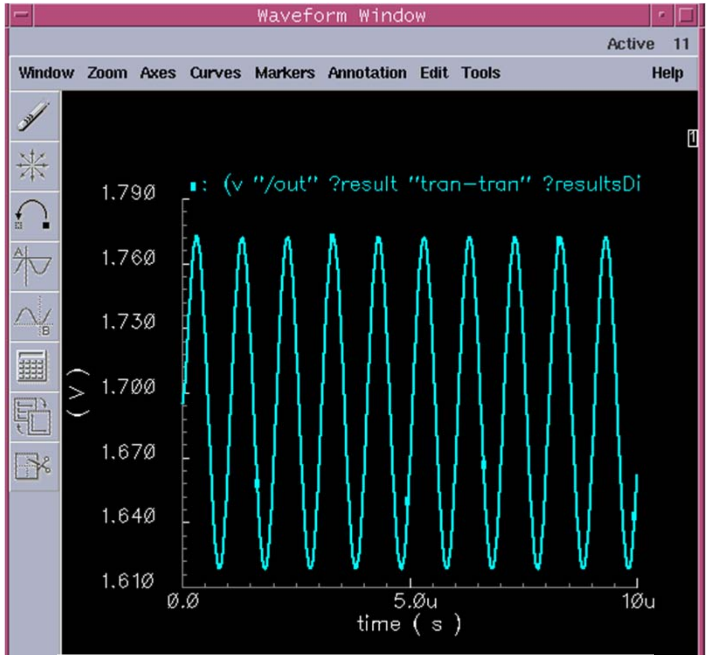
Input conditions (waveform)

Circuit simulator (Spectre, ALICE, HSPICE etc.)



Waveform Window
Active 10
Window Zoom Axes Curves Markers Annotation Edit Tools Help
: v("/out" ?result "ac-ac" ?resultsDir "/
1.80
1.50
1.20
900m
600m
300m
0.00
(V)
1K
10K
100K
1M
10M
100M
freq (Hz)

AC analysis example (frequency characteristics)

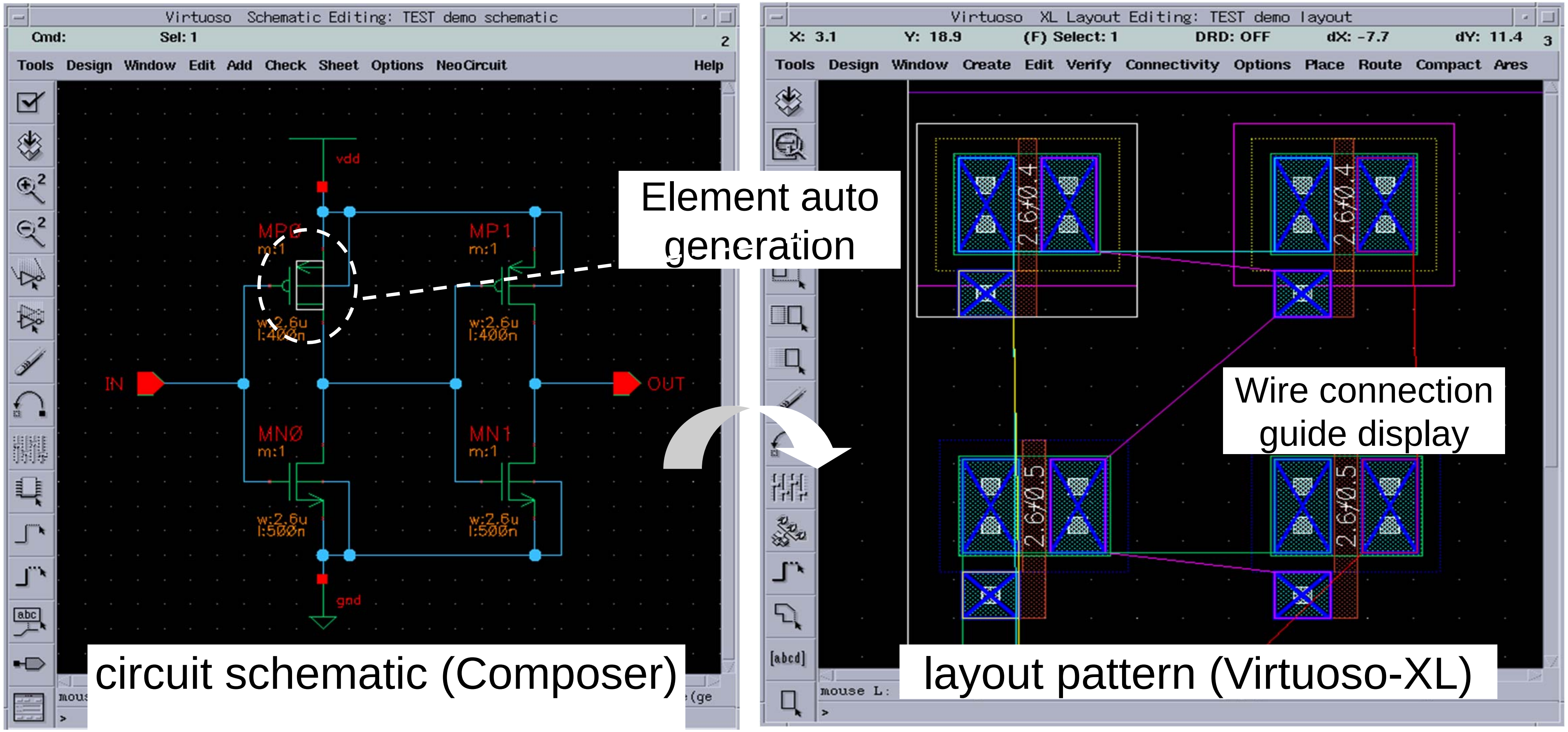


Waveform Window
Active 11
Window Zoom Axes Curves Markers Annotation Edit Tools Help
: (v "/out" ?result "tran-tran" ?resultsDi
1.790
1.760
1.730
1.700
1.670
1.640
1.610
(V)
0.0
5.0u
10u
time (s)

transient analysis example (output waveform)

Layout

- ◆ **Layout design environment by net-list-driven automatic layout tool, coordinated with circuit design environment**


Virtuoso Schematic Editing: TEST demo schematic
Cmd: Sel: 1 2
Tools Design Window Edit Add Check Sheet Options NeoCircuit Help
vdd
MPØ m:1 w:2.6u l:400n
MP1 m:1 w:2.6u l:400n
IN
OUT
MNØ m:1 w:2.6u l:500n
MN1 m:1 w:2.6u l:500n
gnd
Virtuoso XL Layout Editing: TEST demo layout
X: 3.1 Y: 18.9 (F) Select: 1 DRD: OFF dX: -7.7 dY: 11.4 3
Tools Design Window Create Edit Verify Connectivity Options Place Route Compact Ares
2.6+0.4
2.6+0.4
2.6+0.5
2.6+0.5
mouse L:

Element auto generation

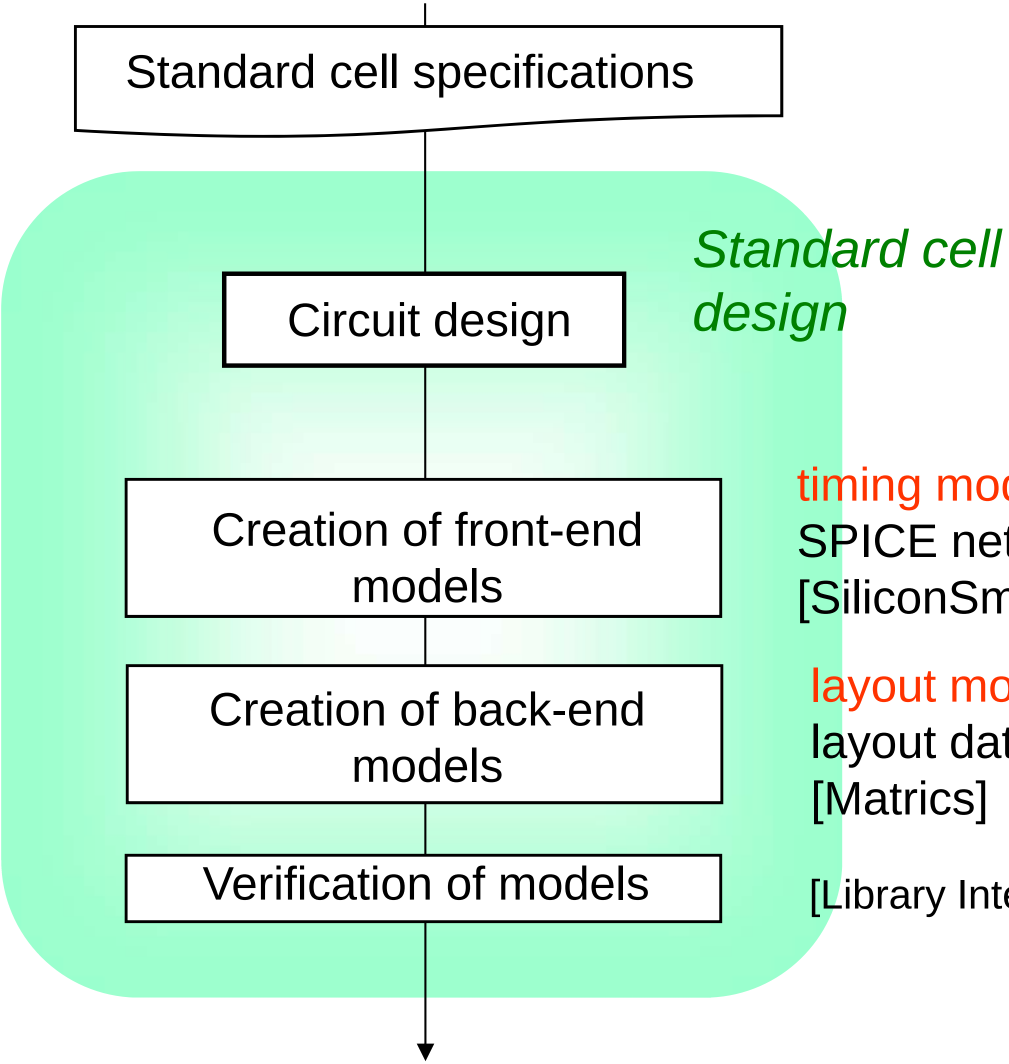
Wire connection guide display

circuit schematic (Composer)

layout pattern (Virtuoso-XL)

Library Design Flow

Example: standard cell


Standard cell specifications
Circuit design
Creation of front-end models
Creation of back-end models
Verification of models
Standard cell design

timing models (e.g. Liberty) created from SPICE net with actual load)
[SiliconSmart CR]

layout models (e.g. LEF) created from layout data (GDSII)
[Matrics]

[Library Integrator/Matrics]

Library Design

Make basic circuits reusable

Definitions

- **Core cell:** commonly used parts of basic logic circuits, such as inverters, logic gates, FFs, and I/O buffer cells; also referred to as standard cell
- **Libraries:** design package commonly used for LSI designs, comprising the following major data:

- ➢Logic model : RTL description, Verilog HDL,VHDL
- ➢Timing model : Liberty (Synopsys model) (including delay, timing constraints, power consumption model)

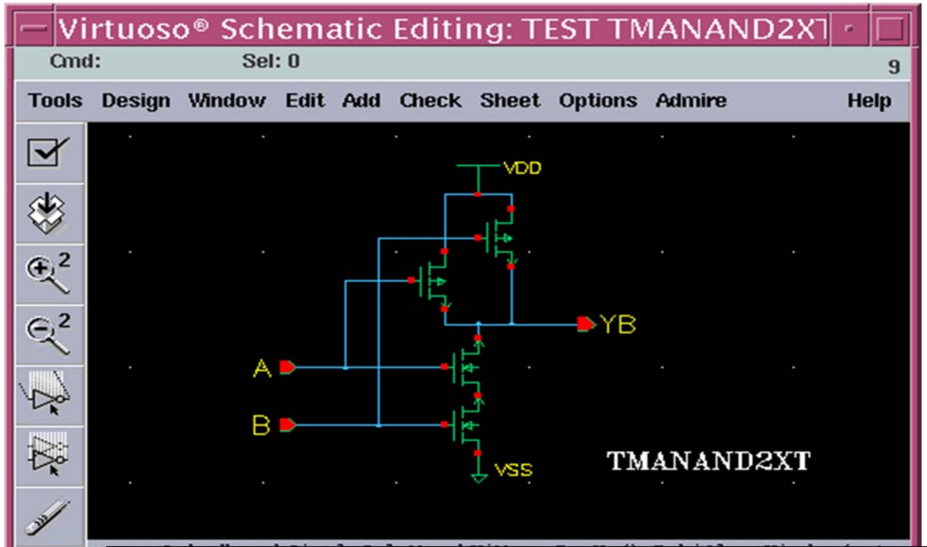
models for front-end designs

- ➢Layout model: GDSII, LEF
- ➢Circuit model: SPICE netlist, CDL netlist

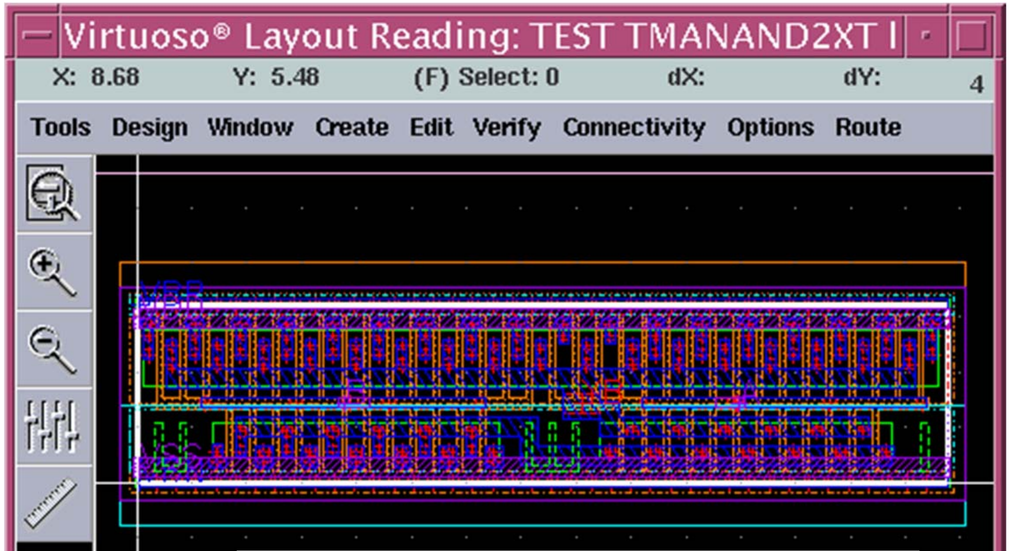
models for back-end designs

Library Examples

Circuit schematics [Composer]



Layout data [Virtuoso]



Timing model [Liberty]

```
cell(TMANAND2XT){
 area : 3;
 cell_leakage_power : 0.290400000000000;
 cell_footprint : NAND2X_;
 pin(A){
  max_transition : 0.6500;
  capacitance : 0.004841;
  direction : input;
  internal_power() {
   when : "!B";
   fall_power(pwr_tin_6){
    index_1 ("0.013587, 0.040058, 0.133002, 0.310767, 0.587093, 0.975000");
    values ("0.002885, 0.002885, 0.002885, 0.002885, 0.002885, 0.002885");
   }
   rise_power(pwr_tin_6){
    index_1 ("0.012175, 0.038681, 0.131763, 0.309630, 0.586612, 0.975021");
    values ("-0.002528, -0.002529, -0.002530, -0.002532, -0.002535, -0.002540");
   }
  }
 }
.
.
}
```

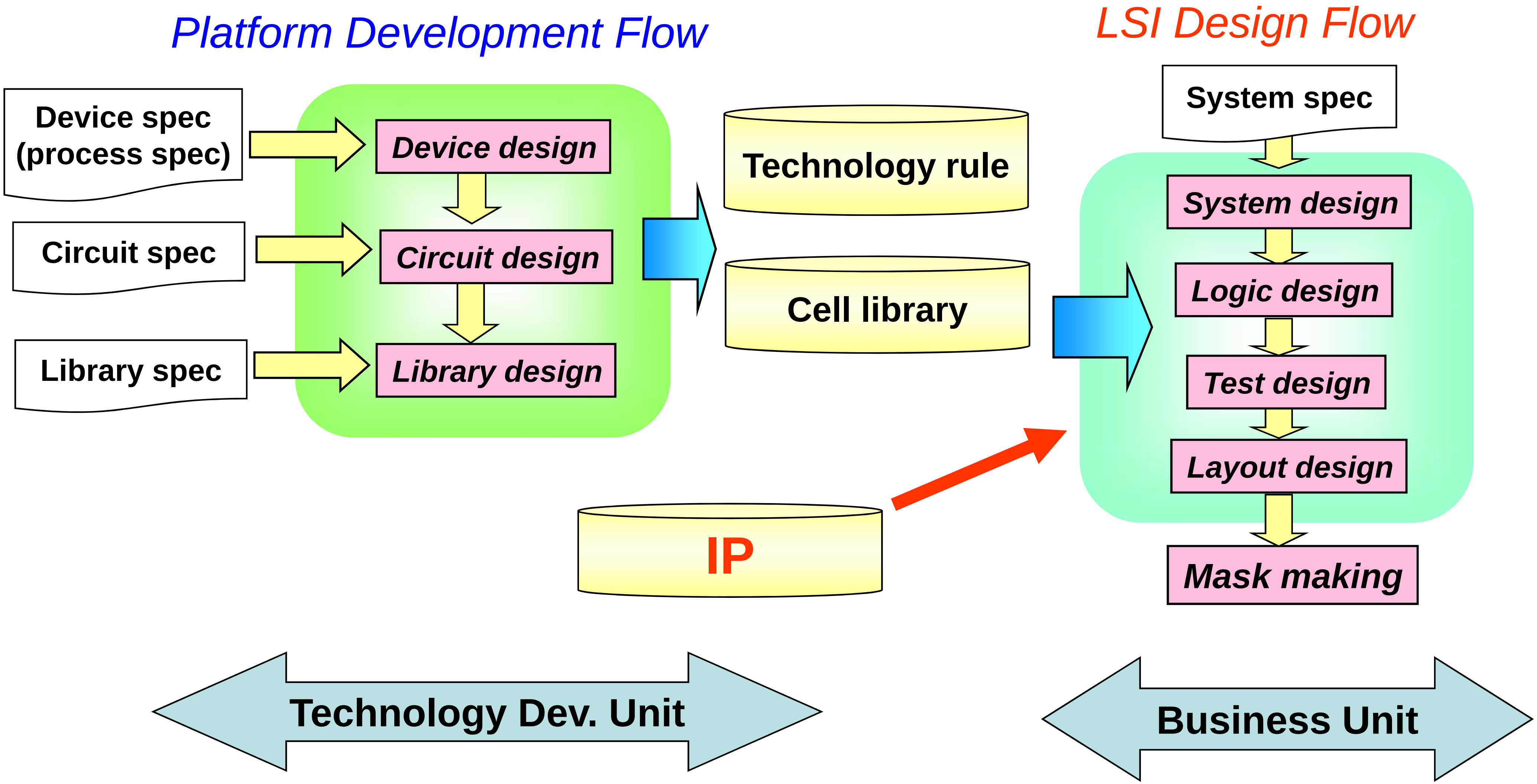
Layout model [LEF]

```
MACRO TMANAND2XT
  CLASS CORE ;
  FOREIGN TMANAND2XT 0.000 0.000 ;
  SIZE 14.760 BY 3.240 ;
  SYMMETRY Y X ;
  ORIGIN 0.0 0.0 ;
  SITE CORE009 ;
  PIN A DIRECTION INPUT ;
       USE SIGNAL ;
        AntennaGateArea 2.492000 LAYER M2 ;
        AntennaPartialMetalArea 0.097200 LAYER M2 ;
        AntennaMaxAreaCAR 0.390048 LAYER M2 ;
        AntennaDiffArea 1 LAYER M2 ;
       PORT
         LAYER M2 ;
         RECT 10.530 1.350 11.070 1.530 ;
       END
  END A
.
.
END TMANAND2XT
```

Total LSI Design Flow

To design LSIs using design environment established by platform development

- Quality of the basic platform determines quality of all products



IPs

IP: commonly used modules, such as

memory module (RAM, ROM, flash memory),

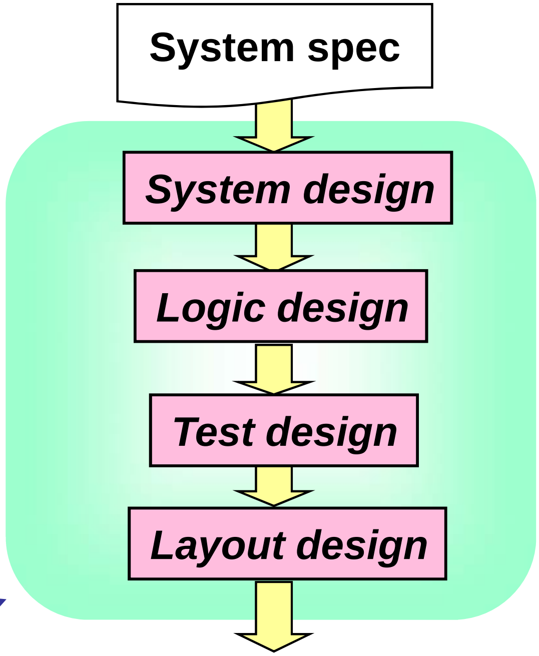
analog module (DAC, ADC, PLL), processing unit (CPU, DSP)

and interfaces(USB, LVDS,

High-speed memory)

RAM: *Random Access Memory*
ROM: *Read Only Memory*
LVDS: *Low Voltage Differential Signaling*

LSI Design Flow



LSI design flow is also adopted in IP designs

Design Methodologies for each LSI device

LSI	Methodology	Design steps and merits	Design TAT	Fab. TAT
Logic	Full custom	Transistor based design for performance optimization	10 months	1~2 months
	Cell based	Combination of pre-designed modules (cells, macros, IPs), logic synthesis, and auto P&R, for large scale integration	2 months	1~2 months
	Gate array	Logic synthesis, and auto P&R, for middle scale integration	1 month	2 weeks
	FPGA	Logic synthesis, and auto routing. Prefabricated.	1 week	(1 day)
Memory	Full custom	Transistor based design for high density memory macro	10 months	1~2 months
Analog	Full custom	Transistor based design for high performance and precision	6 months	1~2 months
Mixed Signal	Full custom/ cell based	Co-verification for analog and digital	8 months	1~2 months

TAT: Turn Around Time

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

    Platform Development

    LSI Design Flow

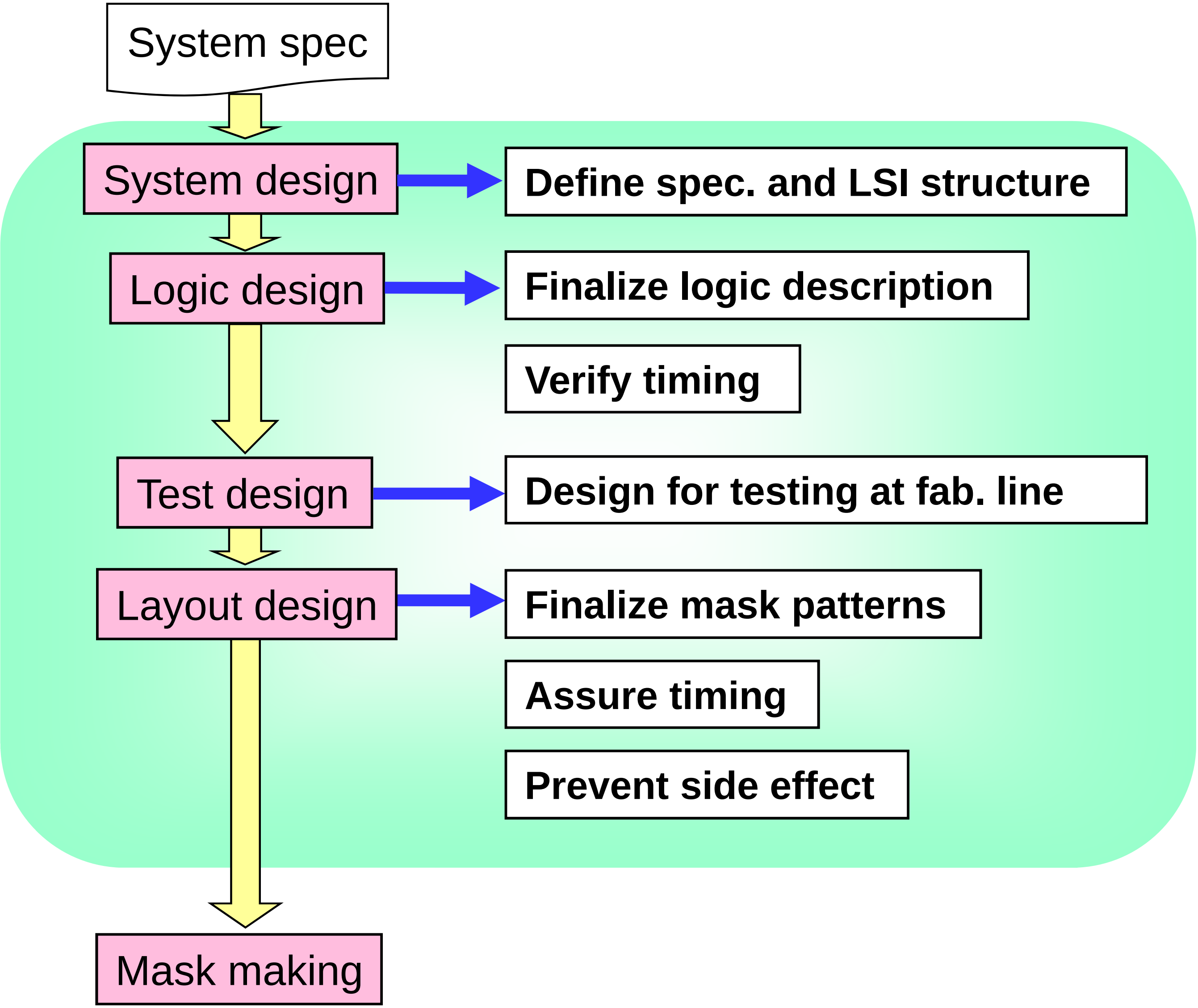
        System Design

        Logic Design

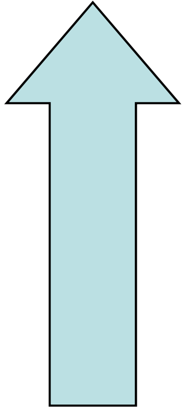
        Test Design

        Layout Design

Before Learning Details



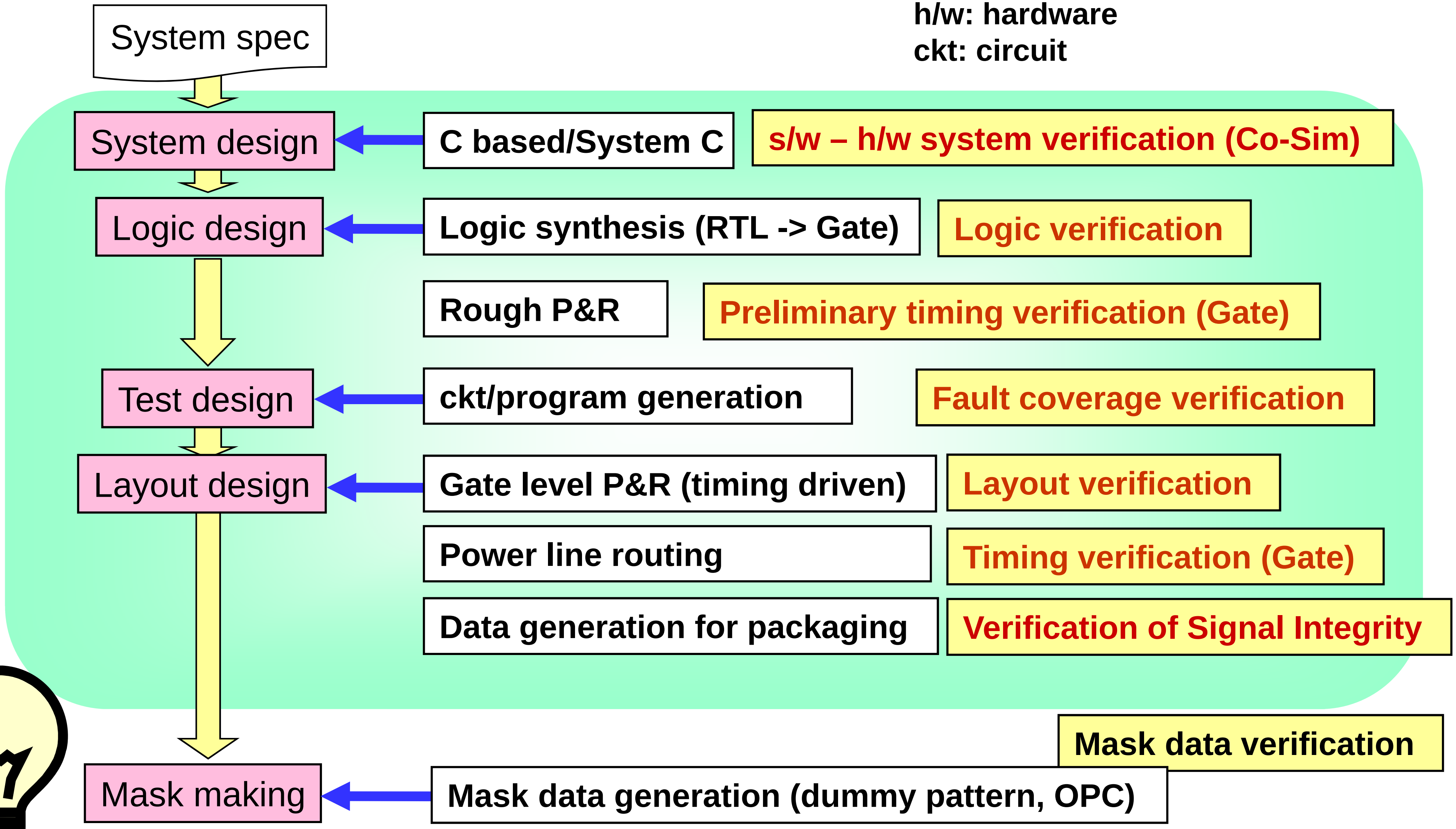
- Design input & output
- Design review
- Documentation



Minimize
Design Errors

LSI Design Flow

RTL: Register Transfer Level
P&R: Place & Route
OPC: Optical Proximity Correction
s/w: software
h/w: hardware
ckt: circuit



System spec

System design ← C based/System C — s/w – h/w system verification (Co-Sim)

Logic design ← Logic synthesis (RTL -> Gate) — Logic verification

Rough P&R — Preliminary timing verification (Gate)

Test design ← ckt/program generation — Fault coverage verification

Layout design ← Gate level P&R (timing driven) — Layout verification

Power line routing — Timing verification (Gate)

Data generation for packaging — Verification of Signal Integrity

Mask making ← Mask data generation (dummy pattern, OPC) — Mask data verification

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

    - Platform Development

    - LSI Design Flow

        - System Design

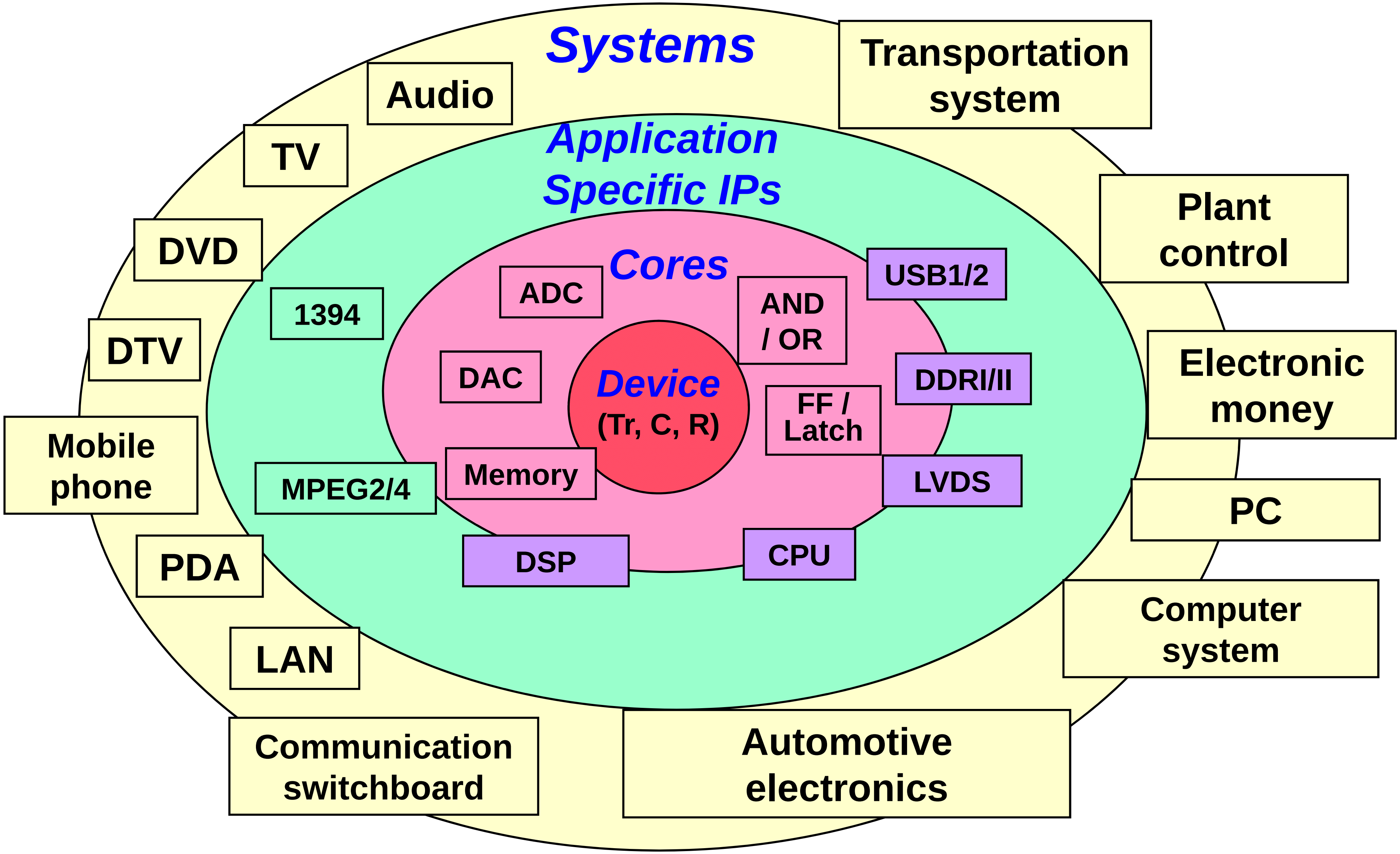
        - Logic Design

        - Test Design

        - Layout Design

System Design

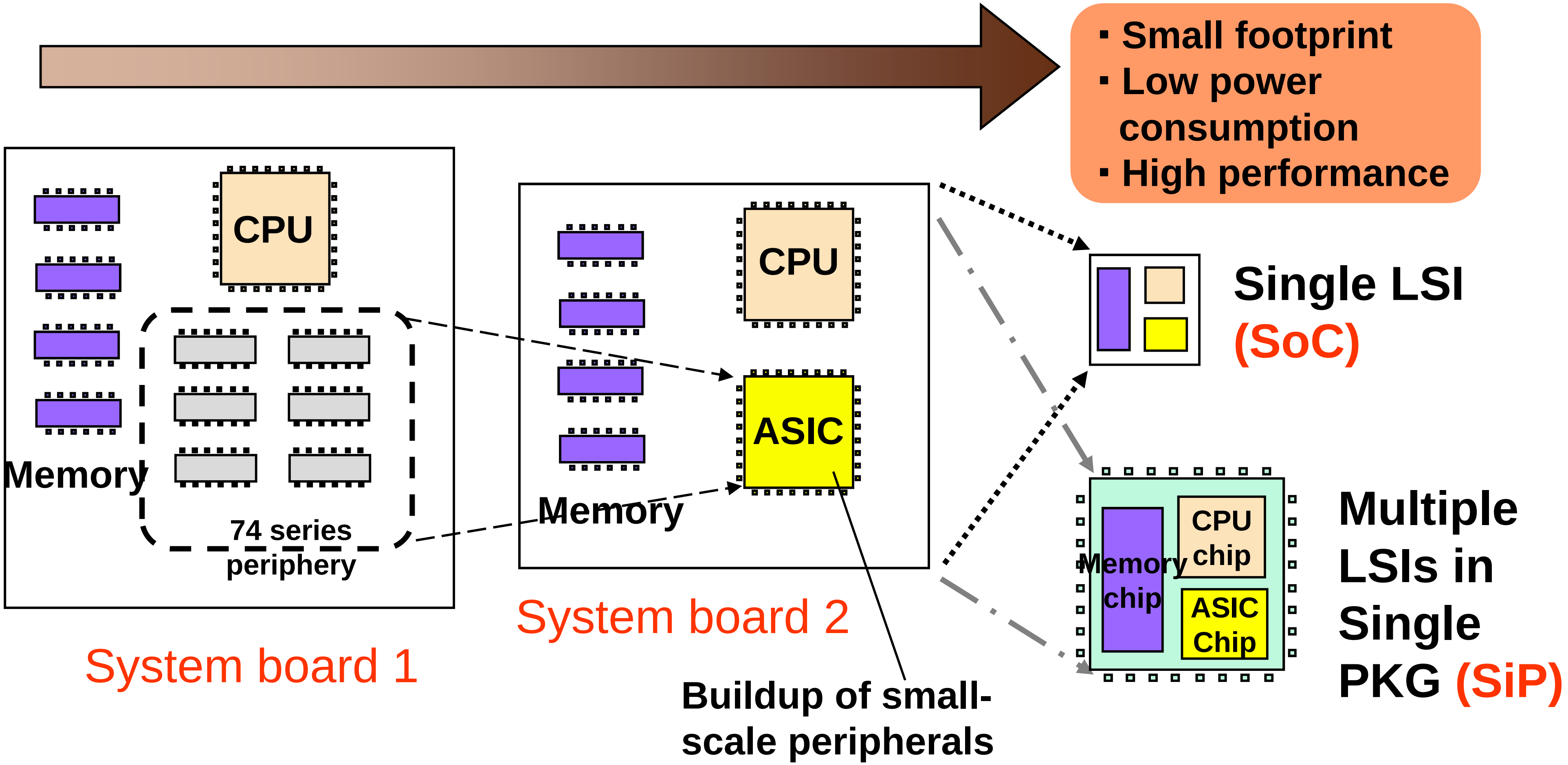
Systems & Components



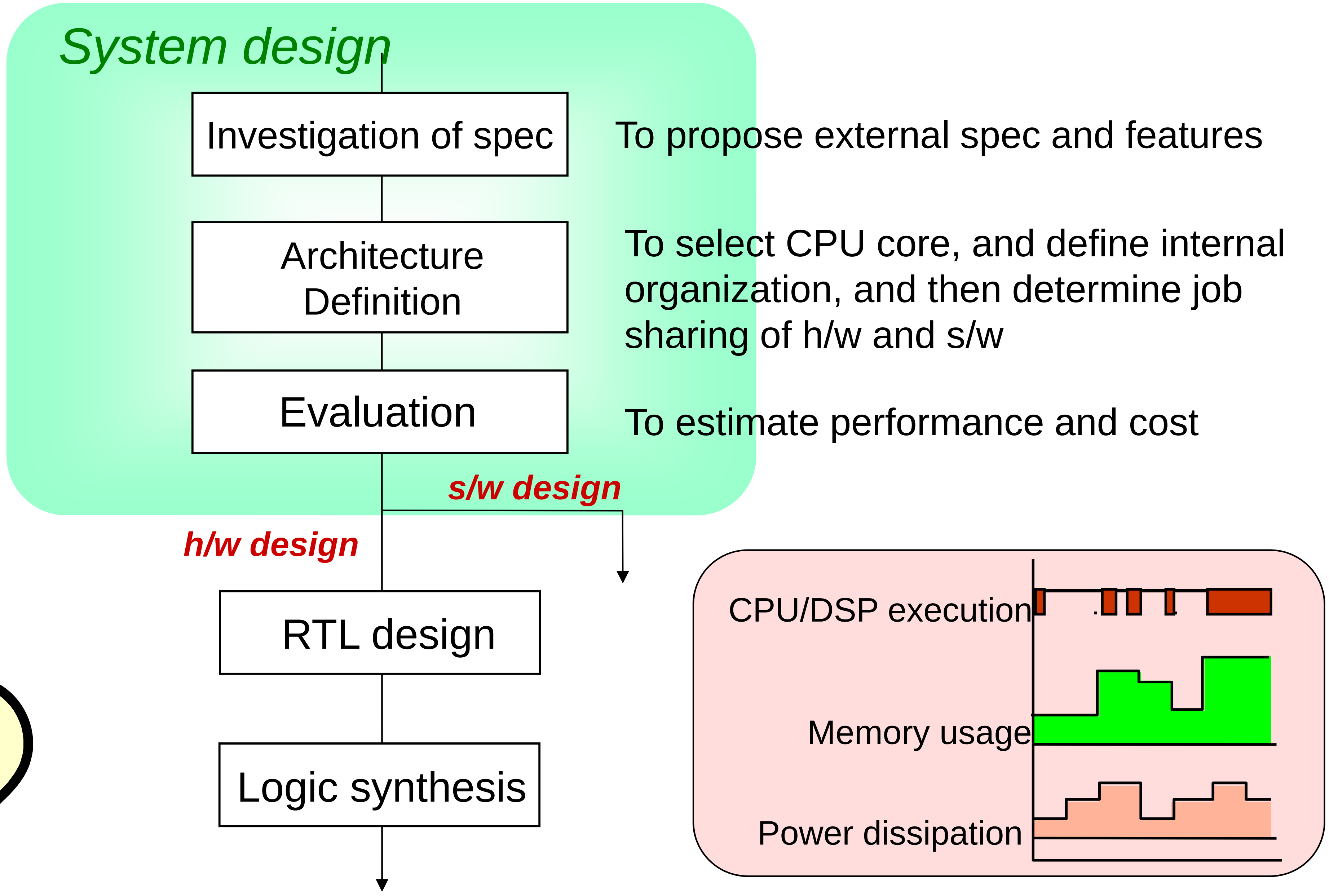
System Integration

SoC (System-on-Chip): System implementation in single LSI

SiP(System-in-Package): System implementation in 1 package



System Design Flow



To propose external spec and features

To select CPU core, and define internal organization, and then determine job sharing of h/w and s/w

To estimate performance and cost

Requirements for System Design

- To optimize LSI specifications by watching market trend and hearing customers' needs, such as performance, features, power dissipation, reliability, price (target die size, package selection, process)

- To remove over-spec and spec bugs

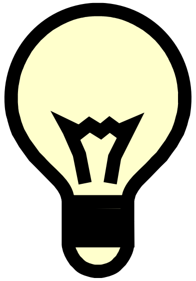
- To avoid specification changes during h/w design phase because h/w design becomes critical:
  - demand for time-to-market is increasing
  - complexity is increasing
  - need for performance is increasing
  - fabrication cost is increasing (more than $1M for one maskset)

System Design Trend

- Conventional design
  - Inspiration and experience
  - Excess margin due to paper plans

- Current trend
  - C/System C based design
  - Co-simulation
  - Early prototype [Logic Bench]
  - Emulator [Celaro]

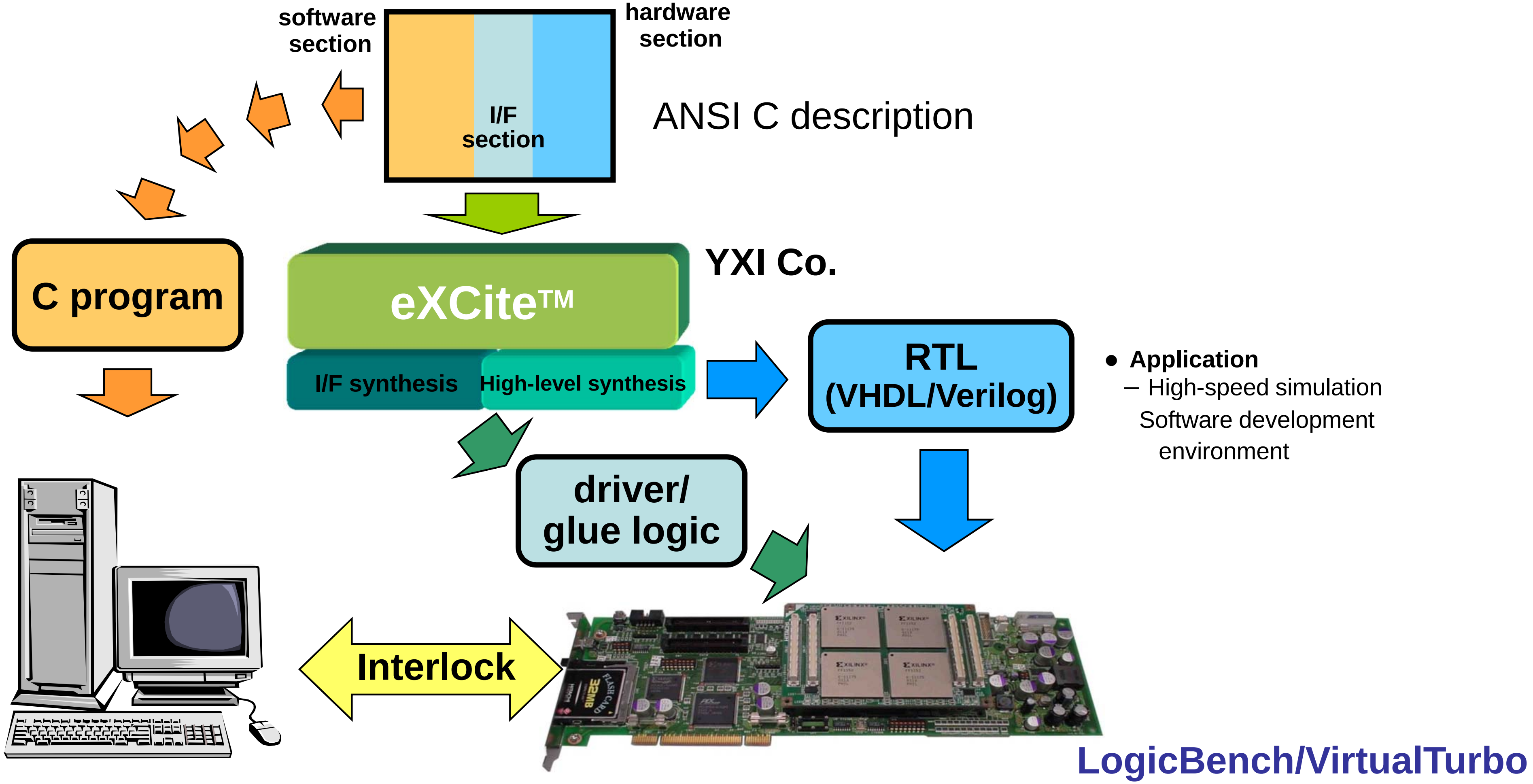
BH003



Early Prototype

- Functional description (algorithm) in ANSI C is converted to form FPGA based early prototype.

  ⇒ platform for software development, and system verification prior to initiation of RTL design



- **Application**
  - High-speed simulation Software development environment

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

   Platform Development

   LSI Design Fflow

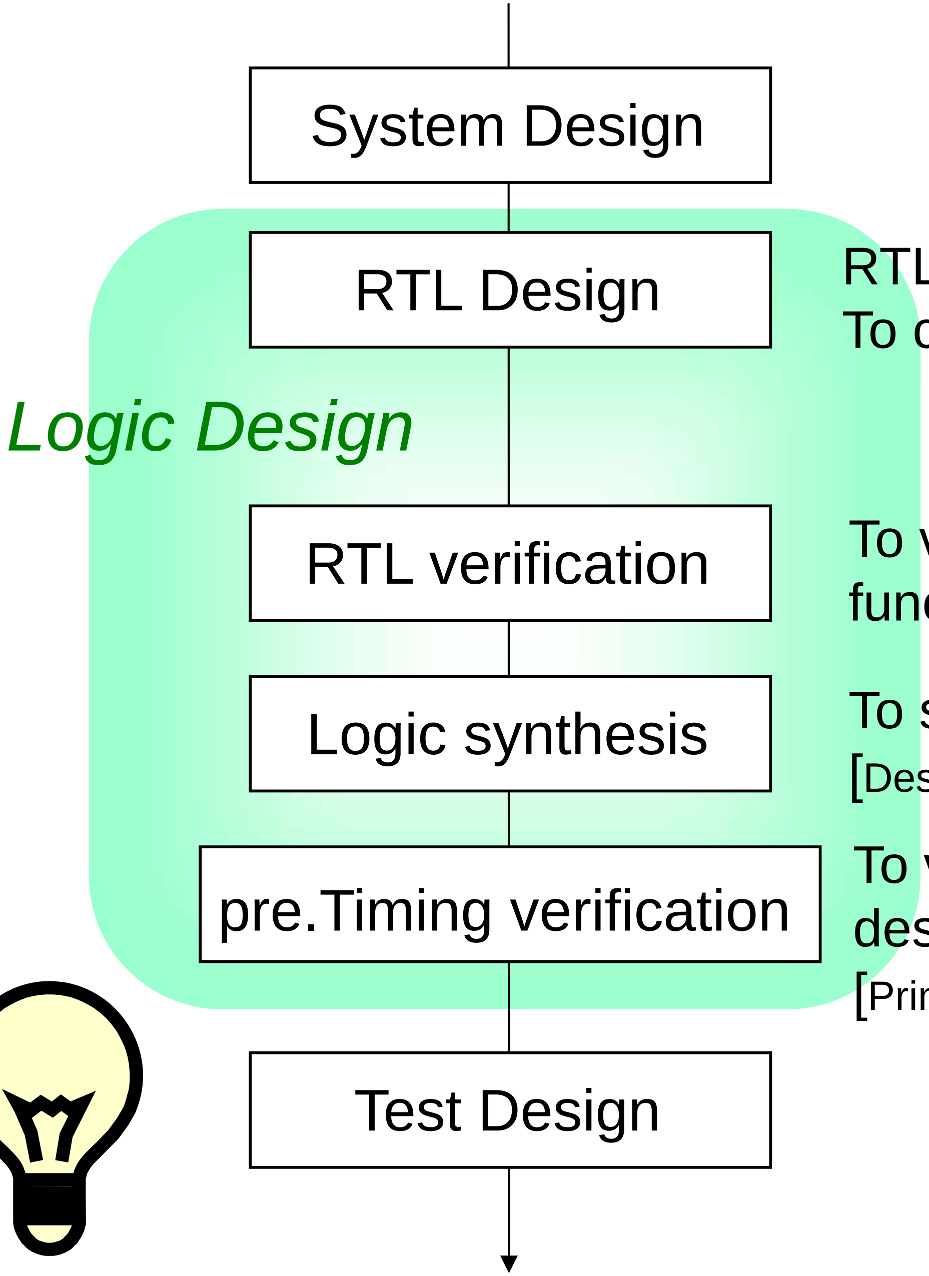
      System Design

      Logic Design

      Test Design

      Layout Design

Logic Design



System Design

RTL Design

Logic Design

RTL verification

Logic synthesis

pre.Timing verification

Test Design

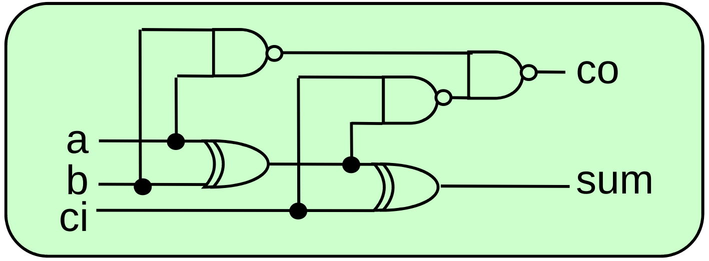
RTL description [VerilogHDL/VHDL]
To check RTL code

```
assign  sum = a ^ b ^ ci;
assign  co  = (a&b)|(a&ci)|(b&ci);
```

To verify RTL description is compliant with functional spec [NC-Verilog/VCS/ModelSim]

To synthesize gate level description from RTL [DesignCompiler]

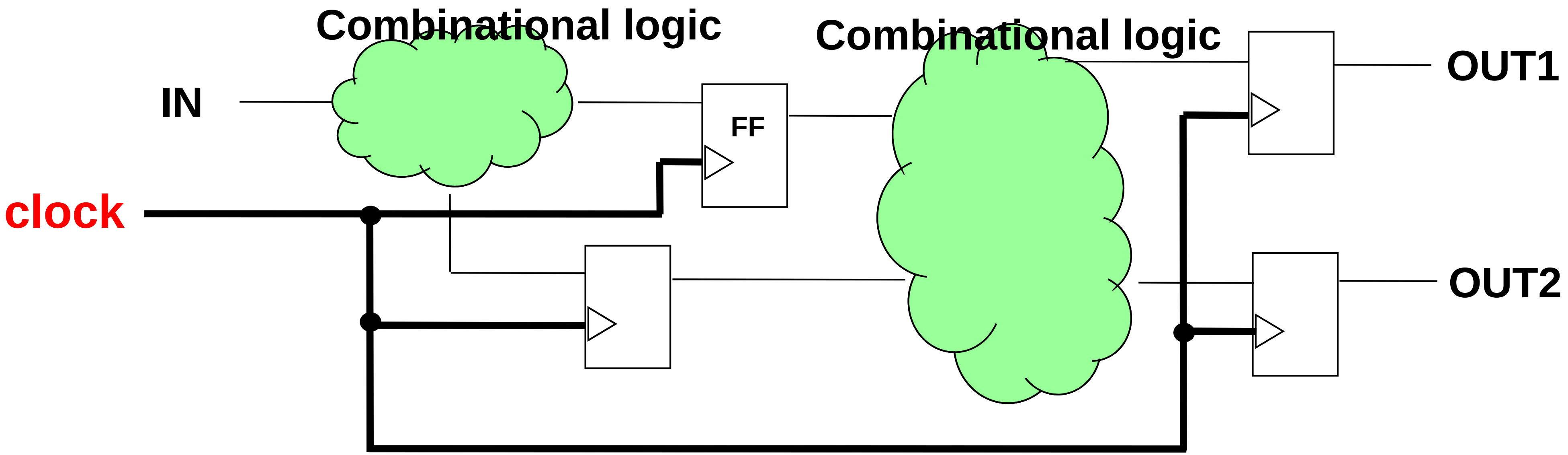
To verify operation speed of gate level description using pre-layout load conditions [PrimeTime]



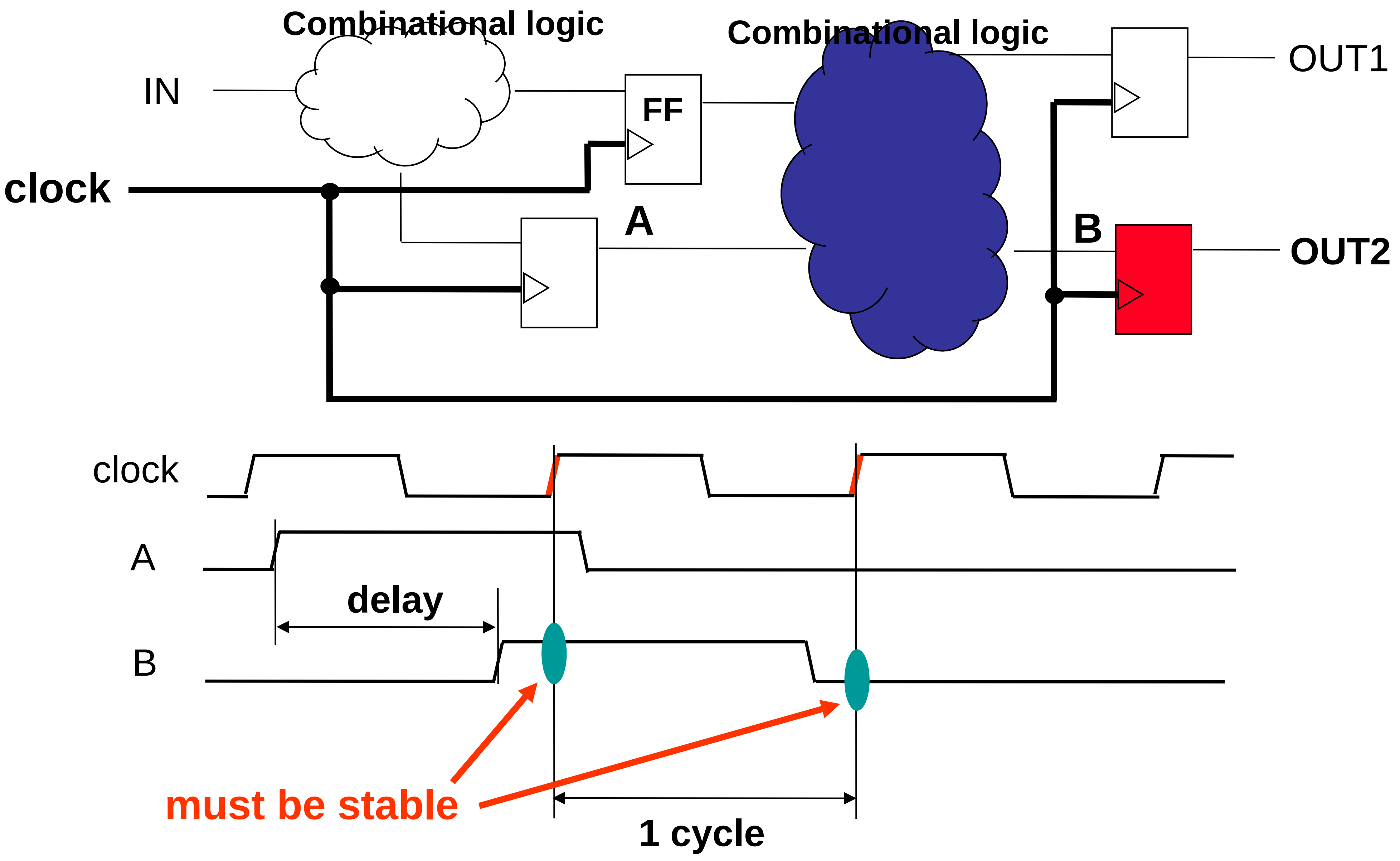
Synchronous Design

All operations of one circuit block of interest are uniquely defined in synchronous design with base clock signal.

RTL design …fully synthesizable design, configured with single clock signal and registers (FF)



One cycle operation



RTL Design

- RTL design enables simple descriptions: less human errors
- Hardware Description Language (HDL) is used in RTL design, such as Verilog_HDL, VHDL

Description example of multiplexer (Verilog_HDL):

```
module  mux (d1, d2, sel, mout);
input  d1;       // data-1
input  d2;       // data-2
input  sel;      // select
output  mout;    // selected data
reg     mout;

always @(d1 or d2 or sel) begin
  if (sel == 1'b1)  mout = d1;
  else              mout = d2;
end

endmodule  // mux
```

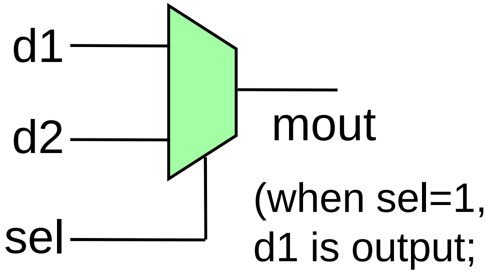
module definition

terminals

functions

end of module definition

multiplexer
（selector）



(when sel=1, d1 is output; while sel=0, d2)

RTL Code Check

The early bird catches the worm!

Purpose

- To improve RTL quality for shortening design period before starting its functional verification

Procedure

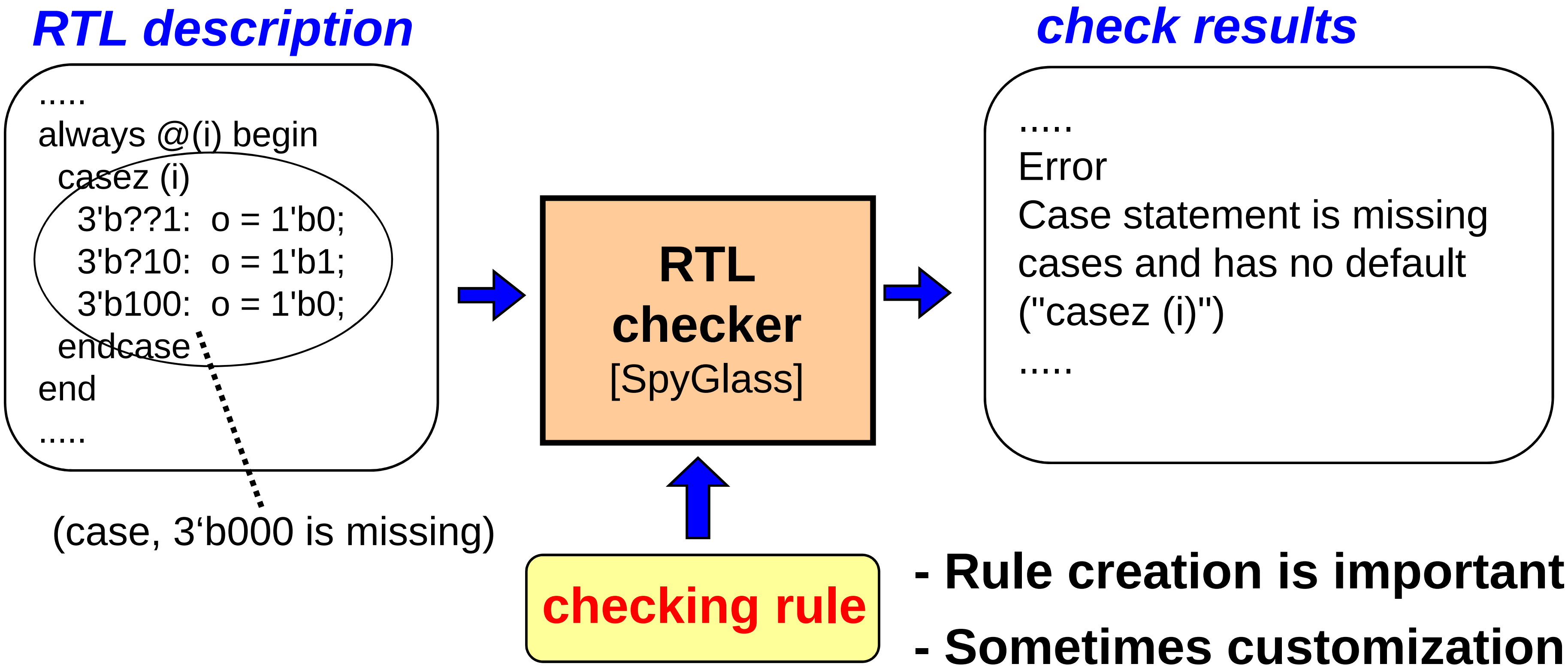
- Naming convention check
- Grammar check
- Inconsistency check
- Structural analysis

Advantageous Effects

- Smooth transfer to succeeding design steps (functional simulation, timing analysis, DFT tools…)
- Reusability of IPs developed by independent teams

RTL Checker

-> Pinpointing structural coding and consistency problems in early stage can shorten the total design period.



RTL description

```
.....
always @(i) begin
  casez (i)
    3'b??1:  o = 1'b0;
    3'b?10:  o = 1'b1;
    3'b100:  o = 1'b0;
  endcase
end
.....
```

(case, 3‘b000 is missing)

RTL checker
[SpyGlass]

checking rule

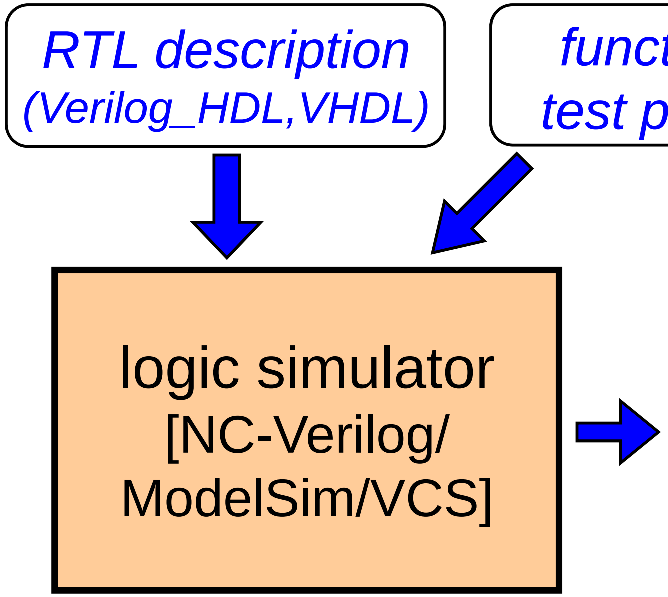
check results

```
.....
Error
Case statement is missing
cases and has no default
("casez (i)")
.....
```

- **Rule creation is important**
- **Sometimes customization is needed**

RTL Verification (1) - functional logic simulation -

To check expected simulation output by applying functional test patterns to RTL description

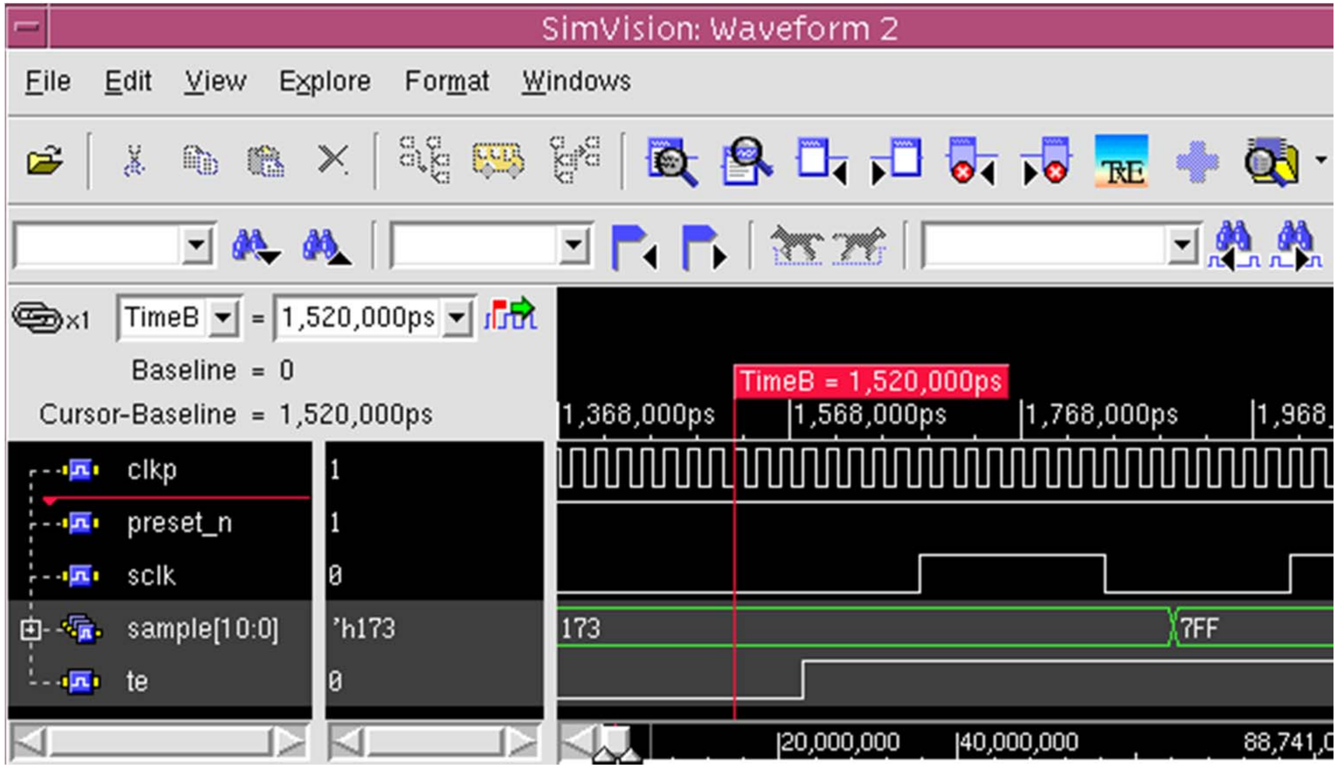


RTL description
(Verilog_HDL,VHDL)

functional test pattern

logic simulator
[NC-Verilog/ ModelSim/VCS]

Output waveform (visual confirmation)



TimeB = 1,520,000ps

Baseline = 0

Cursor-Baseline = 1,520,000ps

TimeB = 1,520,000ps

clkp	1	
preset_n	1	
sclk	0	
sample[10:0]	'h173	173 7FF
te	0	

RTL Verification (2)

➢ Random Test Generation (RTG)

- Quasi random patterns enable exhaustive combinational tests
- Reinforcing complex, time consuming logic simulation
- Reference simulator, and constraints file are required

➢ Assertion Based Verification (ABV)

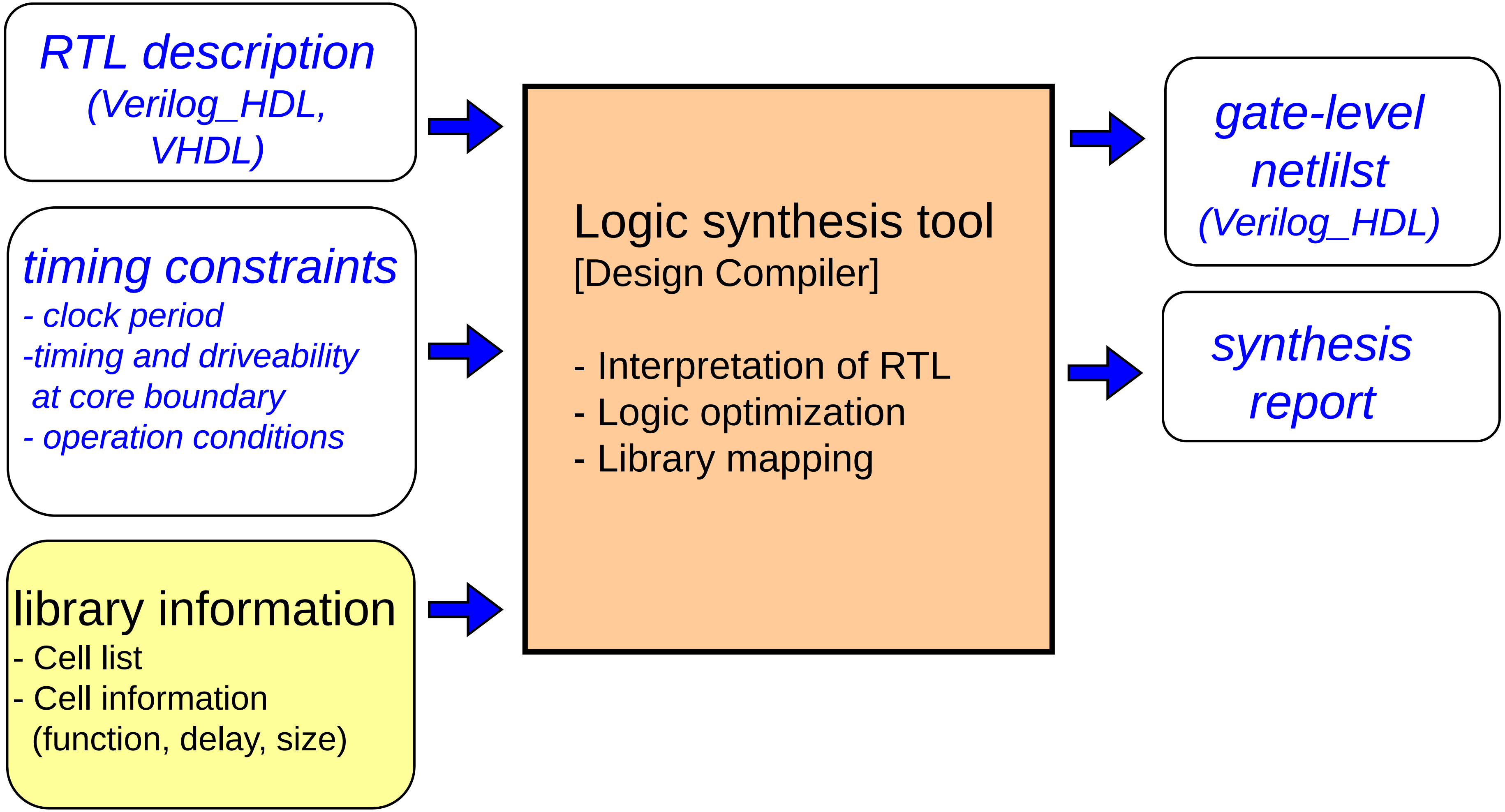
- ABV eliminates visual confirmation miss in logic simulation
- Assertion (prerequisites for functional operation) is embedded in RTL description
- Assertion resembles comment in a statement line

➢ Coverage Measurement

- Code coverage for RTL exhaustiveness check on line-by-line basis: whether such line is executed or not
- Function coverage for occurrence check of specific events: event definitions are required

Logic Synthesis

Synthesis: RTL description is converted to gate-level description.



RTL description
(Verilog_HDL, VHDL)

timing constraints
- clock period
-timing and driveability at core boundary
- operation conditions

library information
- Cell list
- Cell information (function, delay, size)

Logic synthesis tool
[Design Compiler]

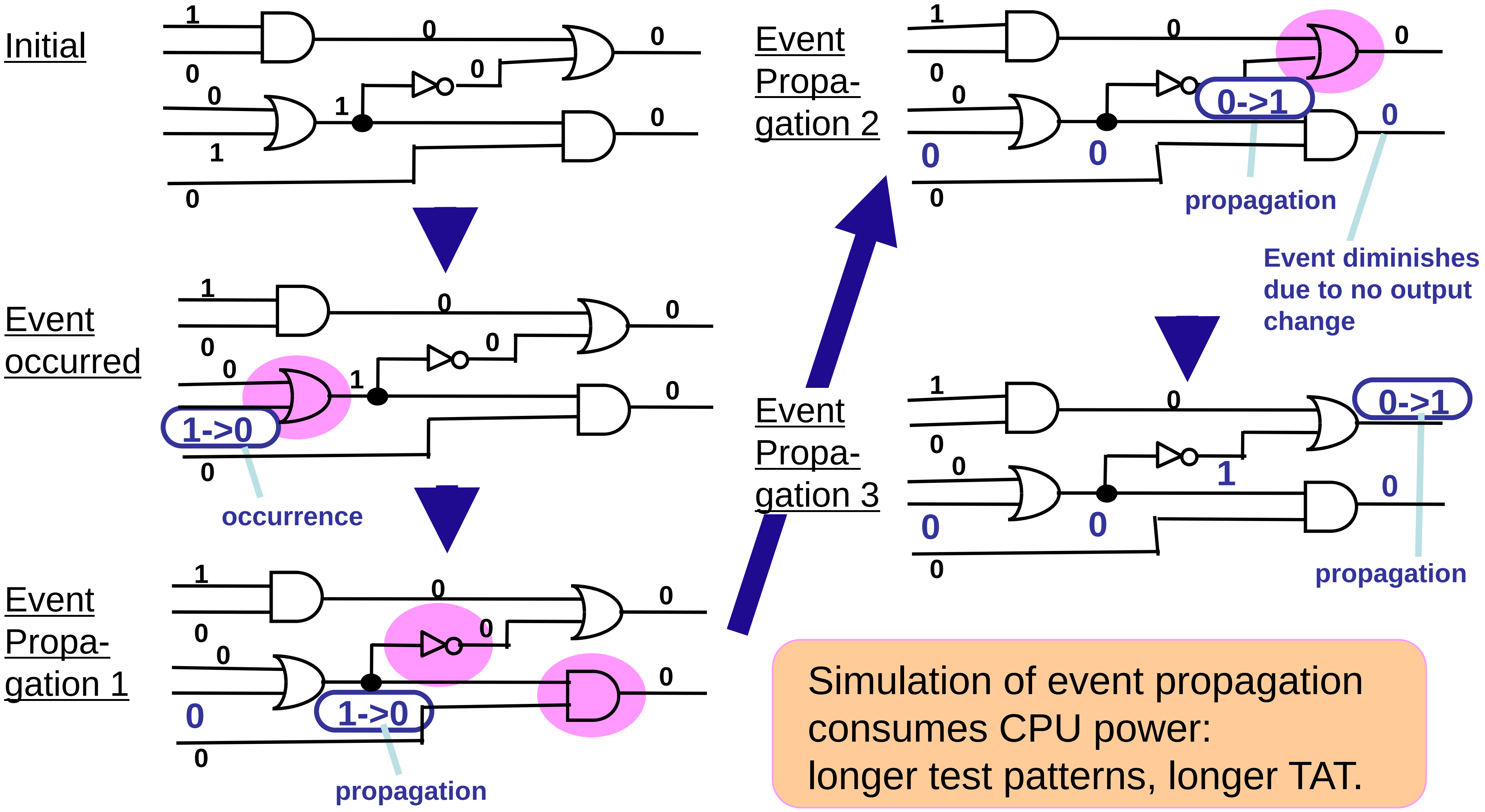
- Interpretation of RTL
- Logic optimization
- Library mapping

gate-level netlilst
(Verilog_HDL)

synthesis report

Logic Verification (1) - functional simulation

Logic simulation: event driven method by test patterns



Initial

Event occurred

Event Propagation 1

Event Propagation 2

Event Propagation 3

occurrence

propagation

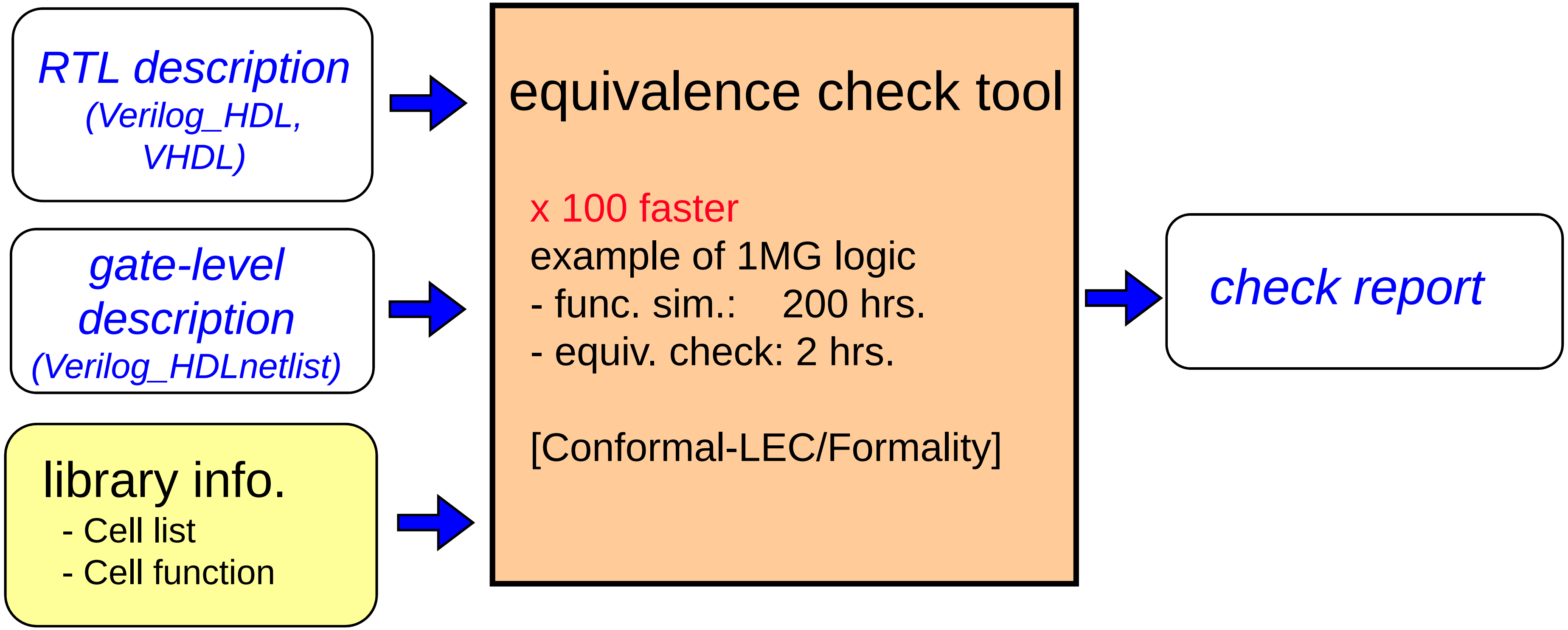
Event diminishes due to no output change

Simulation of event propagation consumes CPU power: longer test patterns, longer TAT.

Logic Verification (2)

- equivalence check

- Equivalence of two logic descriptions is mathematically verified by formality check： no need for test patterns
- Equivalence check is exhaustive and faster than functional simulation



RTL description
(Verilog_HDL, VHDL)

gate-level description
(Verilog_HDLnetlist)

library info.
- Cell list
- Cell function

equivalence check tool

x 100 faster
example of 1MG logic
- func. sim.: 200 hrs.
- equiv. check: 2 hrs.

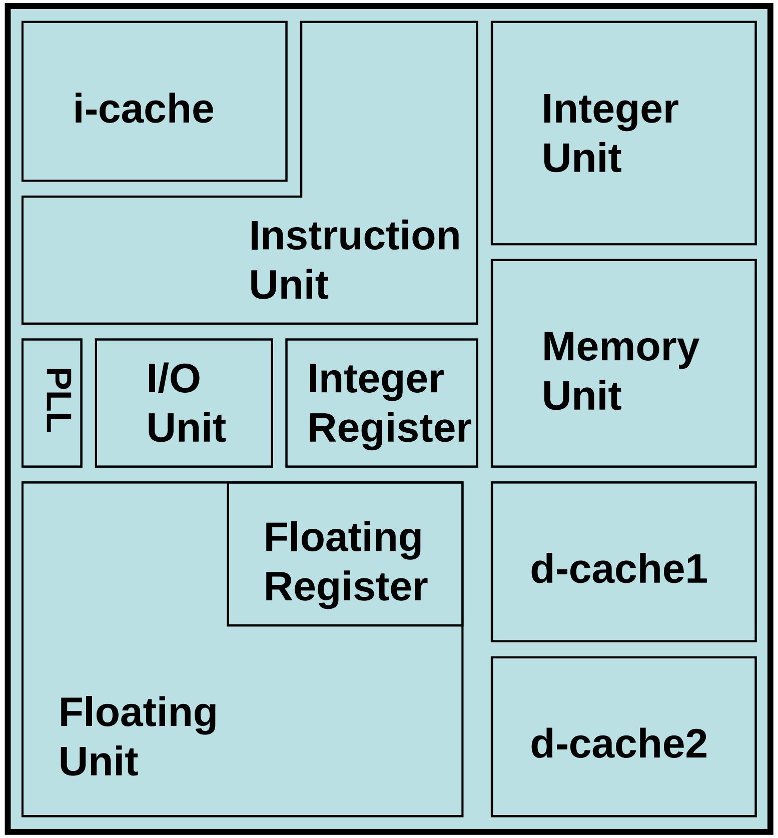
[Conformal-LEC/Formality]

check report

Floor Planning (1)

- for pre-layout timing verification

- Place functional blocks by considering signal flow
- Consider package specifications, minimize pin count, and shorten wiring length
- Bonding pad layout must be well considered



Example for floor plan of processor chip

Timing Verification

Pre-layout verification

➢ STA: Static Timing Analysis

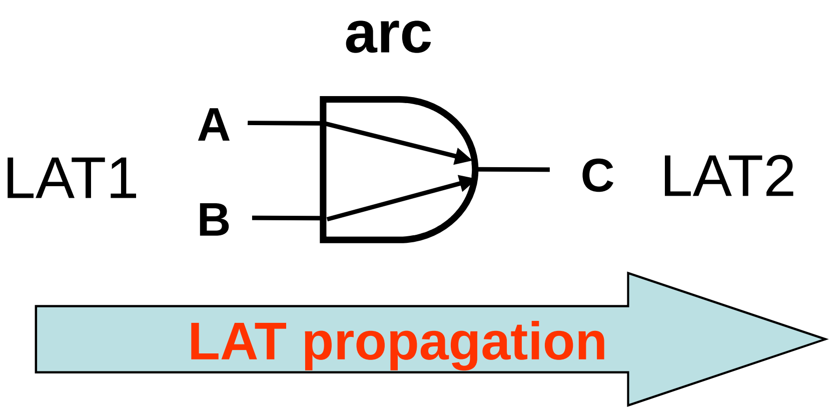
- Structure of gate-level logic description is analyzed and then compared with timing constraints
- Test patterns are not required
- Synchronous paths can be 100% verified
- Asynchronous logic and other special circuit cannot be analyzed
- Special paths must be specified manually

➢ DTA: Dynamic Timing Analysis

- Logic simulation handling delay timing is executed
- Verification of all paths is unrealistic because huge test patterns are required

-> Verify synchronous paths with STA, and use DTA for limited paths that cannot be checked by STA. Always do synchronous design except for unavoidable case!

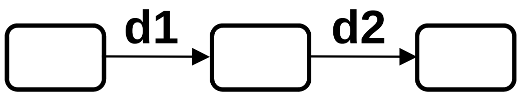
STA Basic Algorithm



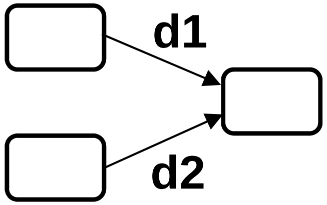
STA tool propagates LAT by selecting the maximum of LATs and adding delay of ARC to the LAT

LAT: Latest Arrival Time

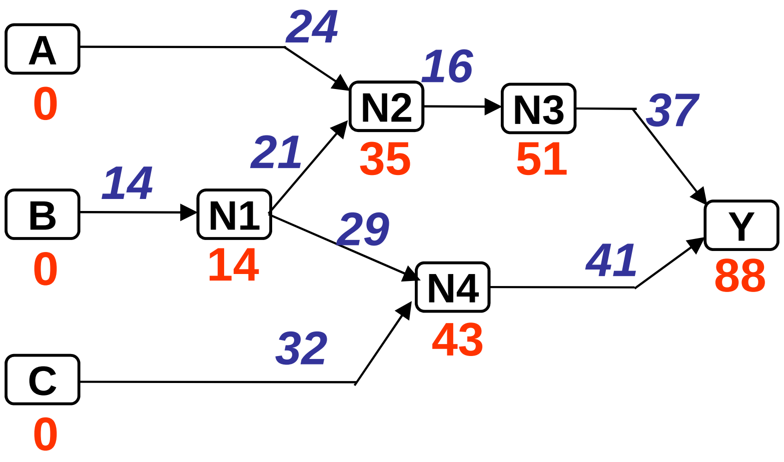
sum : d1 + d2



max : max (d1, d2)



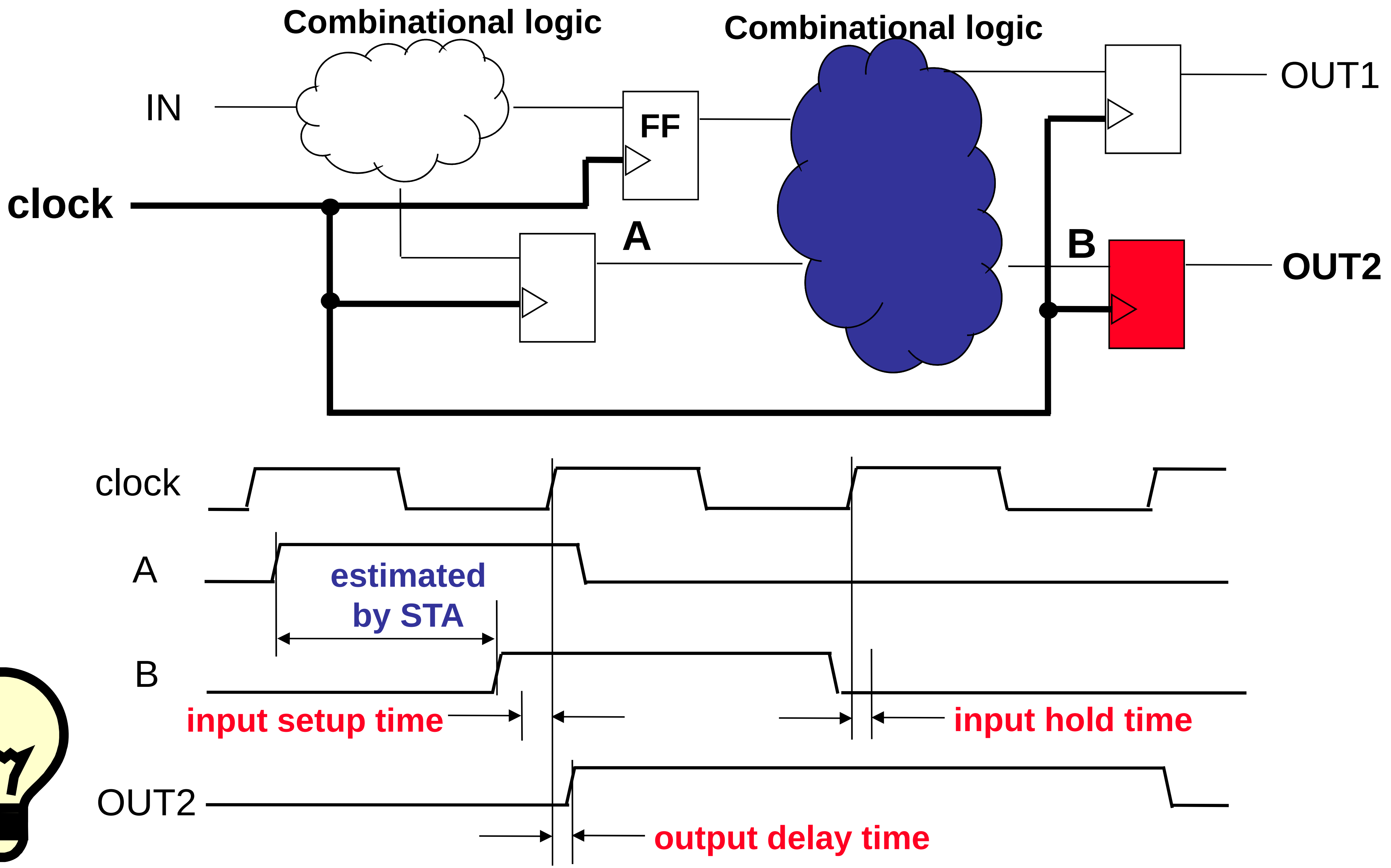
example



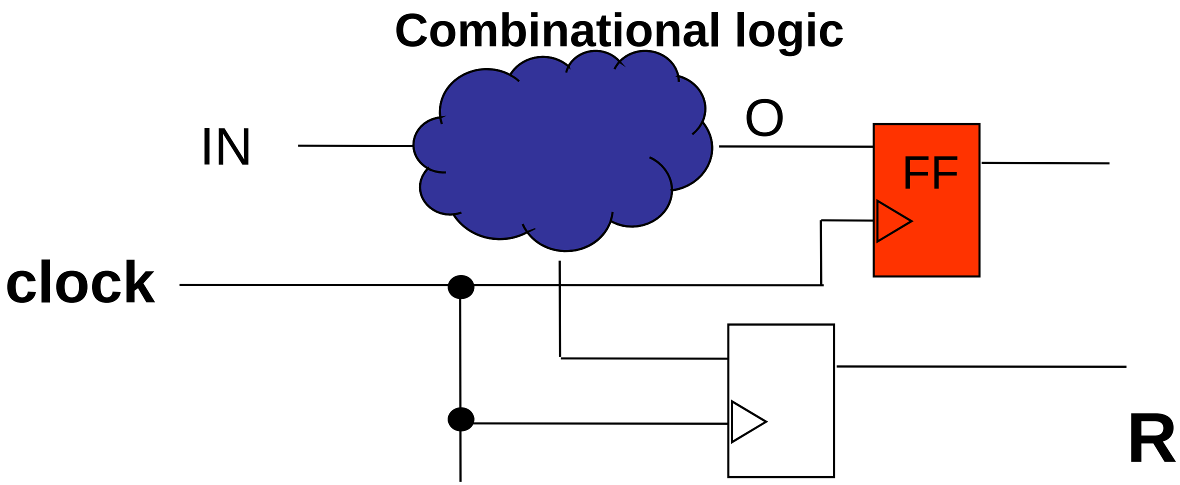
blue italic: gate delay

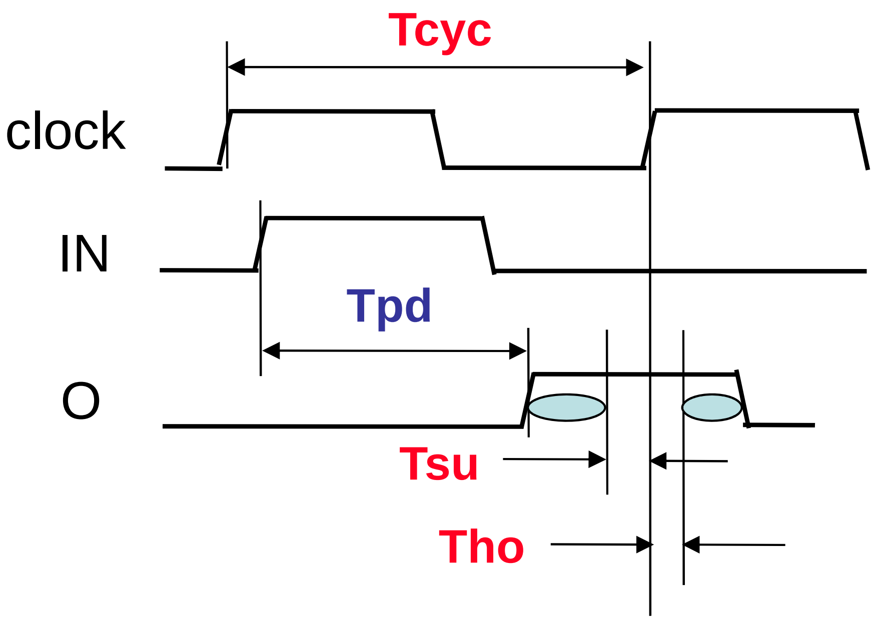
red: LAT

Synchronous Design



Timing Requirements





Requirements

Tcyc – Tsu > Tpd > Tho

⬭ : timing margin

timing specifications

- **Tcyc: clock cycle time**
- **Tsu: input setup time**
- **Tho: input hold time**

estimated by STA

- **Tpd: propagation delay time**

Comparison of STA and DTA

	STA	DTA
analyzing objects	synthesizable, synchronous circuit (any combinational circuit)	any logic circuit
asynchronous circuit	NG	OK
special circuit, such as mixed signal	NG	OK for logic portion
special paths	need timing exception info.	not necessary
analyzing method	structure analysis of gate level logic and comparison with timing constraints	event driven, logic simulation
tools	timing analysis tool (ex. Synopsys Prime Time)	logic simulation tool (ex. Synopsys VCS, Cadence Ncsim)
input	gate level netlist R/C parasitic information (net) cell library (timing information) constraint file	gate level netlist delay information (net/cell) cell library (functional info.) test patterns
output	timing analysis result (violated path list, slack value)	simulation result (pass/fail, simulation wave form)
analysis coverage	100%	depending upon test patterns
execution time	short	long (depending on test patterns)

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

    Platform Development

    LSI Design Flow

        System Design

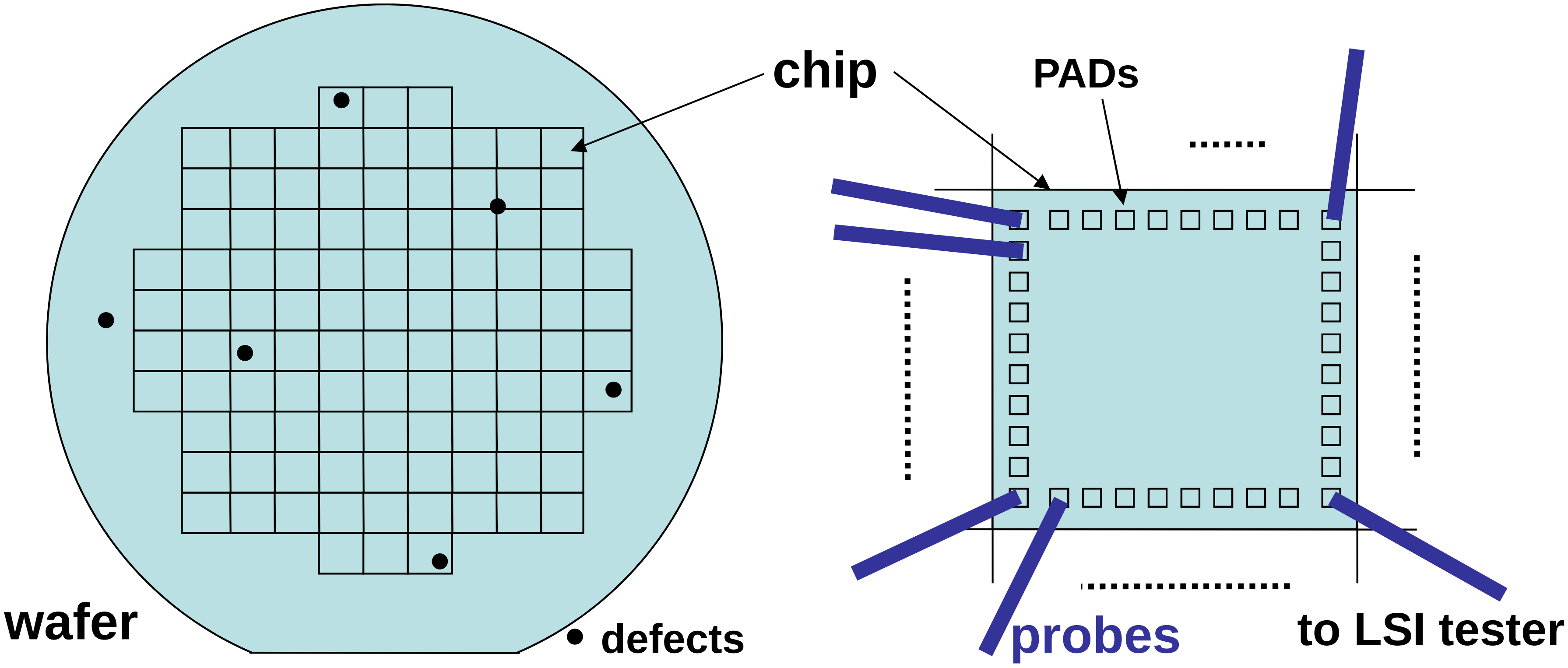
        Logic Design

        Test Design

        Layout Design

What is Test Design?

The purpose of Test Design is to establish testing environment to be used after wafers are fabricated. Fabricated wafers contain defects at several rates. Defective chips must be screened by an LSI tester before being diced and packaged



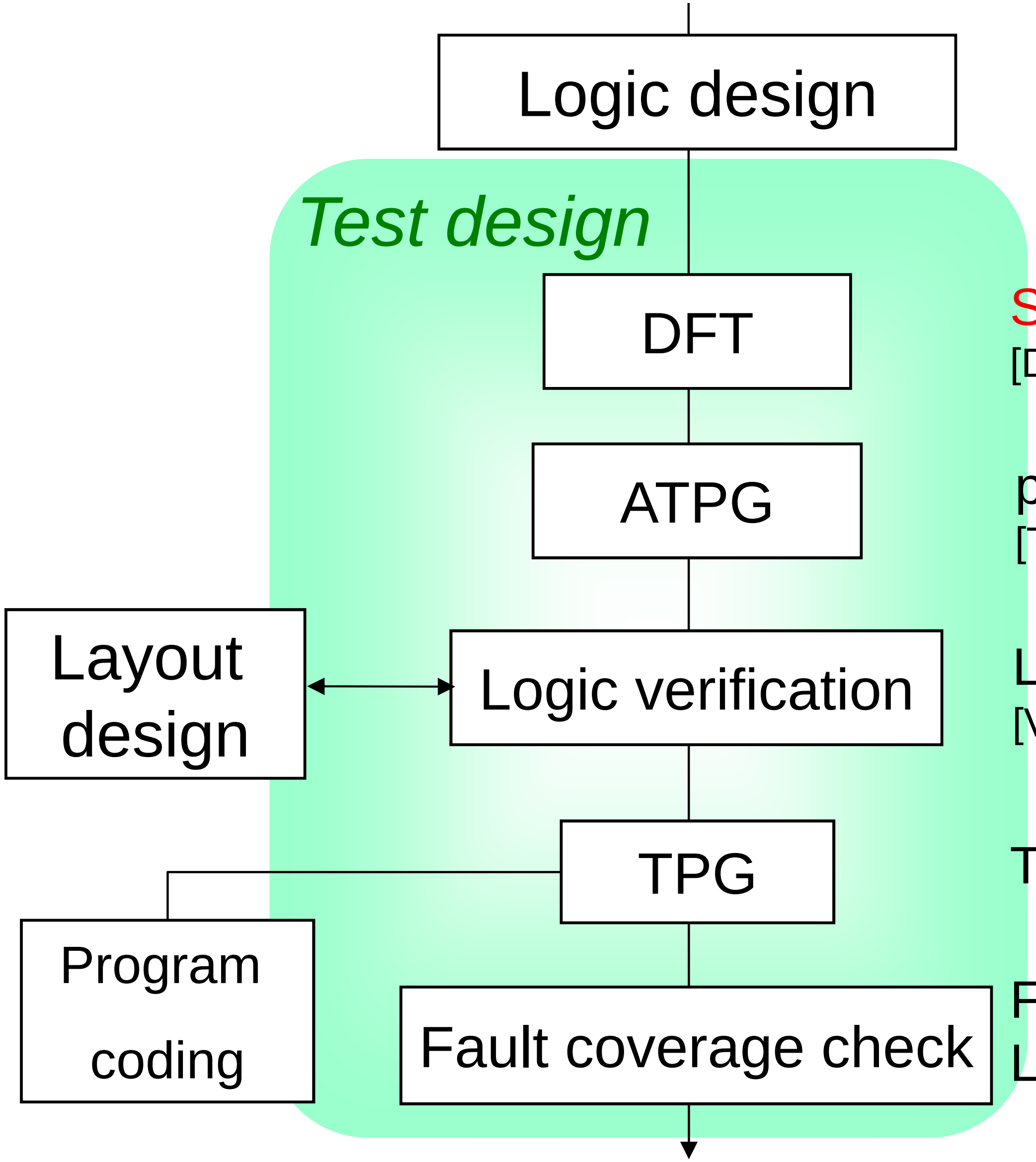
The LSI tester is also used to screen defective LSIs damaged during packaging

Test Design

DFT (Design for Testability)

ATPG (Automatic Test Pattern Generation)

BIST (Built In Self Test)



Logic design

Test design

DFT

ATPG

Layout design

Logic verification

TPG

Program coding

Fault coverage check

SCAN/BIST insertion
[DFTAdviser/DFTCompiler/Renesas inhouse]

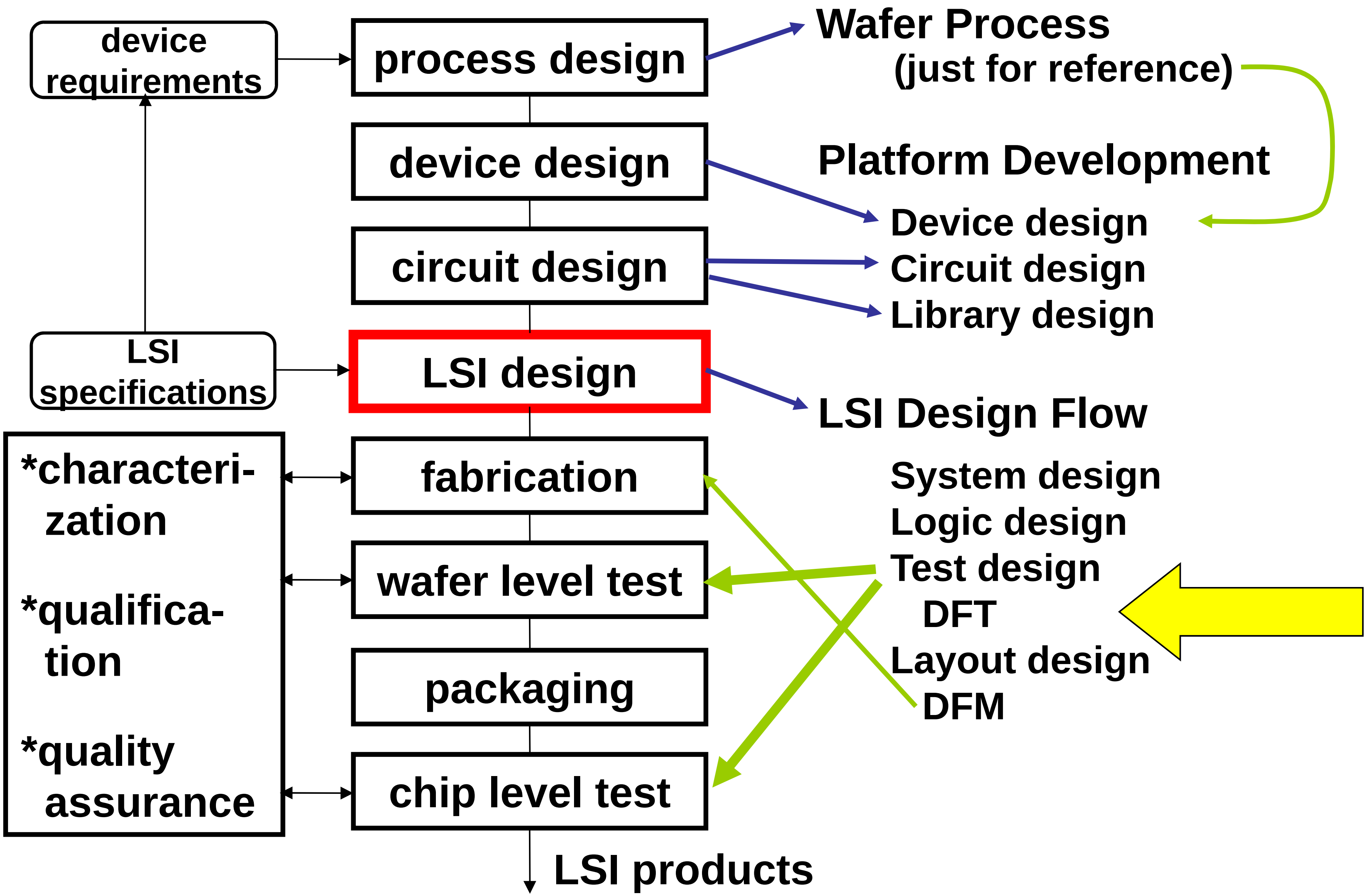
pattern for SCAN/BIST
[TetraMAX/FastScan/Renesas inhouse]

Logic verification including test circuit
[Verilog_XL/NC-Verilog/Conformal-LEC]

Test pattern generation for LSI tester

Fault coverage check with
LSI tester program [Z01X!/Verifault-XL]

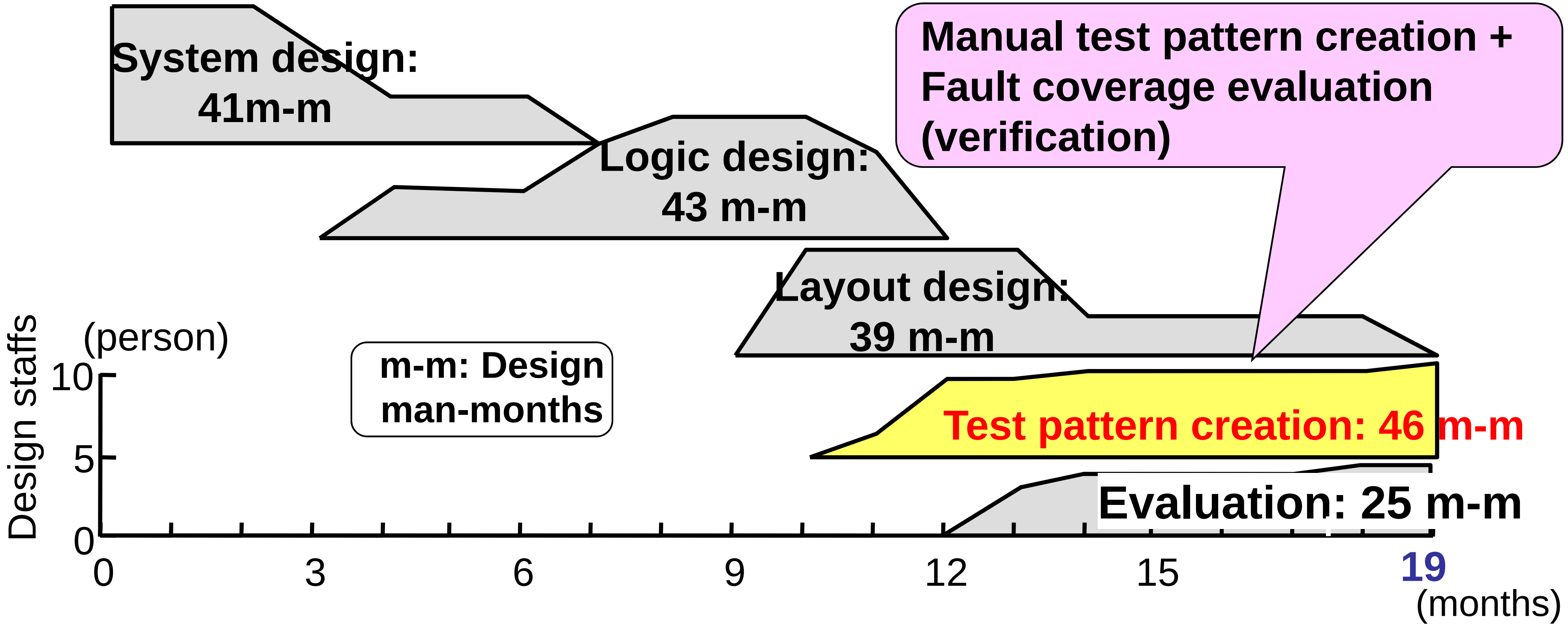
DFT in LSI Development



DFT (1)

- to solve manpower problem

Example of SH7040 development: totally 19 man-months

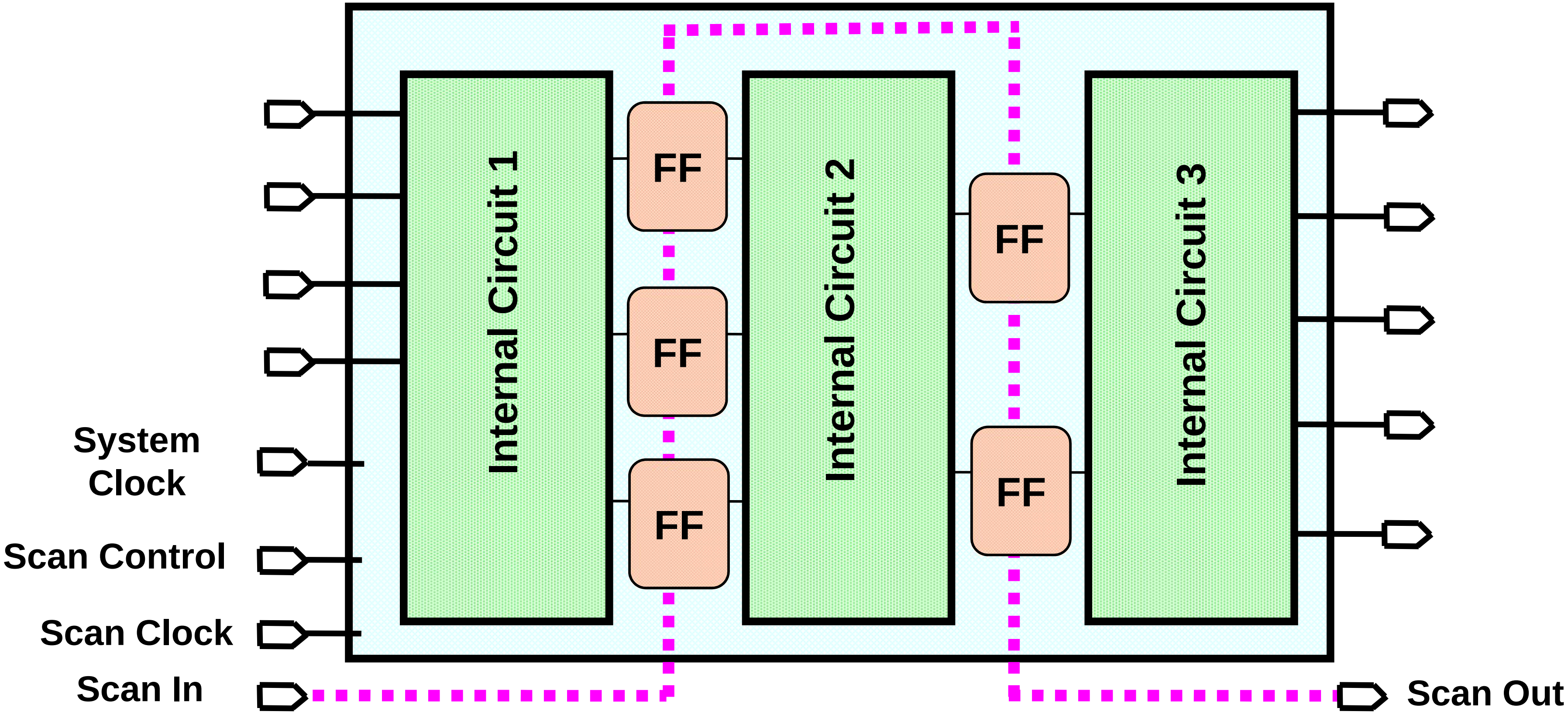


→ SCAN and ATPG techniques were then introduced

SCAN

<Testability improvement by SCAN>

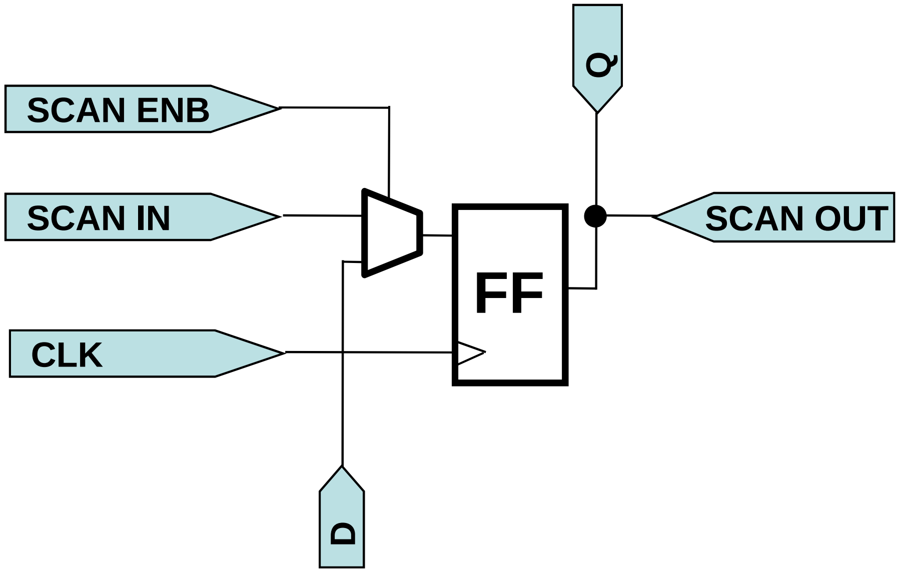
FF=Flip flop



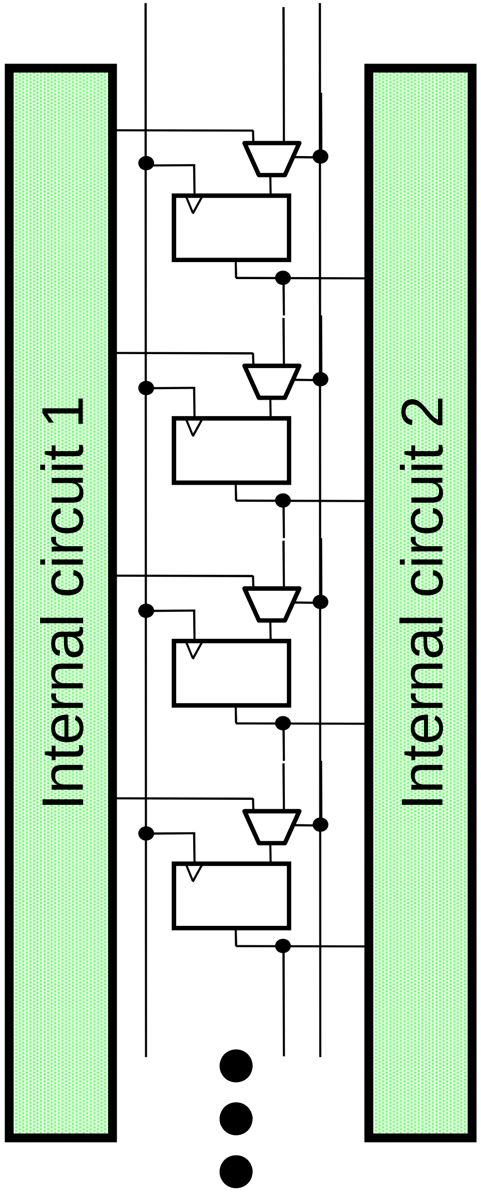
◆ Internal circuits become controllable and observable by applying test signals by ATPG to FFs serially connected to form SCAN chain and by observing chain output, respectively ➔ **Defects are easily detected**

◆ Die size penalty is slight and testability improvement is overwhelming

SCAN Cell

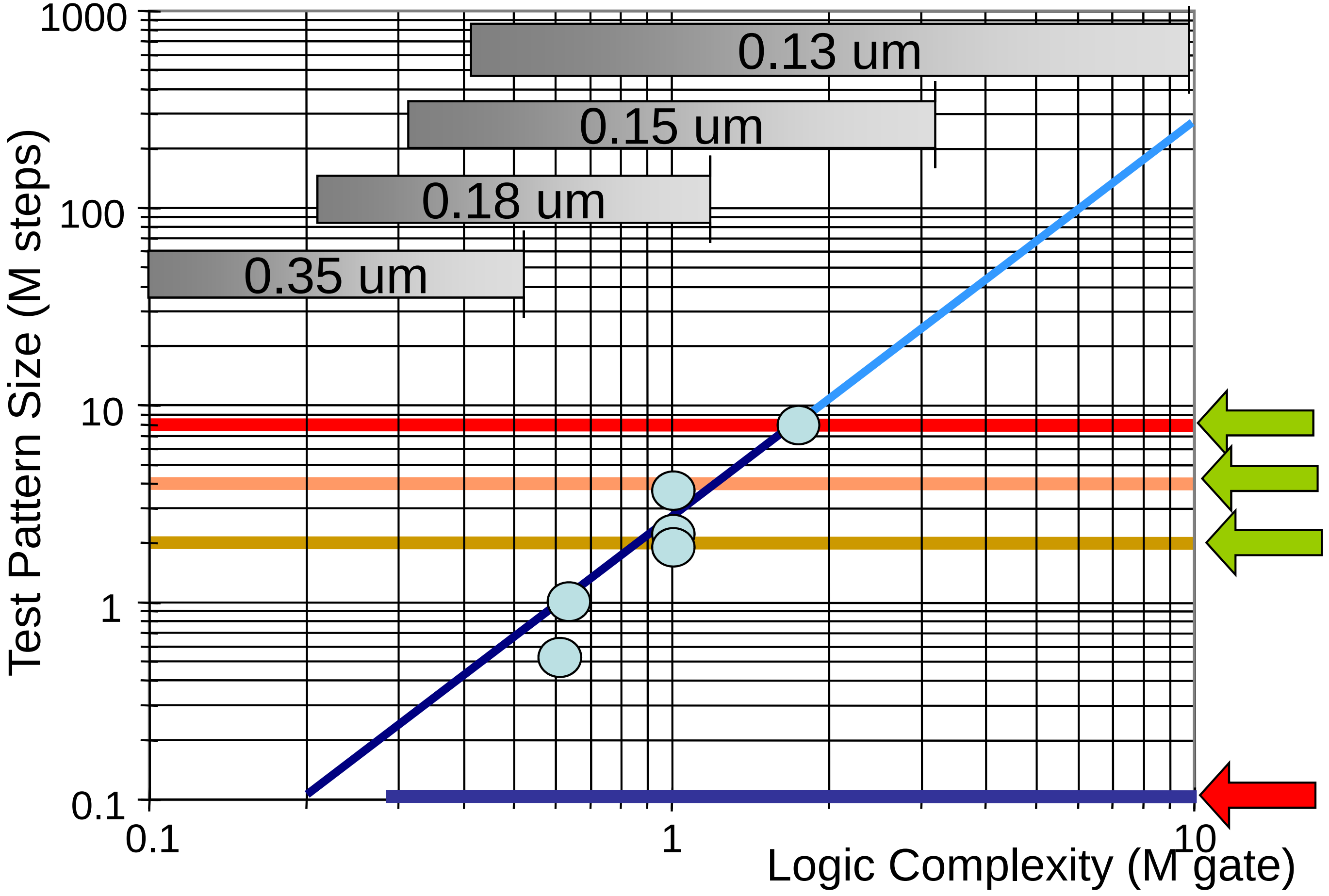


Typical organization of SCAN Cell



DFT (2)

- to cool down test pattern explosion



Numbers of test patterns have been restricted in each generation of LSI testers

BIST only requires less, constant test patterns

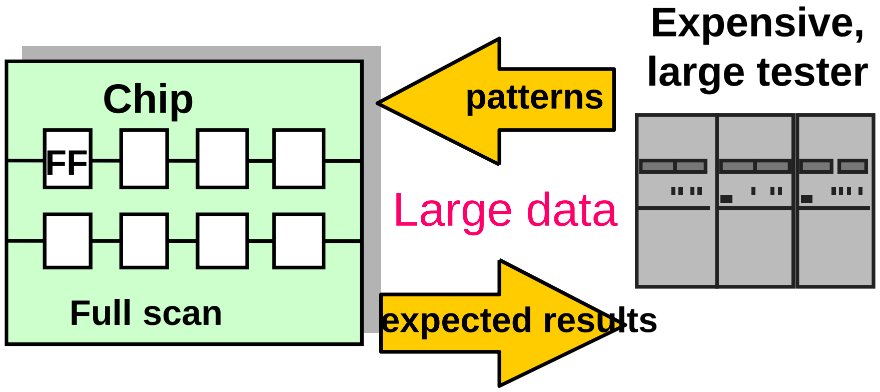
→ BIST is used to reduce test pattern size and shorten testing time

BIST

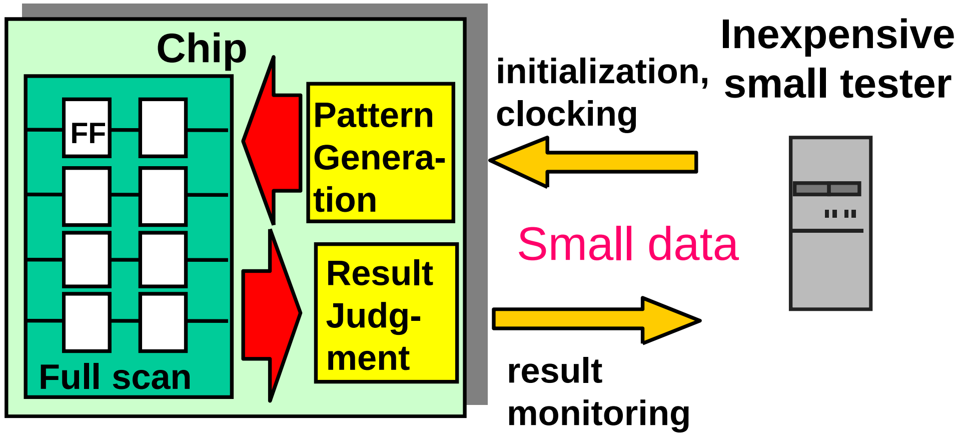
- BIST (Built-In Self Test) design embedded in single chip is comprised of the modules for generating test patterns and judging test results. An LSI tester can only apply clock signals and read testing results

- Pros: to downsize the tester and to shorten test time by concurrent tests
- Cons: die size penalty of 2% (BIST) together with 10% (SCAN) in typical cases, and constraints on clock signals and propagation of undefined operations

Scan test



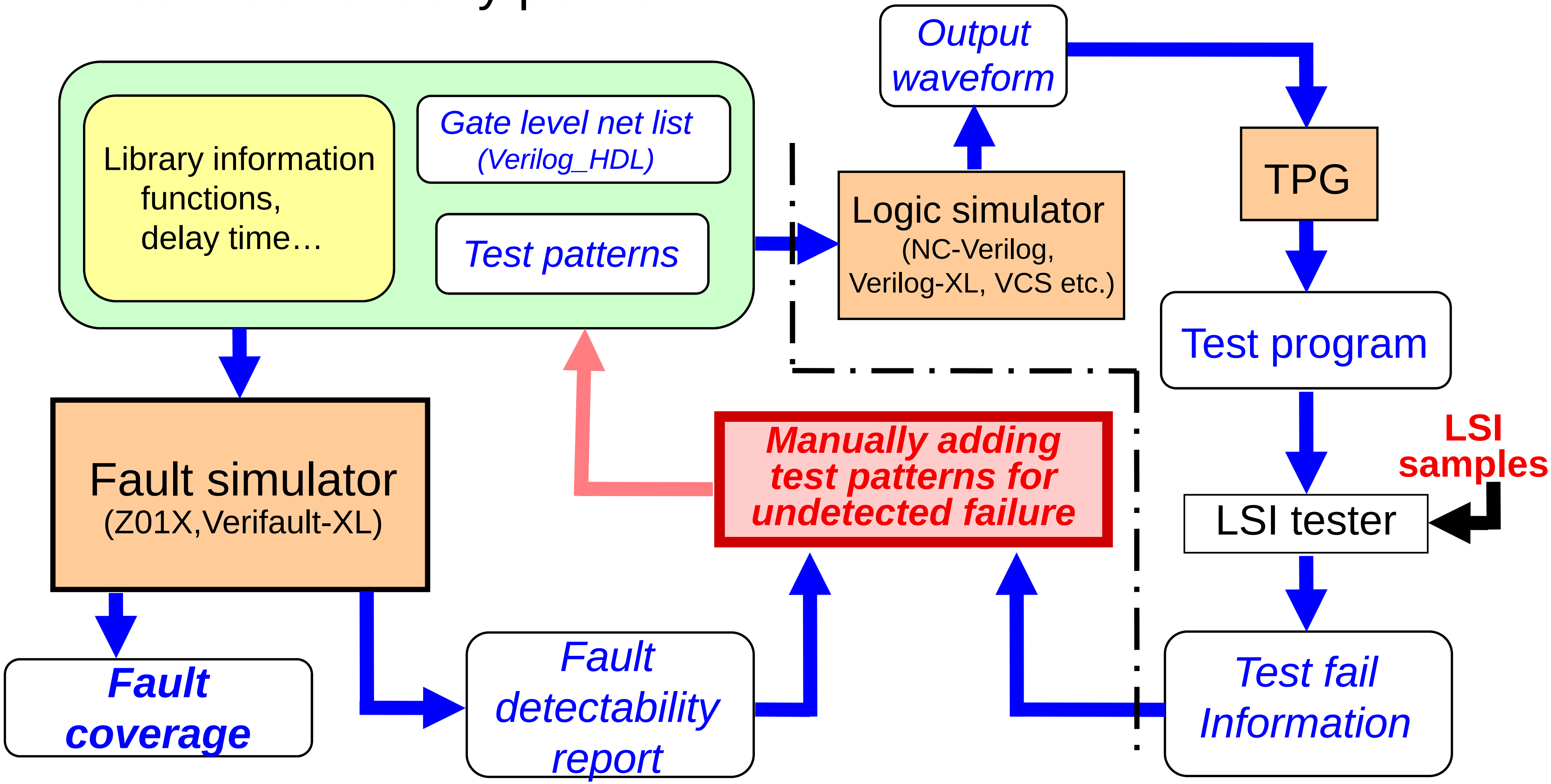
Logic BIST



Test Pattern Generation

- Fault coverage can be assessed by using fault simulator

  → Fault coverage can be improved by adding test patterns that can locate faulty points



Library information functions, delay time…
Gate level net list (Verilog_HDL)
Test patterns
Output waveform
Logic simulator (NC-Verilog, Verilog-XL, VCS etc.)
TPG
Test program
LSI samples
LSI tester
Test fail Information
Fault simulator (Z01X,Verifault-XL)
Manually adding test patterns for undetected failure
Fault coverage
Fault detectability report

Contents

1. Introduction

2. Photolithography, Mask Set, and MOS Transistor

3. SoC Design Flow

    Platform Development

    LSI Design Flow

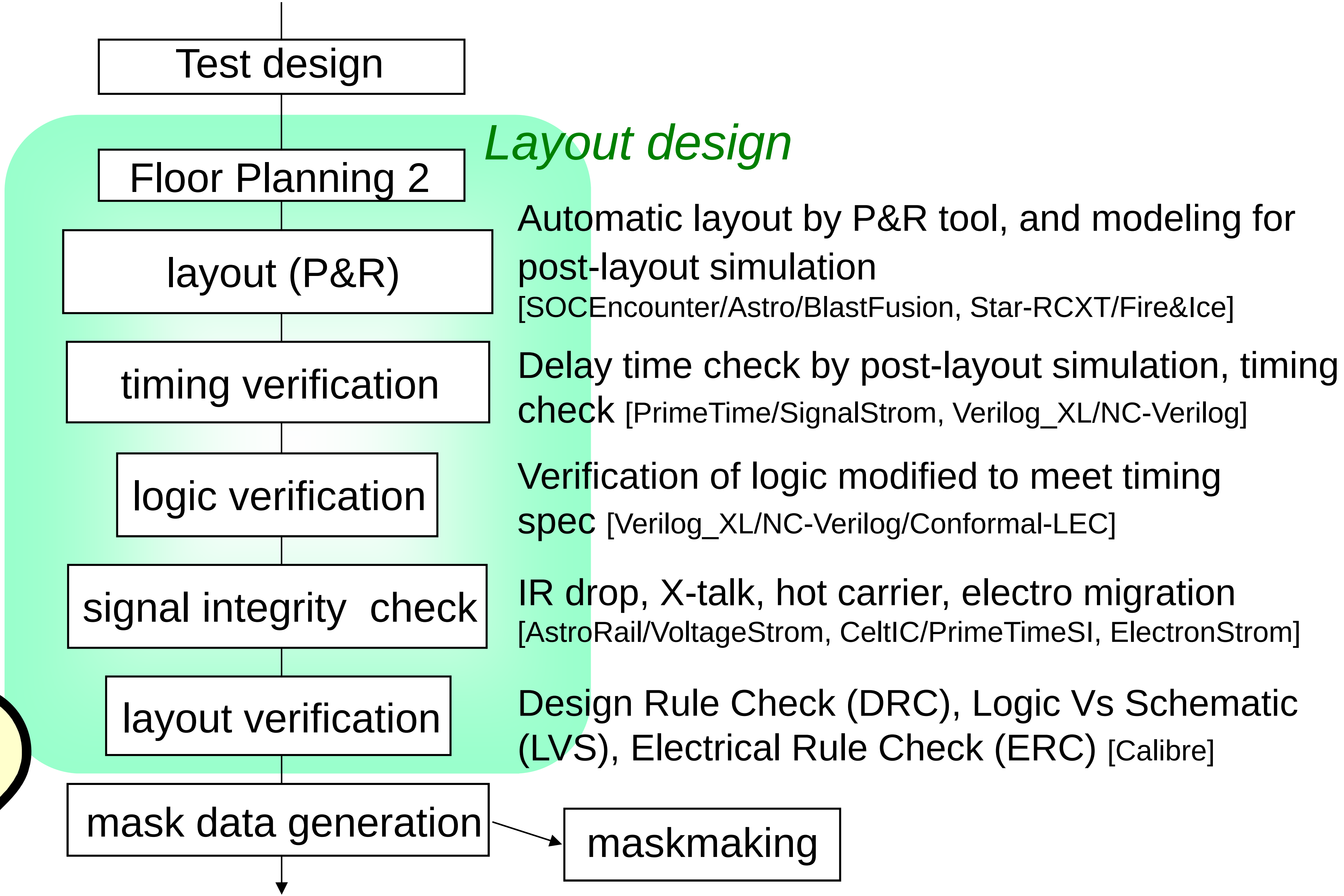
        System Design

        Logic Design

        Test Design

        Layout Design

Layout Design



Test design

Layout design

Floor Planning 2

layout (P&R)

Automatic layout by P&R tool, and modeling for post-layout simulation [SOCEncounter/Astro/BlastFusion, Star-RCXT/Fire&Ice]

timing verification

Delay time check by post-layout simulation, timing check [PrimeTime/SignalStrom, Verilog_XL/NC-Verilog]

logic verification

Verification of logic modified to meet timing spec [Verilog_XL/NC-Verilog/Conformal-LEC]

signal integrity check

IR drop, X-talk, hot carrier, electro migration [AstroRail/VoltageStrom, CeltIC/PrimeTimeSI, ElectronStrom]

layout verification

Design Rule Check (DRC), Logic Vs Schematic (LVS), Electrical Rule Check (ERC) [Calibre]

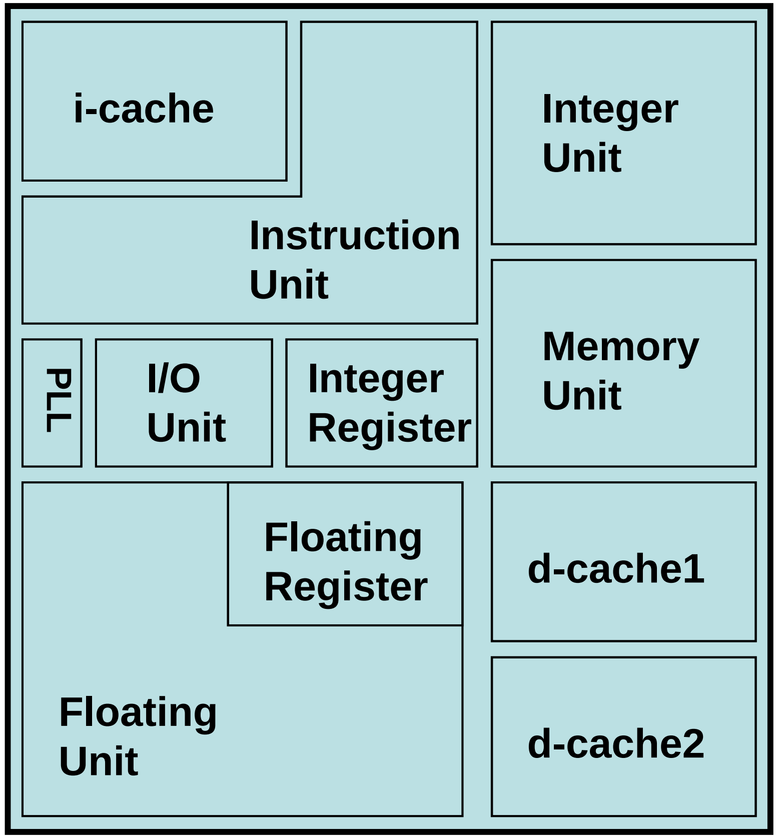
mask data generation

maskmaking

Floor Planning (2)

- Start design by using the floor plan considered in pre-layout timing verification

- **Bonding pad allocation**
- **Each block is P&Red by layout tool**
- **Power lines and signal lines should be deeply considered**



Example for floor plan of processor chip

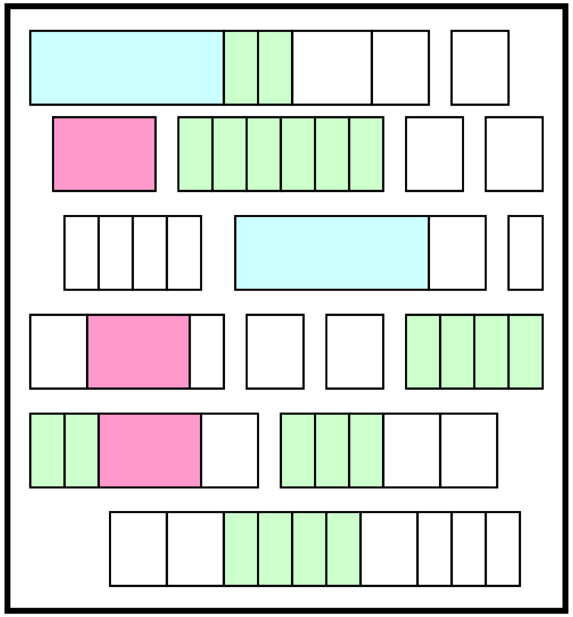
Automatic P&R

- Then, P&R (Place and Route）

<Automatic Placement>

Cells are automatically placed in each island under the following conditions:

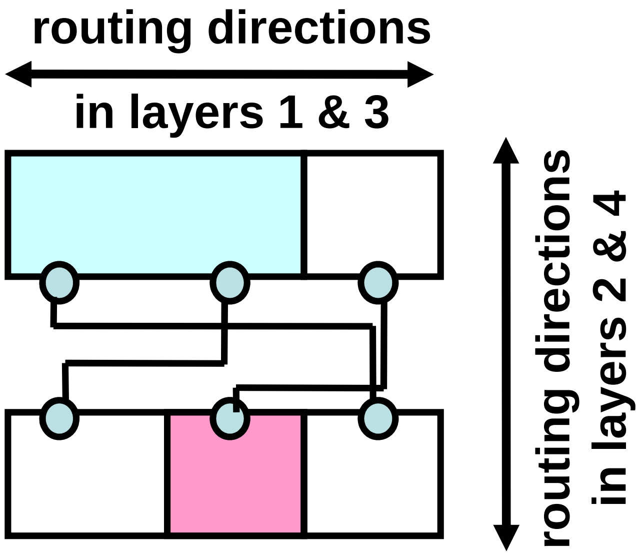
- to minimize die size
- to shorten inter-cell routing
- to enable fully automatic routing



<Automatic Routing>

Cells are automatically connected under the following conditions:

- to minimize wire length avoiding roundabouts
- to restrict routing directions in each layer
- to use available grids for wiring which are predetermined

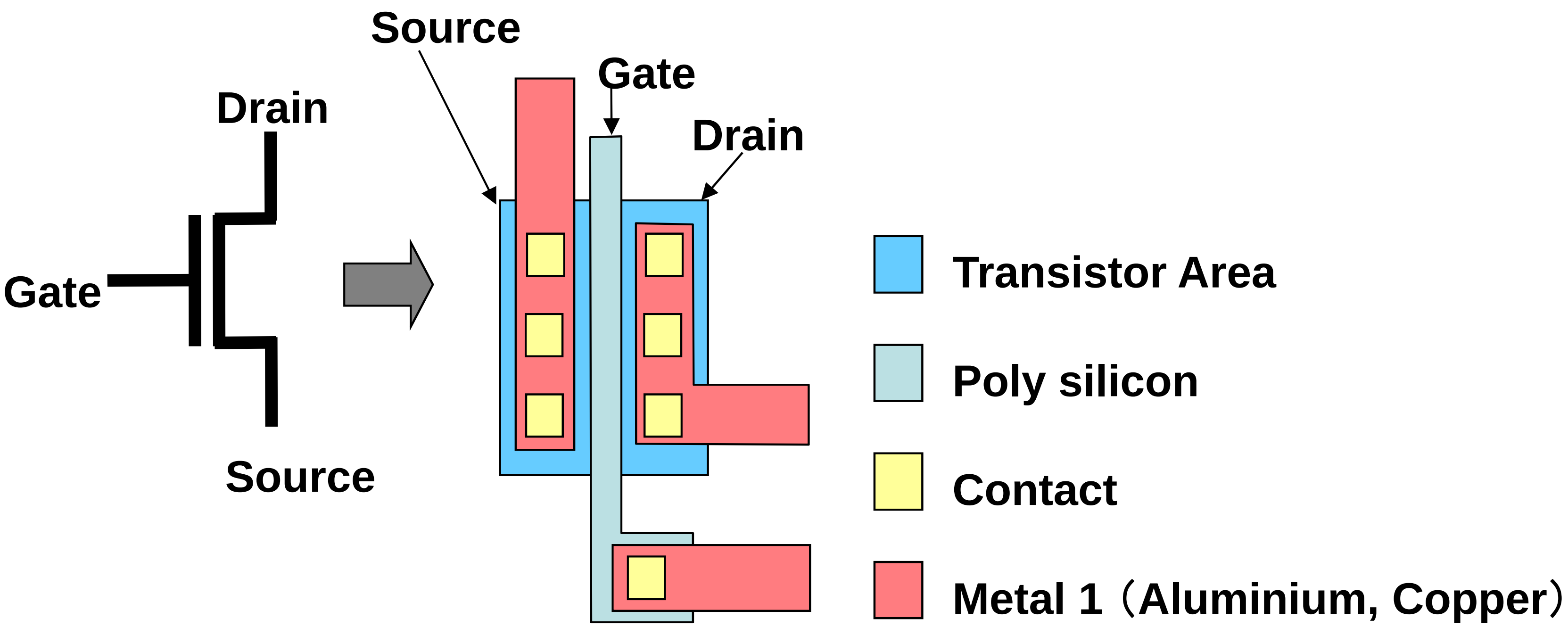


Manual Layout

- Manual layout if necessary

- Manual layout is adopted in logic LSI pursuing ultra high performance and/or smaller die size, or in analog LSI
- Based on design rules described in DM (Design manual)
- Manual layout area should be restricted, since its design cost is extremely high in comparison with automatic P&R

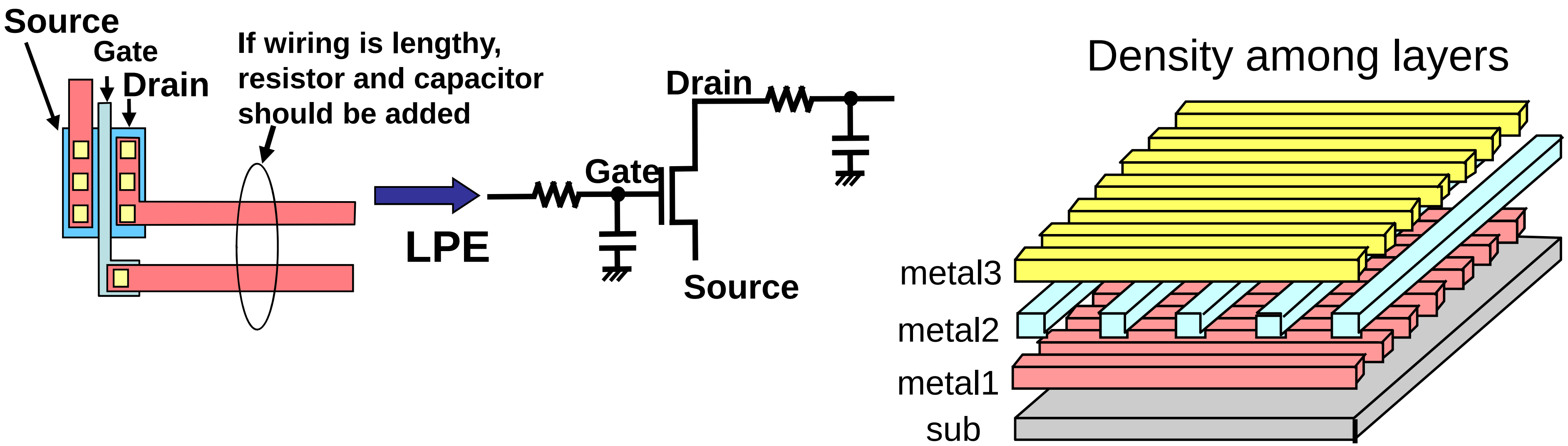
<Transistor layout image>



RC Extraction

- LPE (Layout Parasitic Extraction) for timing verification

- LPE extracts parasitic capacity and resistance from layout database and outputs circuit diagram to be used for simulation
- Fringe effect and cross couple are taken into consideration
- Density among multiple layers and effect of adjacent lines are further considered


If wiring is lengthy, resistor and capacitor should be added

Density among layers

Timing Verification

Post-layout verification

➢ STA:Static Timing Analysis)

- Extracted RC parameters are used in STA: post-layout
- Violations of setup time and hold time must be resolved by modifying logic description or layout
- Device samples with delay time violation could be evaluated by lowering operating frequency. But in setup and/or hold time violation case, there is no way to evaluate samples

➢ DTA: Dynamic Timing Analysis

- Logic simulation handling delay timing is executed
- Verification of all paths is unrealistic because huge test patterns are required

-> Verify synchronous paths with STA, and use DTA for limited paths that cannot be checked by STA. Always do synchronous design except for unavoidable case!

Signal Integrity

- In LSI design adopting nanometer technologies, preserving signal integrity and its verification become indispensable
- Signal Integrity is classified:

1) IR drop and ground bounce

Voltage across parasitic resistors reduces power supply voltage and raises ground voltage at each transistor.

These bring about malfunction or timing delay increase at each gate.

2) Cross Talk (X-talk)

Noise between multiple signal lines brings about malfunction or timing delay increase at each gate.

3) Electro Migration (EM)

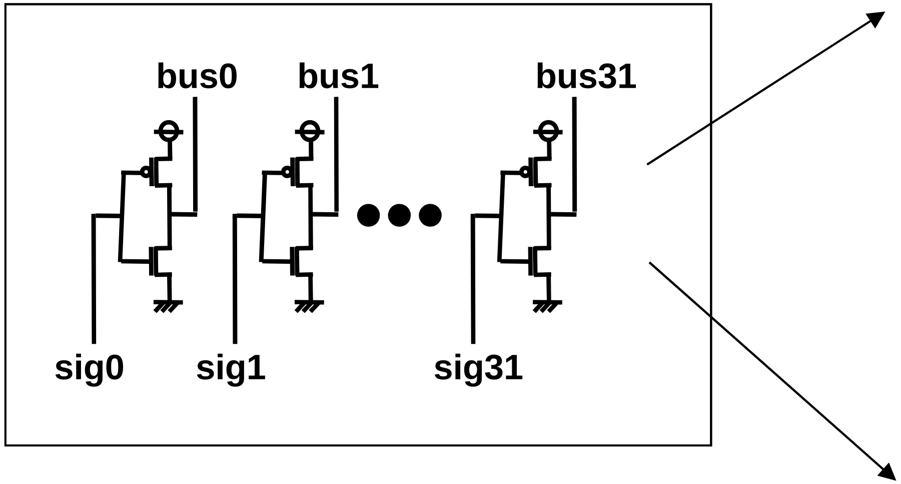
Excessive, repeated electric current damages wiring (wiring stress).

4) Hot Carrier (HC)

Excessively energized electrons damage transistor (Trs. stress).

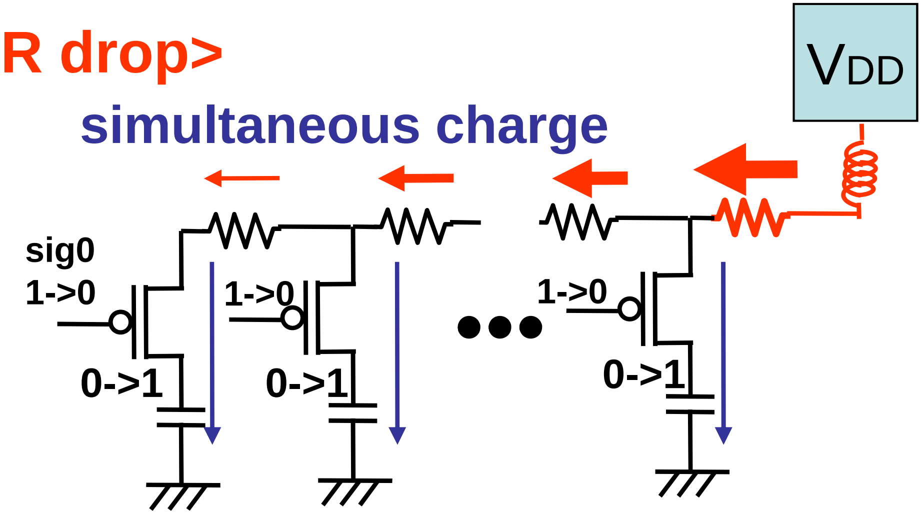
IR Drop and Ground Bounce

<example: bus driver>



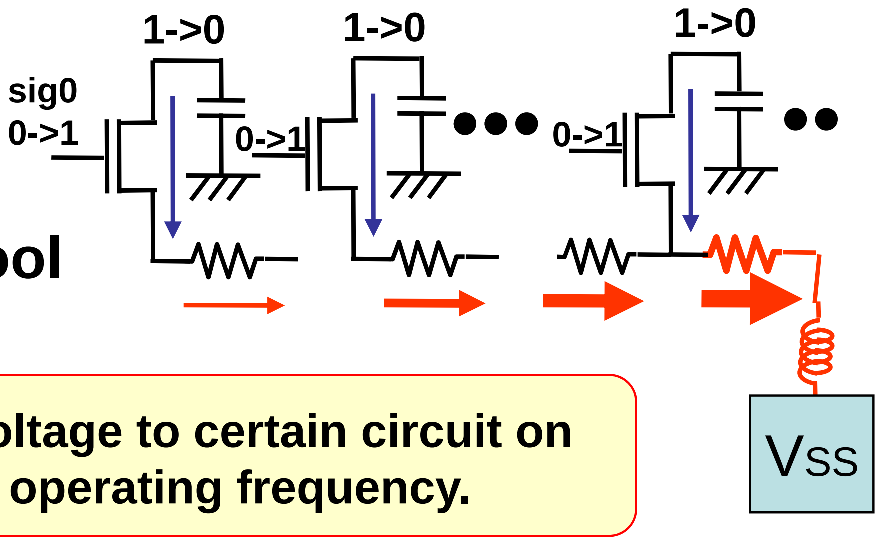
<IR drop>

simultaneous charge



<ground bounce>

simultaneous discharge



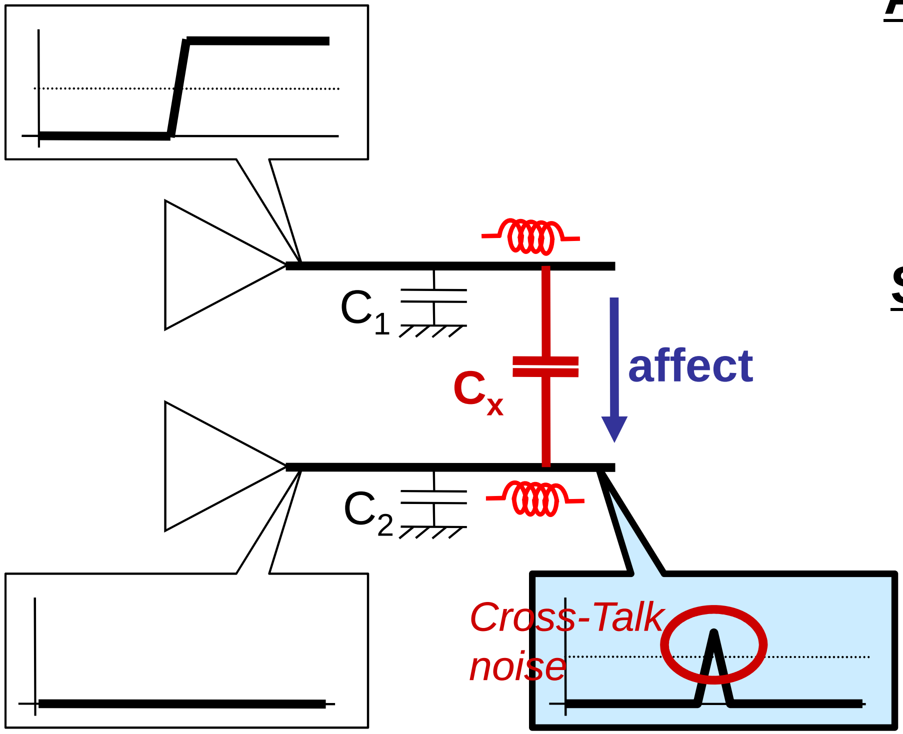
Detectable by checker within P&R tool

IR drop deteriorates power supply voltage to certain circuit on silicon, and limit miniaturization and operating frequency.

Cross Talk (1)

- **Cross-talk means signal deterioration due to noise generated at adjacent wiring by capacitive and/or inductive coupling**
  - **→ Malfunctions and delay timing shift will occur.**
  - **→ Cross talk becomes dominant as wiring pattern becomes finer.**

- **Logical malfunctions**



Asynchronous logic inversion

Noise on set/reset signals may cause asynchronous logic inversion.

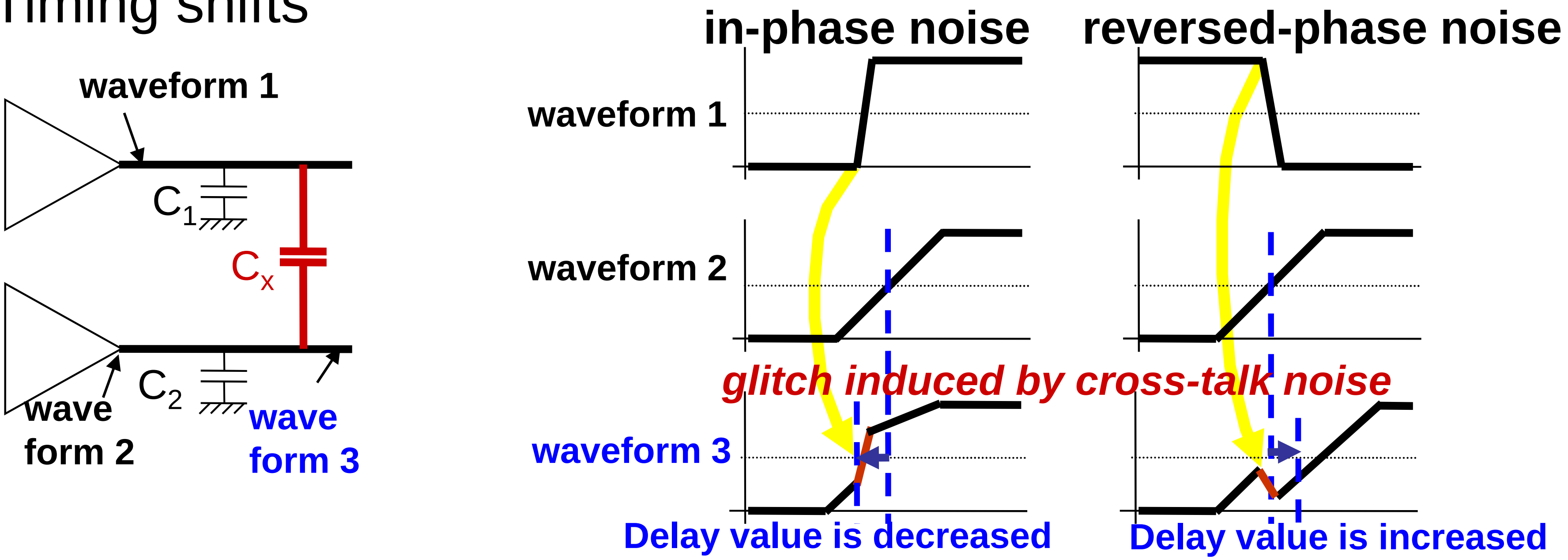
Synchronous logic inversion

Noise on clock signal may synchronously flip FF output.

Logic 0 ➔ momentarily recognized as logic 1

Cross Talk (2)

- Timing shifts

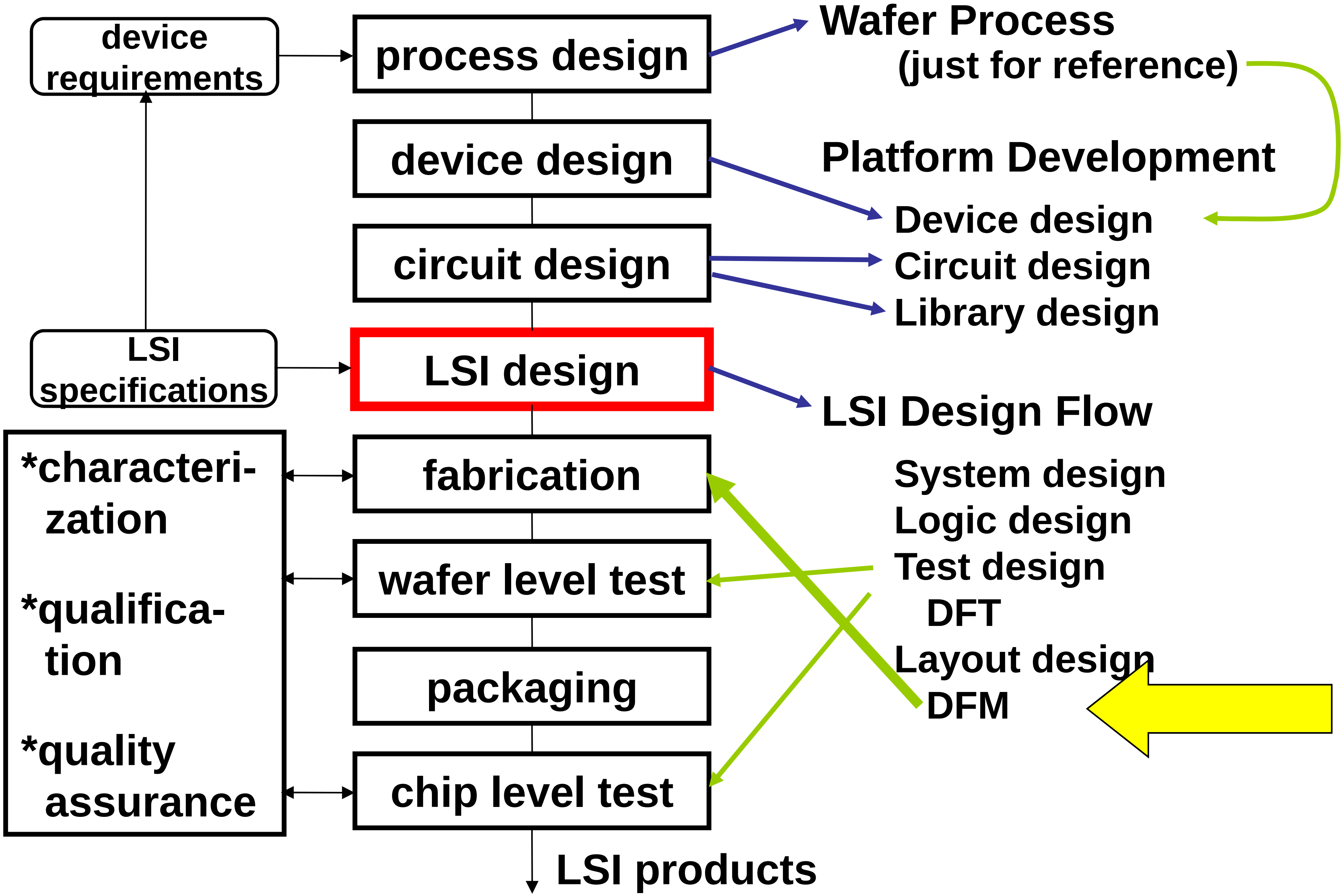


**Delay value decreases in the in-phase noise
and
increases in the reversed phase noise**

Unexpected timing violation may occur.

Detectable by P&R tools

DFM in LSI Development



device requirements

process design

device design

circuit design

LSI specifications

LSI design

fabrication

wafer level test

packaging

chip level test

LSI products

*characterization

*qualification

*quality assurance

Wafer Process
(just for reference)

Platform Development
- Device design
- Circuit design
- Library design

LSI Design Flow
- System design
- Logic design
- Test design
  - DFT
- Layout design
  - DFM

Design For Manufacturing

DFM: technologies to solve various fabrication related problems in design phases ahead

Precise estimation of variance

➔Performance optimization

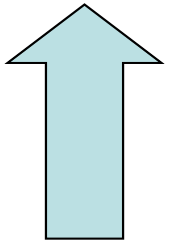
➔Yield improvement

Measures to reduce fabrication variance

- 130nm node or earlier: process and device engineers' job
- 90nm node: LSI designers consider variance effect of transistor and wiring
- 65nm node: LSI designers further consider variance effect of CMP (Chemical Mechanical Polishing) and photolithography

Abstract of DFM

Production process	Cause	Phenomenon	Measures
forming of films (deposition)	+ foreign particles	+ random o/s	+ double VIA + cell swapping + spacing change*
lithography	+ misalignment of masks + diffraction	+ o/s + characteristics deviation	+ VIA shift + spacing change*
CMP	+ over etching	+ open	+ dummy metal

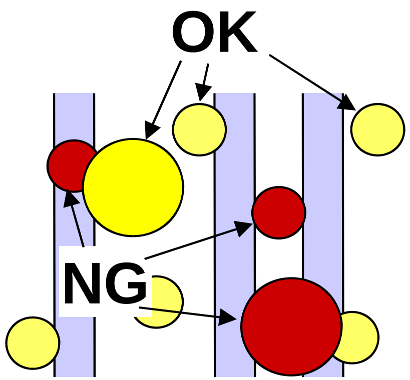


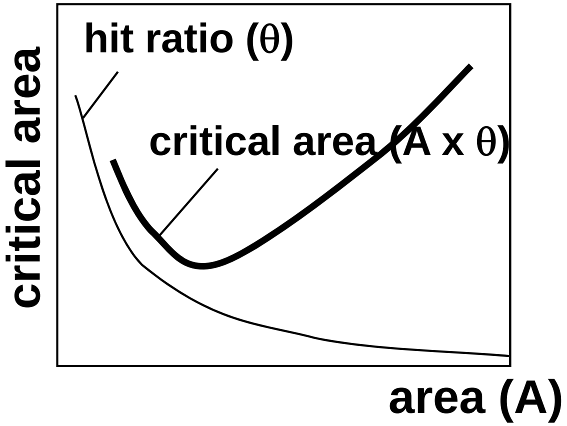
Process Variance

* Common to platform development and LSI design (others are specific to LSI design)

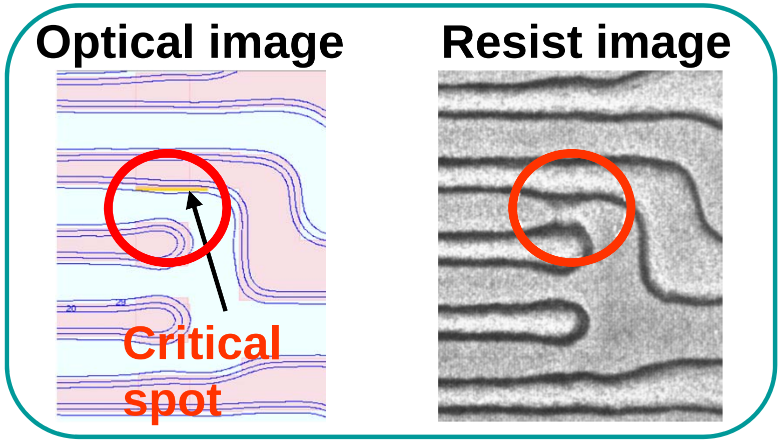
Defects

Random defects: due to foreign particles
Systematic defects: dependent on layout pattern





Critical Area Analysis
- random defects -

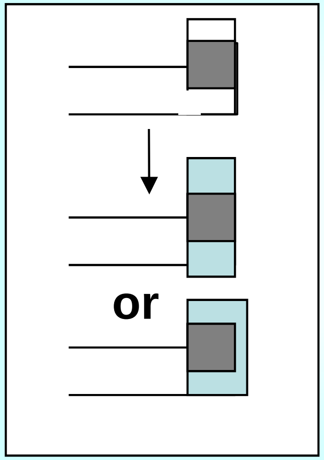


Lithography Simulation
- systematic defects-

DFM in Platform Development

Lithography-friendly technology data development

Modification of technology data and layout rules



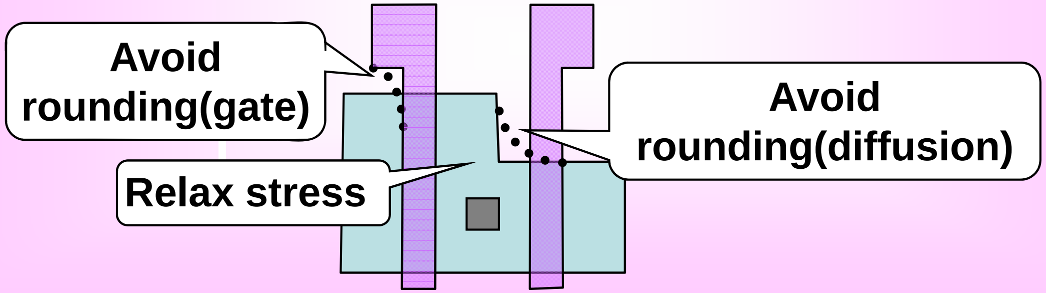
ex. resizing of dog bone shape for improving via margins

enlarging space

Lithography-friendly cell layout design

Optimization of space margins

ex. Optimization of diffusion and poly



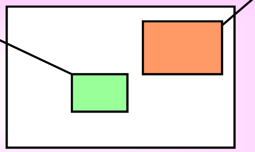
DFM in LSI Design

Particle-aware layout

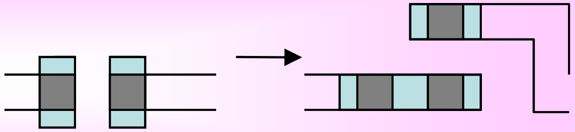
- cell swapping

Cell used in crowded area (small cell, high failure rate)

Cell used in less crowded area (large cell, low failure rate)



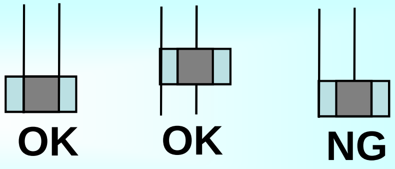
- double/redundant via
- wire spreading
- wire widening



ex. redundant via and detour

Lithography-friendly layout

- critical spot analysis



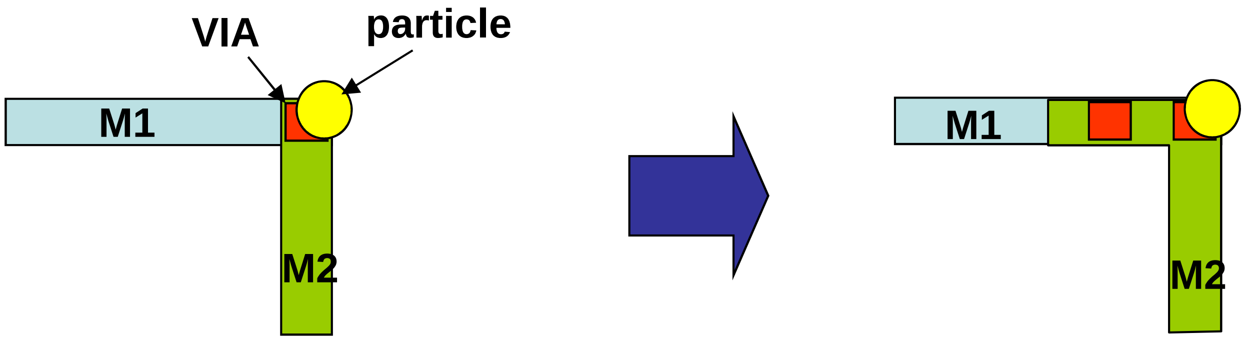
ex. critical spot analysis

CMP-aware layout

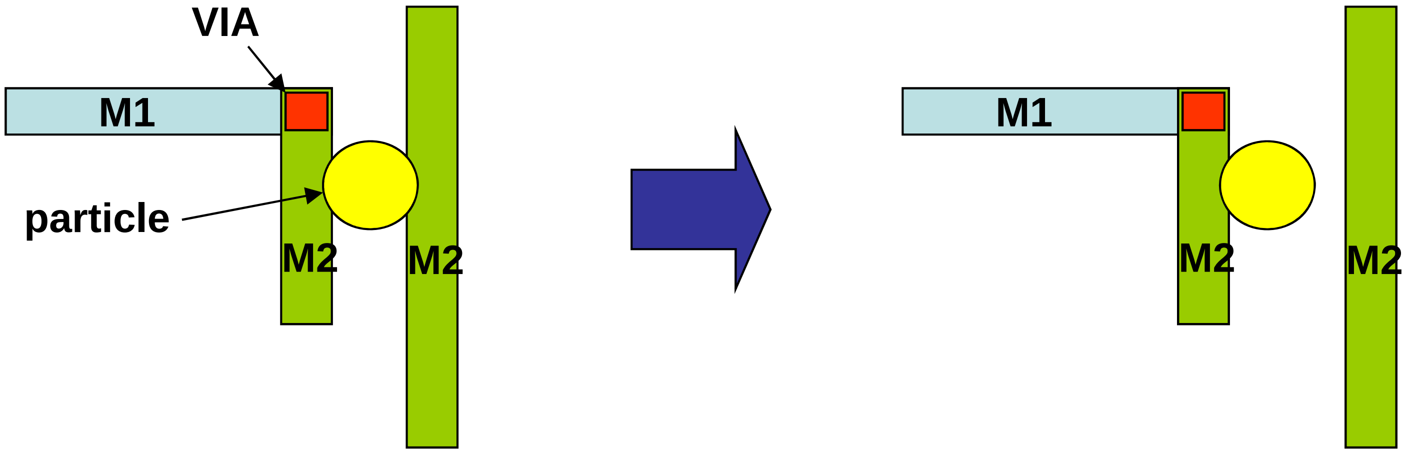
- dummy metal insertion

Measures against Random Defects

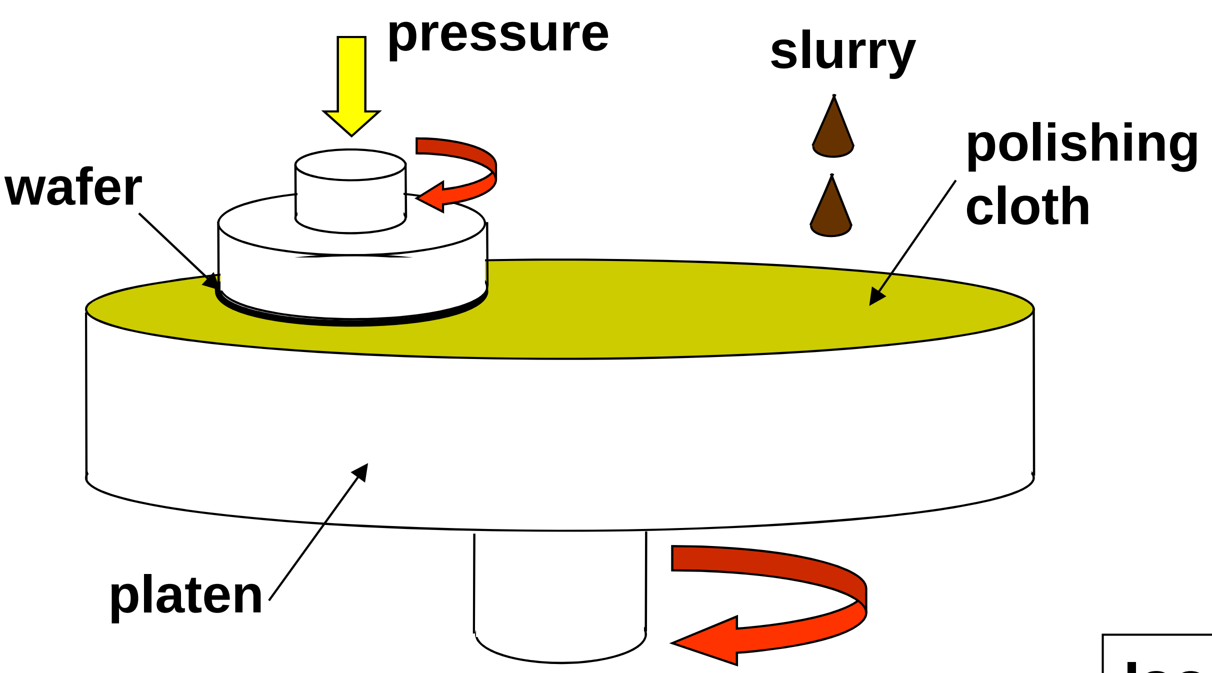
Double VIA

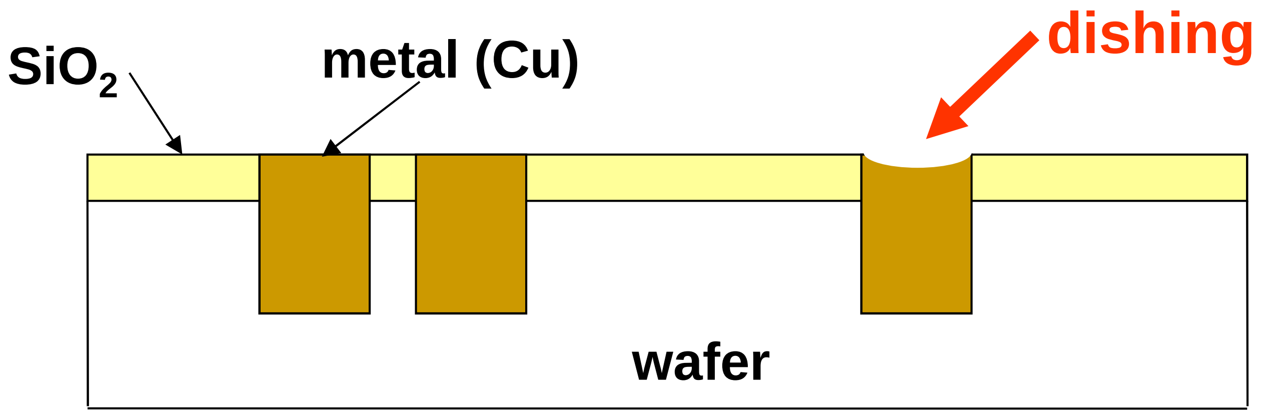


Wire Spreading



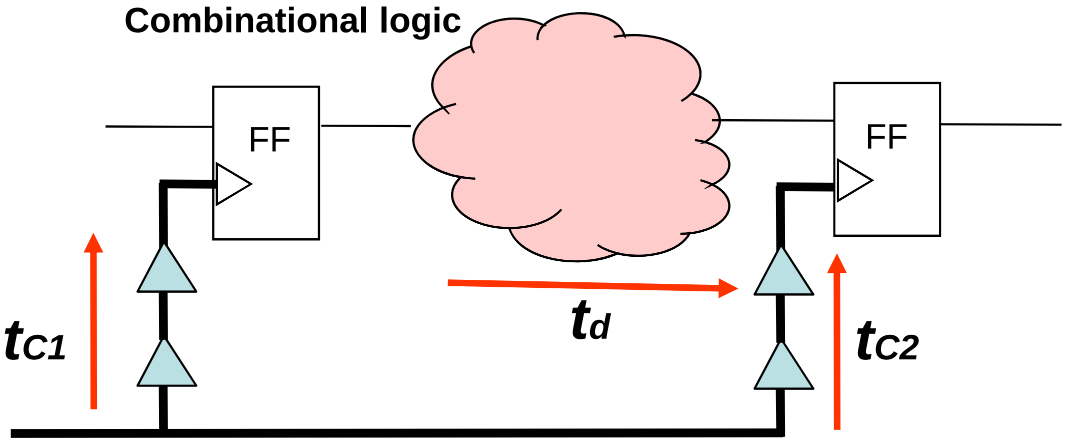
CMP





Isolated wider wiring will be over-polished. To avoid this, dummy metal is inserted in layout design.

Statistical Static Timing Analysis



Not DFM but…

Performance (timing) is variant depending on process variance.

Conventional STA

$$\max t_{C1} + \max t_d - \min t_{C2} < t_{CYC} - t_{SETUP} \quad \text{: setup time}$$

$$\min t_{C1} + \min t_d - \max t_{C2} > t_{HOLD} \quad \text{: hold time}$$

excessive timing margins

Statistical STA (SSTA)

Delay timing or slack is handled by statistical variables based on distribution functions

SSTA is different from DFM, but sometimes discussed with DFM.

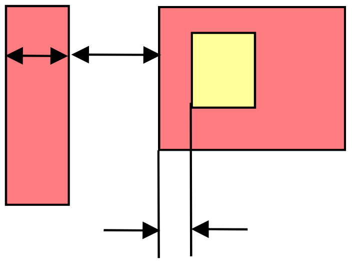
Layout Verification

- DRC and LVS are the two major tools for layout verification.

<DRC: Design Rule Check>

- Check width, space, and overlap of patterns among several layout layers according to layout rules.

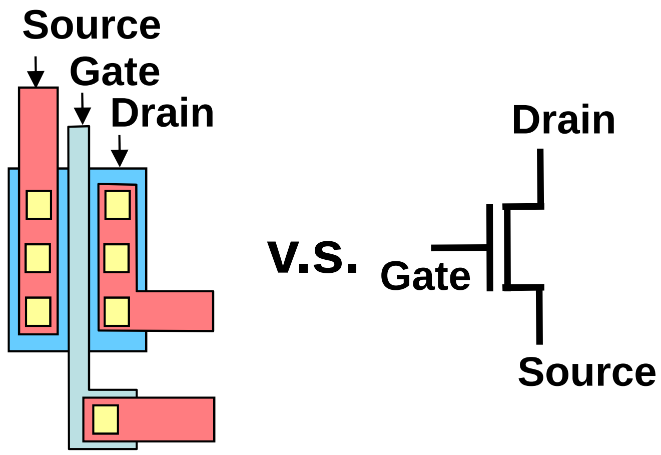
▪ In case of any violations, correct layout and check again. Repeat this process until no violation is reported at all.



<LVS: Layout Vs Schemathic Check>

▪Check whether layout corresponds to reference, i.e. circuit diagram (netlist).

▪ In case of inconsistency, correct layout and check again. Repeat this process until inconsistency is solved.

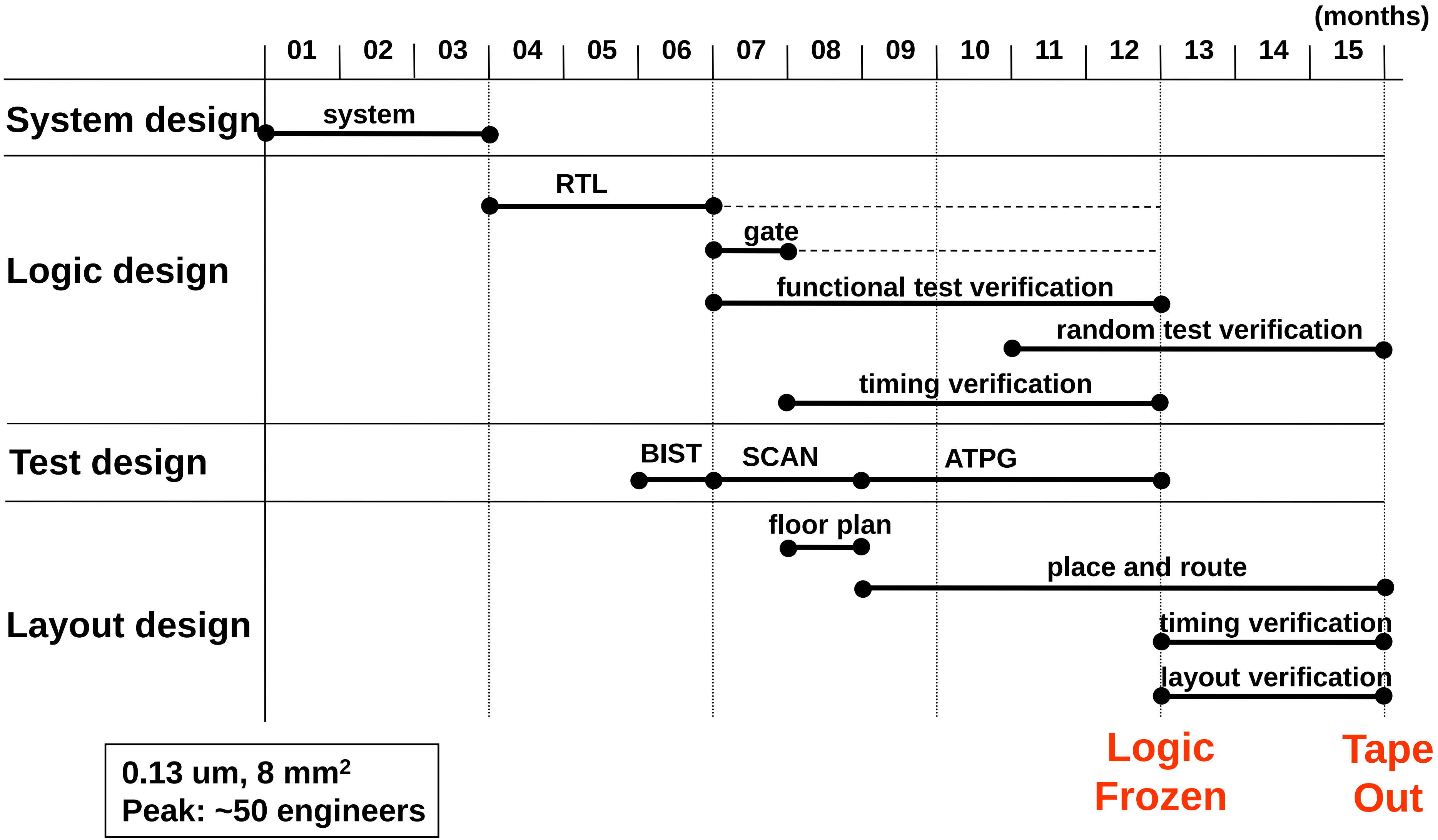


<Layout>

<Circuit Diagram>

Example of LSI Design

- development of new IP (processor core) -



0.13 um, 8 mm^2
Peak: ~50 engineers

LSI Product Troubles and Designs

➢Functional errors

- deterministic fault → to be avoided by logic verification
- intermittent fault → to be avoided by logic and timing verification

➢Marginal problems

- heat, low power supply voltage → to be avoided by circuit simulation
- noise → to be avoided by circuit simulation

➢Fabrication problems

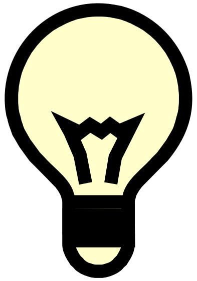
- screening of defective devices → to be solved by DFT (Design For Test)

➢Fluctuation of fabrication conditions

- yield problem → to be solved by DFM
  (Design For Manufacturing)

➢Aged deterioration

- hot carrier → to be avoided by signal integrity check
- electro migration → be avoided by signal integrity check



Thank you for your attention !

Renesas Electronics Corporation

Rev 5.4