

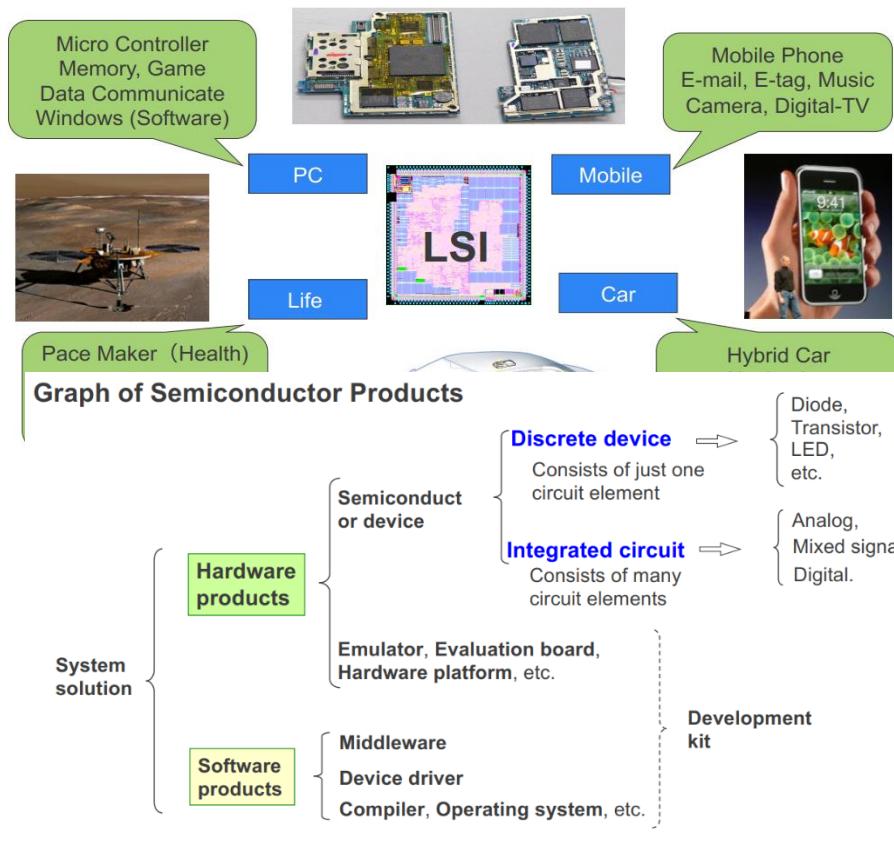
CHAPTER 1. Introduction to LSI Development

1.1. Semiconductor Products.

1. Phân loại IC theo mức tích hợp

(Integration Scale): Dựa vào số lượng linh kiện tích hợp bên trong chip:

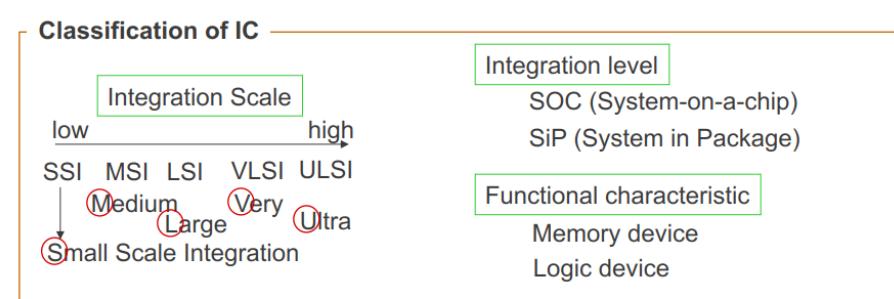
- SSI (Small Scale Integration)**: Tích hợp ít cổng logic (từ 1 đến vài chục).
- MSI (Medium Scale Integration)**: Vài chục đến vài trăm cổng logic.
- LSI (Large Scale Integration)**: Hàng ngàn cổng logic.
- VLSI (Very Large Scale Integration)**: Hàng chục ngàn đến hàng triệu cổng logic.
- ULSI (Ultra Large Scale Integration)**: Hàng triệu đến hàng tỷ cổng logic.



2. Phân loại IC theo mức tích hợp hệ thống

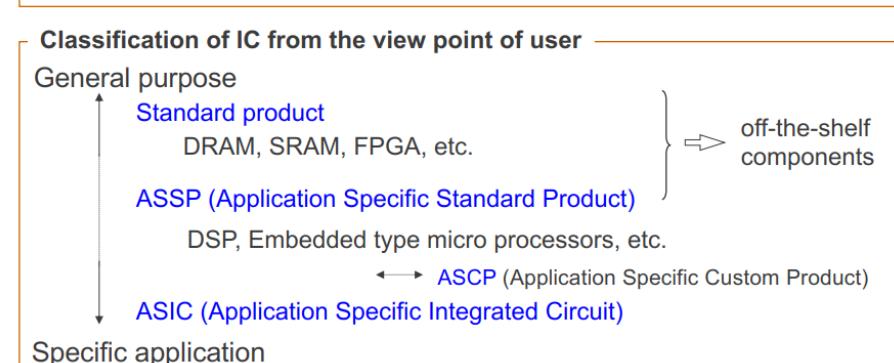
(Integration Level):

- SoC (System-on-a-Chip)**: Một hệ thống đầy đủ (CPU, bộ nhớ, I/O,...) tích hợp trên một chip duy nhất.
- SiP (System-in-Package)**: Nhiều chip tích hợp trong cùng một gói (package), có thể bao gồm CPU, bộ nhớ, RF,... nối với nhau qua interposer hoặc bonding.



3. Phân loại IC theo chức năng (Functional Characteristic):

- Memory device**: IC lưu trữ (ví dụ: DRAM, SRAM,...).
- Logic device**: IC xử lý logic (ví dụ: CPU, FPGA,...).



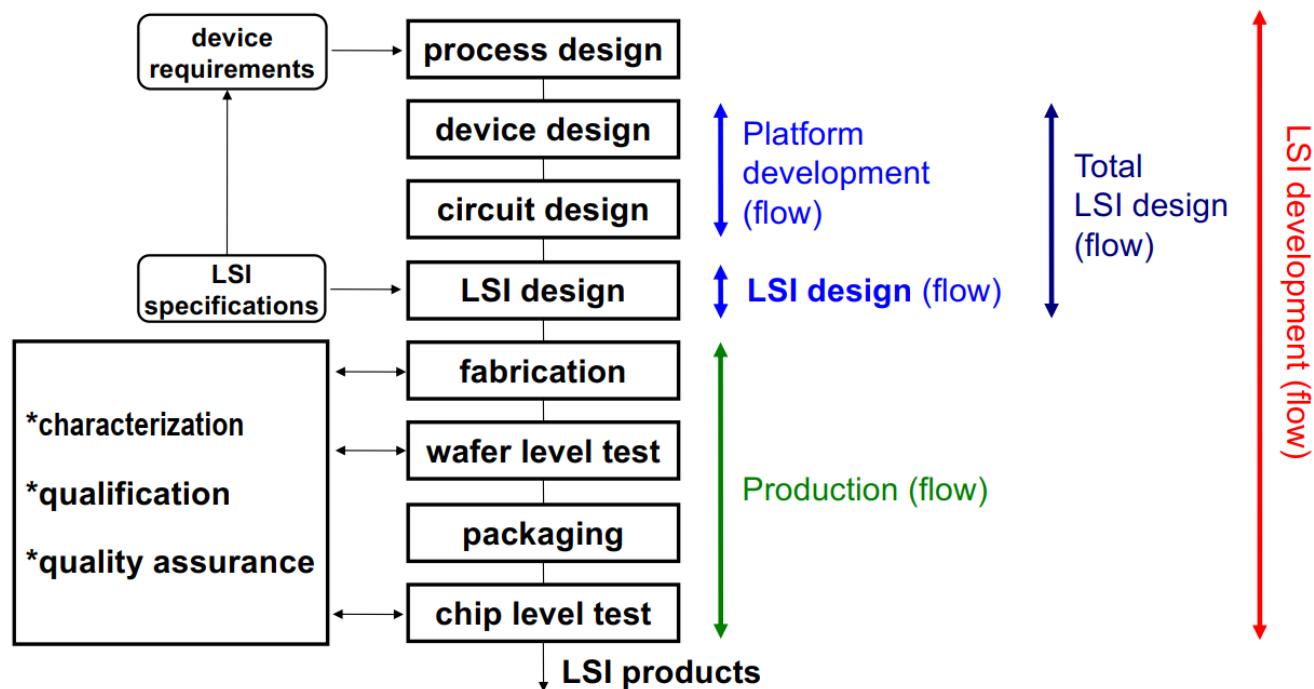
4. Phân loại IC theo quan điểm người sử dụng (From the view point of user):

Từ mục đích sử dụng tổng quát (general purpose) đến mục đích chuyên biệt (specific application):

- Standard Product** (Sản phẩm tiêu chuẩn):
 - Dùng chung cho nhiều ứng dụng.
 - Ví dụ: DRAM, SRAM, FPGA.
 - Là **off-the-shelf components** – có thể mua sẵn.
- ASSP (Application Specific Standard Product)**:
 - Dành cho các ứng dụng cụ thể nhưng vẫn phổ biến (ví dụ: DSP, bộ vi xử lý nhúng).
 - Không cần thiết kế lại từ đầu.
- ASCP (Application Specific Custom Product)**:
 - Thiết kế tùy chỉnh dựa trên yêu cầu cụ thể.
- ASIC (Application Specific Integrated Circuit)**:
 - Thiết kế cho một ứng dụng rất cụ thể, không linh hoạt.
 - Hiệu năng cao, tiêu thụ điện năng thấp, nhưng chi phí thiết kế ban đầu cao.

1.2. LSI Development Flow.

LSI Development Flow



LSI (Large Scale Integration) Development Flow

1. Tổng quan các giai đoạn chính:

Quy trình phát triển LSI được chia thành nhiều bước nối tiếp nhau từ yêu cầu thiết kế đến sản phẩm cuối cùng.

Yêu cầu và đặc tả:

- Device requirements:** Yêu cầu hệ thống/mạch từ phía khách hàng hoặc nhóm ứng dụng.
- LSI specifications:** Đặc tả chi tiết để thiết kế IC, dựa trên yêu cầu hệ thống.

2. Quy trình phát triển chính (LSI development flow - mũi tên đậm):

- 1. Platform Development Flow (mũi tên xanh dương nhạt):**
 - Process design:** Thiết kế công nghệ bán dẫn (cấu trúc bóng bán dẫn, lớp metal, v.v).
 - Device design:** Thiết kế thiết bị bán dẫn (transistor, diode, v.v).
- 2. LSI Design Flow (mũi tên xanh dương):**
 - Circuit design:** Thiết kế mạch (gồm sơ đồ mạch, mô phỏng, chọn linh kiện).
 - LSI design:** Thiết kế layout, netlist, sơ đồ kết nối các khối logic.
- 3. Production Flow (mũi tên xanh lá cây):**
 - Fabrication:** Quá trình sản xuất wafer (in mẫu mạch lên silicon).
 - Wafer level test:** Kiểm tra chip trực tiếp trên wafer (chưa cắt ra).
 - Packaging:** Đóng gói chip (thường là plastic/ceramic).
 - Chip level test:** Kiểm tra chức năng và hiệu năng của từng chip sau đóng gói.

3. Khâu đánh giá và đảm bảo chất lượng:

- Characterization:** Đặc trưng hóa các thông số điện/tín hiệu của chip.
- Qualification:** Kiểm tra độ tin cậy theo thời gian, nhiệt độ, điện áp, v.v.
- Quality Assurance:** Đảm bảo chất lượng toàn diện trước khi xuất xưởng.

4. Kết quả cuối cùng:

- LSI products:** Các sản phẩm LSI hoàn chỉnh sẵn sàng đưa ra thị trường.

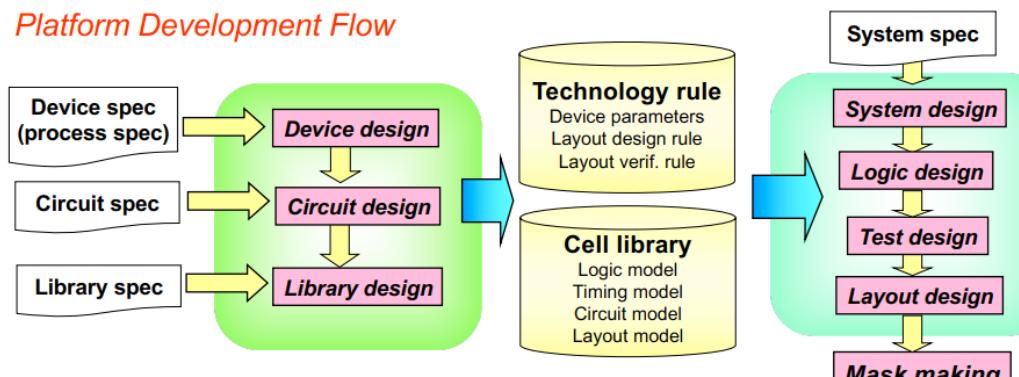
Total LSI Design Flow

1. Tổng Quan:

Quy trình thiết kế LSI tổng thể được chia làm hai phần chính:

- Platform Development Flow** (màu cam)
- LSI Design Flow** (màu xanh dương)

Platform Development Flow



2. Platform Development Flow

(Nền tảng thiết kế):

► Input:

- Device spec (Process spec)**: Thông số thiết bị, công nghệ chế tạo.
- Circuit spec**: Thông số mạch điện.
- Library spec**: Thông số thư viện cell (gates, flip-flops,...).

► Các bước thực hiện:

- Device design**: Thiết kế thiết bị bán dẫn cơ bản.
- Circuit design**: Thiết kế và tối ưu mạch logic cơ bản (gồm logic gates, mux, latch,...).
- Library design**: Tạo thư viện các cell cơ bản có thể tái sử dụng.

► Output:

- Technology rule**: Quy tắc công nghệ (thông số thiết bị, quy tắc layout, quy tắc kiểm tra layout).
- Cell library**: Bao gồm:
 - Logic model
 - Timing model
 - Circuit model
 - Layout model

3. LSI Design Flow (Thiết kế hệ thống cụ thể):

► Input: System spec

Yêu cầu hệ thống (chức năng, hiệu năng, kiến trúc,...).

► Các bước thiết kế:

- System design**: Thiết kế cấp hệ thống (chia khối chức năng, định nghĩa kiến trúc).
- Logic design**: Thiết kế logic chi tiết (viết HDL như Verilog/VHDL, RTL).
- Test design**: Thiết kế test (DFT – Design for Testability, scan chain,...).
- Layout design**: Thiết kế vật lý (floorplanning, placement, routing).
- Mask making**: Tạo mặt nạ quang khắc (dùng cho quá trình chế tạo chip).

4. Kết nối giữa hai flow:

- Technology rule** và **Cell library** từ Platform Development Flow → được sử dụng trong LSI Design Flow.

1. Các bước trong quy trình phát triển LSI

Từ trên xuống dưới:

- Process/Device** – thiết kế quy trình công nghệ và thiết bị.
- System spec** – đặc tả hệ thống.
- System design** – thiết kế cấp hệ thống.
- Logic design** – thiết kế logic.
- Test design** – thiết kế kiểm thử.
- Layout design** – thiết kế vật lý.
- Mask making** – tạo mặt nạ quang khắc.
- Fabrication & QA** – sản xuất và đảm bảo chất lượng.

2. Các mô hình kinh doanh tham gia vào các giai đoạn khác nhau:

◆ Lib. House (Library House):

- Phụ trách phần Process/Device.
- Cung cấp thư viện cell (standard cells, I/O cells...).

◆ Design House:

- Tham gia từ Logic design → Layout design.
- Có thể kết hợp với đối tác (\leftrightarrow ký hiệu hai chiều).

◆ Fabless:

- Là công ty thiết kế chip nhưng **không có nhà máy sản xuất (foundry)**.
- Tham gia từ System spec → Mask making.
- Giao việc sản xuất cho Foundry (vd: TSMC, UMC).

◆ Foundry:

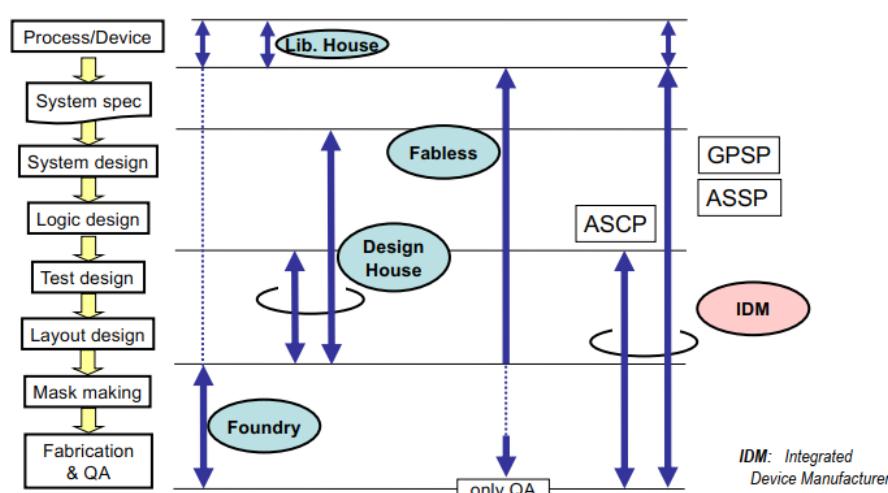
- Chuyên trách Mask making → Fabrication & QA.
- Không tham gia vào thiết kế logic.

◆ ASCP (ASIC Service Chip Provider):

- Là trung gian cung cấp dịch vụ thiết kế chip theo yêu cầu.
- Kết nối các khâu thiết kế → sản xuất.

◆ IDM (Integrated Device Manufacturer):

- Là công ty **tự làm mọi thứ từ đầu đến cuối**.
- Bao gồm: thiết kế, sản xuất, kiểm thử → sản phẩm cuối (vd: Intel, Samsung, Texas Instruments).
- Là mô hình khép kín.



3. Các sản phẩm liên quan (bên phải):

- **GPSP**: General Purpose Standard Product.
- **ASSP**: Application Specific Standard Product.

Các sản phẩm này có thể được tạo ra bởi ASCP, Fabless hoặc IDM.

1.3. LSI Design Flow and Methodologies.

1. Platform Development Flow (bên trái):

Phát triển nền tảng cho việc thiết kế LSI — bao gồm:

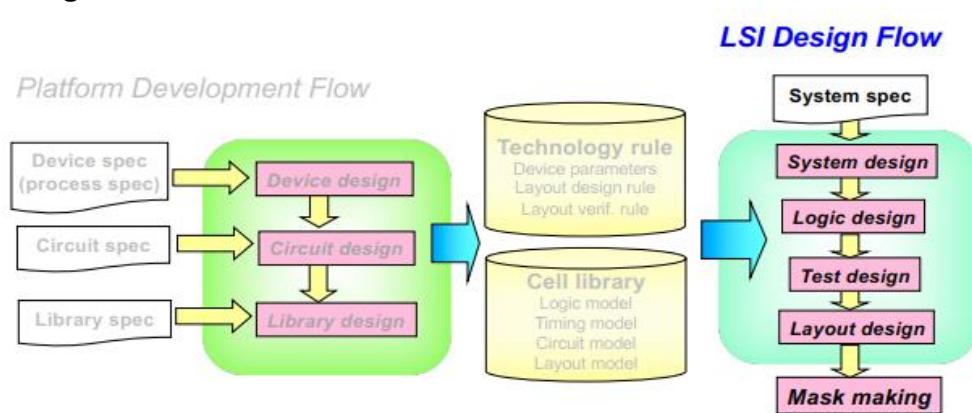
- **Device spec → Device design**
- **Circuit spec → Circuit design**
- **Library spec → Library design**

➡ Kết quả của bước này là:

- **Technology rule**: quy tắc thiết kế công nghệ (parameters, rules).
- **Cell library**: thư viện các cell tiêu chuẩn (logic, timing, layout...).

2. LSI Design Flow (bên phải):

Dựa trên system spec, sử dụng **Technology rule** và **Cell library** để thực hiện:



- **System design**
- **Logic design**
- **Test design**
- **Layout design**

- **Mask making**

✓ Ý nghĩa tổng thể:

- **Platform Development Flow** là phần nền tảng và được thực hiện **một lần cho mỗi công nghệ/chip platform**.
- **LSI Design Flow** là quá trình thiết kế **từng con chip cụ thể** dựa trên nền tảng đã có.
- Mỗi giai đoạn liên quan đến các công cụ EDA, kiểm tra tính đúng đắn (verification), và chuẩn bị sản xuất (mask making).

1. Các bước trong quy trình LSI Design Flow:

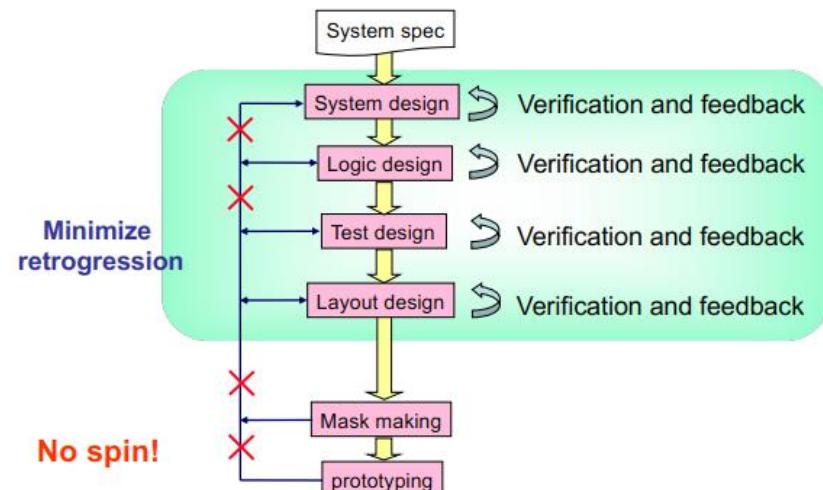
- **System design**
- **Logic design**
- **Test design**
- **Layout design**
- **Mask making**
- **Prototyping**

2. Verification and Feedback:

Mỗi bước đều có mũi tên cong thể hiện vòng

Verification và Feedback — để đảm bảo thiết

kế đúng trước khi sang bước tiếp theo.



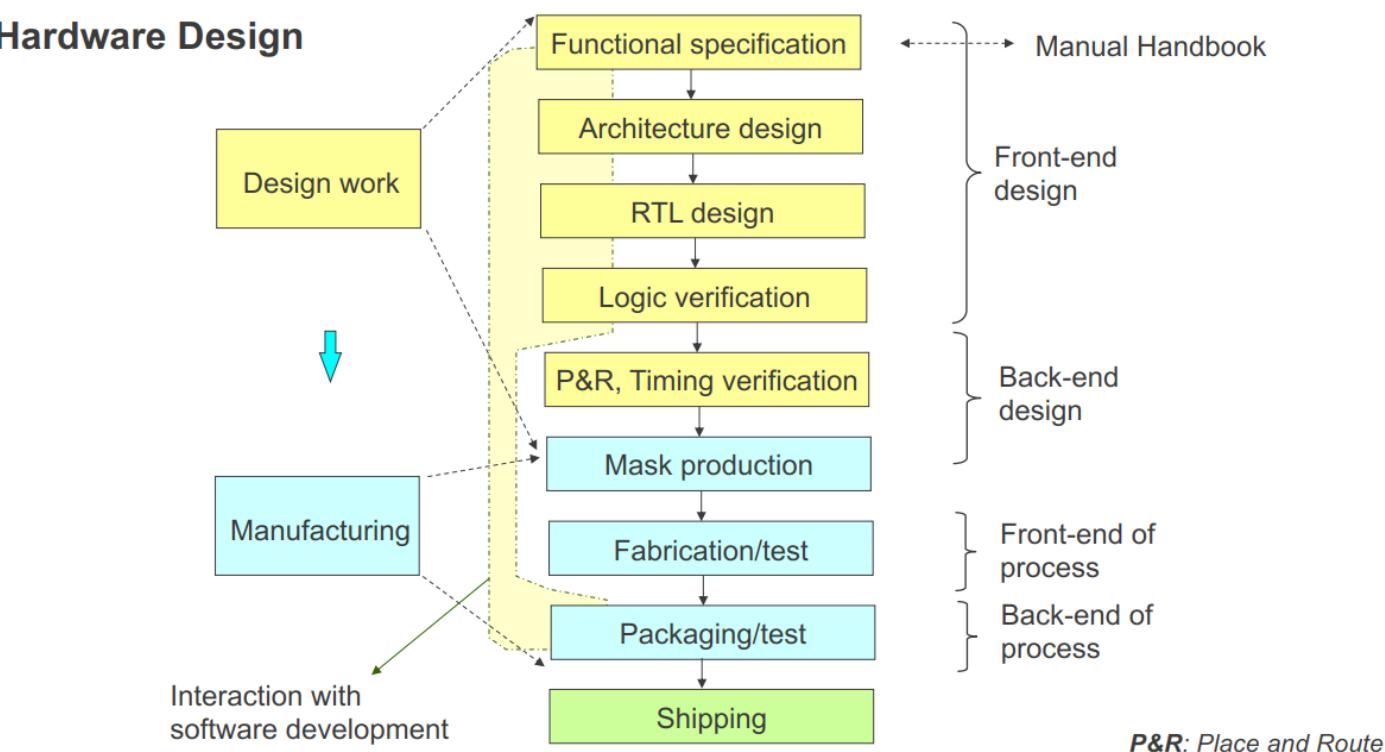
3. Minimize Retrogression (giảm việc quay lui):

- Các dấu “X đỏ” và mũi tên quay ngược thể hiện các trường hợp quay lại bước trước đó khi phát hiện lỗi.
- **Mục tiêu là giảm tối thiểu các lần phải quay lại (“retrogression”)**, đặc biệt là sau khi đã đến các bước gần cuối như Mask Making.

4. No Spin!:

- “Spin” nghĩa là việc **lặp lại mask making và prototyping** do phát hiện lỗi muộn.
- Điều này rất **tốn kém và mất thời gian**, nên khẩu hiệu “**No spin!**” nhấn mạnh rằng ta cần kiểm tra kỹ lưỡng trước khi bước vào chế tạo thực tế.

Hardware Design



Các giai đoạn chính trong quy trình thiết kế phần cứng:

1. Front-End Design (Thiết kế phía trước):

- **Functional specification:** Xác định chức năng của chip.
- **Architecture design:** Thiết kế kiến trúc tổng thể (pipeline, bus, memory...).
- **RTL design:** Thiết kế ở cấp độ register-transfer (thường dùng Verilog/VHDL).
- **Logic verification:** Xác minh chức năng logic đã đúng với yêu cầu ban đầu.

2. Back-End Design (Thiết kế phía sau):

- **Place & Route (P&R), Timing verification:** Gắn kết vị trí và định tuyến cell, sau đó kiểm tra timing (Static Timing Analysis).

3. Process Stages (Quy trình sản xuất):

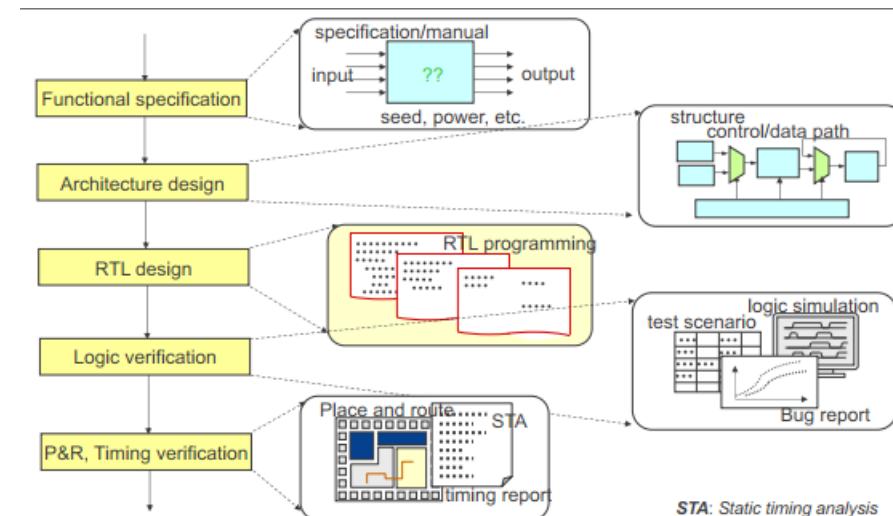
- **Mask production:** Tạo mặt nạ quang học để in mạch.
- **Fabrication/test:** Quá trình sản xuất silicon và kiểm tra.
- **Packaging/test:** Đóng gói chip và kiểm tra lại.
- **Shipping:** Gửi sản phẩm đến khách hàng.

4. Tương tác & Ghi chú:

- **Design work** tương tác với các bước phía trong để cung cấp thông tin kỹ thuật.
- **Manufacturing** có sự tương tác 2 chiều với thiết kế và quá trình đóng gói.
- **Manual Handbook:** Được dùng làm tài liệu chuẩn cho thiết kế frontend.
- **Interaction with software development:** Thiết kế phần cứng cần phối hợp với phát triển phần mềm, nhất là trong các hệ thống SoC.

◆ 1. Functional Specification

- **Nội dung:** Xác định rõ yêu cầu hệ thống (đầu vào, đầu ra, công suất, seed, v.v...).
- **Công cụ/đầu ra:** Tài liệu đặc tả (spec/manual), dùng làm cơ sở cho các bước tiếp theo



◆ 2. Architecture Design

- **Nội dung:** Thiết kế kiến trúc tổng thể: control unit, datapath, pipeline, bus structure...
- **Hình minh họa:** Sơ đồ cấu trúc gồm các khối xử lý dữ liệu và điều khiển.

◆ 3. RTL Design

- **Nội dung:** Viết code RTL (Verilog, VHDL), phản ánh kiến trúc hệ thống ở cấp độ register-transfer.
- **Hình minh họa:** Các đoạn mã RTL (màu đỏ), đại diện cho việc lập trình hành vi và cấu trúc.

◆ 4. Logic Verification

- **Nội dung:** Mô phỏng hành vi logic để phát hiện lỗi (logic simulation).
- **Công cụ:** Testbench, test scenario để tạo các trường hợp kiểm tra.
- **Kết quả:** Bug report nếu logic không đúng với yêu cầu.

◆ 5. Place & Route, Timing Verification

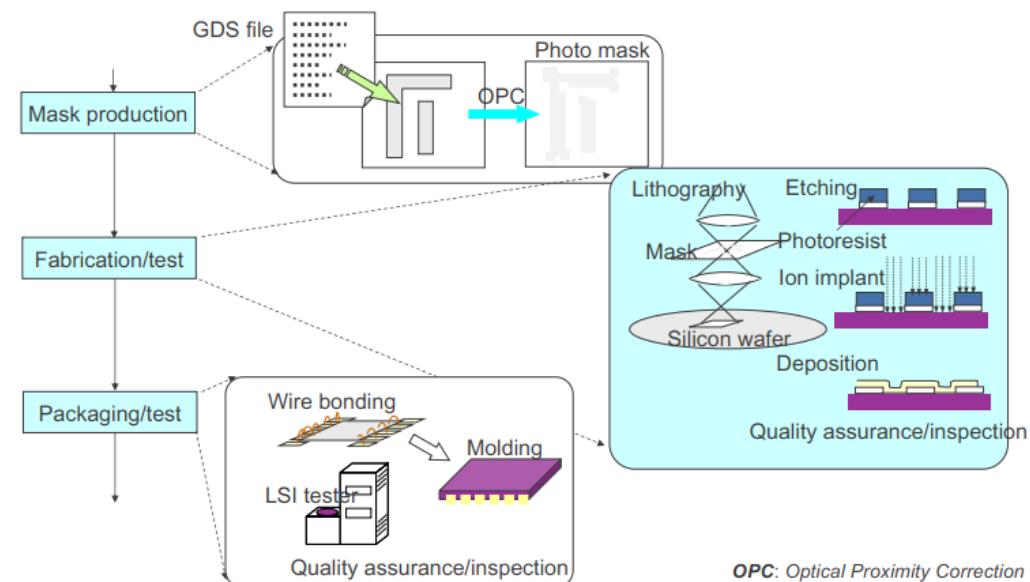
- **Nội dung:**
 - **Place & Route:** Gắn cell vào vị trí và định tuyến kết nối.
 - **Timing Verification:** Phân tích thời gian tĩnh (STA).
- **Kết quả:** Báo cáo thời gian (timing report) đảm bảo hệ thống đáp ứng yêu cầu về timing.

Ghi chú:

- STA (Static Timing Analysis) là công cụ then chốt trong kiểm tra độ trễ/timing của mạch.
- Quy trình này thường được chia thành hai phần:
 - Front-end: Functional Spec → RTL Design → Logic Verification
 - Back-end: P&R → STA

◆ 1. Mask Production

- **GDS file:** File thiết kế layout chuẩn (Graphical Data System) được xuất từ công đoạn P&R.
- **OPC (Optical Proximity Correction):** Kỹ thuật hiệu chỉnh sai số do hiện tượng quang học khi in mẫu trên wafer (đặc biệt quan trọng với node nhỏ).
- **Photo mask:** Tấm mặt nạ quang học dùng để chiếu mẫu thiết kế lên lớp photoresist.



◆ 2. Fabrication/Test

- **Lithography:** Chiếu ánh sáng qua mask để in hình thiết kế lên lớp photoresist trên **silicon wafer**.
- **Etching:** Loại bỏ lớp không cần thiết để tạo hình mong muốn.
- **Ion implantation:** Nhúng ion vào wafer để điều chỉnh tính chất bán dẫn.
- **Deposition:** Phủ lớp vật liệu cần thiết như kim loại hoặc oxit.
- **Quality assurance/inspection:** Kiểm tra sau mỗi bước để phát hiện lỗi sớm.

◆ 3. Packaging/Test

- **Wire bonding:** Kết nối die (chip) với chân package qua dây dẫn siêu nhỏ.
- **Molding:** Bọc đóng gói chip để bảo vệ.
- **LSI tester:** Thiết bị kiểm tra tính đúng đắn và hiệu suất của IC.
- **Quality assurance/inspection:** Kiểm tra cuối cùng trước khi xuất xưởng.

Ghi chú thêm:

- Giai đoạn này thuộc phần **hậu kỳ của quy trình sản xuất (back-end of process)**.
- Các công đoạn kiểm tra (inspection) diễn ra lặp lại để đảm bảo **không có lỗi chip**, giảm chi phí hỏng hóc sau này.

Software Design

Software Design Flow

◆ Các bước trong thiết kế

phần mềm:

1. **Functional specification**
 - Xác định rõ chức năng phần mềm cần thực hiện.
 - Có thể liên kết với **Manual Handbook** (hướng dẫn sử dụng).

- Architecture design:** Thiết kế tổng thể kiến trúc phần mềm (module, luồng dữ liệu, interface...).
- Logic design:** Chi tiết hóa logic xử lý, thuật toán, điều kiện rẽ nhánh.
- Programming:** Lập trình thực thi logic đã thiết kế.
- Logic verification**
 - Mô phỏng (Simulation) hoặc
 - Test trên thiết bị thực (Test on actual device).
 - Mục tiêu: đảm bảo phần mềm chạy đúng, hiệu quả.

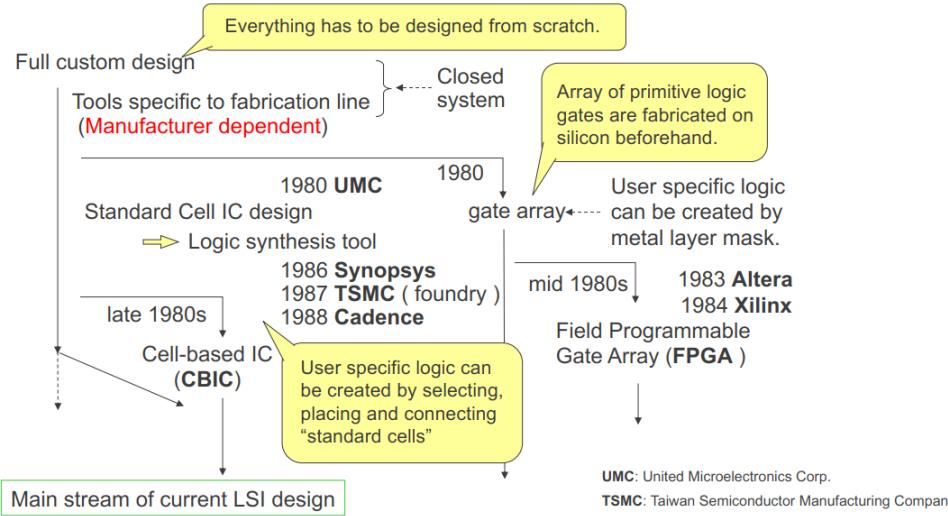
◆ Các bước sau xác minh:

- Packaging/Test:** Tích hợp, đóng gói và kiểm thử lại lần cuối.
- Shipping:** Gửi đến khách hàng hoặc triển khai.

▣ Tương tác với phần cứng:

- Trong quá trình lập trình, thiết kế phần mềm cần **tương tác với quy trình thiết kế phần cứng** (từ sơ đồ phần cứng, driver, thời gian phản hồi,...).
- Điều này giúp đảm bảo phần mềm phù hợp với cấu trúc phần cứng đích và tăng tính ổn định của sản phẩm cuối.

History of LSI Design Methodologies



So sánh thiết kế LSI cũ (Old LSI Design) và thiết kế LSI hiện đại (Current LSI Design)

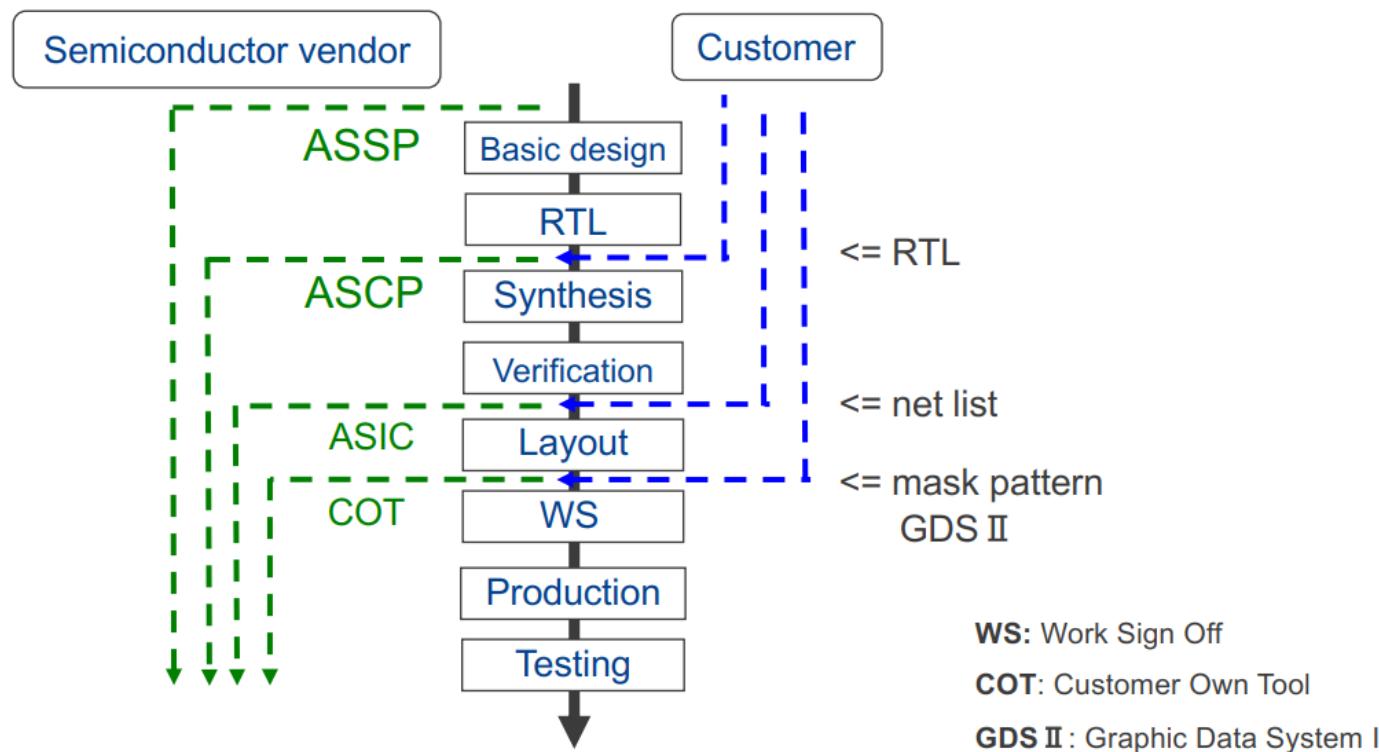
Tóm tắt sự khác biệt chính:

Tiêu chí	Thiết kế LSI cũ	Thiết kế LSI hiện tại
Tổng quan	Phụ thuộc vào nhà sản xuất, hệ thống đóng	Dựa trên Cell-base, hệ thống mở
Chức năng thiết kế	Dùng công cụ riêng của nhà sản xuất	Dùng ngôn ngữ mô tả phần cứng (C, Verilog, VHDL...)
Tạo logic mức cổng	Mapping thủ công, tối ưu hóa tay do công cụ tổng hợp yếu	Tự động mapping vào cell chuẩn, tối ưu hóa với công cụ mạnh
	Cổng không tương thích giữa các dây chuyền	Cổng có thể tương thích giữa các dây chuyền
Tạo mẫu MASK	Làm thủ công do công cụ layout yếu	Tự động tối ưu với công cụ layout mạnh
Sản xuất LSI	Chỉ chạy được trên dây chuyền cụ thể, không tương thích	Có thể sản xuất trên nhiều dây chuyền tương thích của các vendor khác nhau

Nhận định:

- Thiết kế hiện đại giúp tăng năng suất, giảm lỗi thiết kế thủ công, và dễ dàng chuyển đổi vendor sản xuất.
- Hệ sinh thái mở cho phép sử dụng các công cụ tiêu chuẩn, giảm chi phí phát triển và tăng khả năng tái sử dụng thiết kế.

Classification of IC from the view point of design



Phân loại các loại IC theo quan điểm thiết kế (Classification of IC from the view point of design), thể hiện mức độ tham gia giữa khách hàng (Customer) và nhà cung cấp bán dẫn (Semiconductor Vendor) qua từng loại sản phẩm IC: ASSP, ASCP, ASIC, COT.

Các loại IC và mức độ đóng góp:

Loại IC	Viết tắt	Thiết kế do ai đảm nhiệm	Đầu vào từ phía khách hàng
ASSP	Application Specific Standard Product	Hầu hết do vendor làm (từ basic design đến testing)	Không cần đầu vào
ASCP	Application Specific Custom Product	Vendor làm synthesis, verification, layout,...	Khách hàng cung cấp RTL
ASIC	Application Specific IC	Vendor làm layout trở đi	Khách hàng cung cấp netlist
COT	Customer Own Tool	Khách hàng làm mọi thứ tới mask pattern (GDS II)	Khách hàng cung cấp GDS II (hoặc mask)

Luồng công việc tương ứng:

- ASSP:** Khách hàng mua sẵn sản phẩm, không tham gia thiết kế.
- ASCP:** Khách hàng chỉ cần thiết kế RTL, các bước còn lại do vendor xử lý.
- ASIC:** Khách hàng cần thực hiện synthesis + verification, cung cấp netlist cho vendor.
- COT:** Khách hàng tự chịu trách nhiệm từ đầu đến file **GDS II**, chỉ sử dụng vendor cho sản xuất & test.

Ghi chú:

- WS (Work Sign Off):** Điểm kiểm tra/ký duyệt hoàn tất trước khi sản xuất.
- COT (Customer Own Tool):** Mô hình mà khách hàng dùng công cụ riêng để tự thiết kế IC.
- GDS II:** Định dạng tiêu chuẩn cho layout IC – đầu vào cuối cùng cho quá trình tạo mask.

