

30/8/2013

Friday, August 30, 2013

3:14 PM

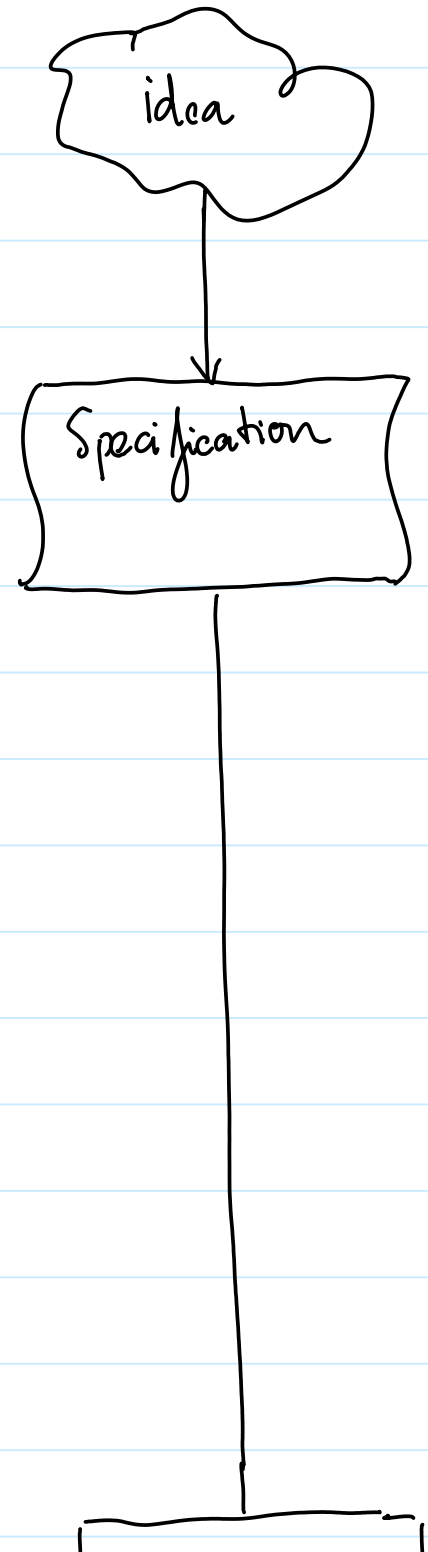
1.3. Quy trình thiết kế IC số, hệ thống số sử dụng ngôn ngữ mô tả phần cứng (Phân tích xây dựng tài liệu kỹ thuật; Mô tả hoạt động mạch bằng HDL; Kiểm tra chức năng logic; Tổng hợp; Phân tích kiểm tra tốc độ; Tối ưu mạch ở mức RTL; Thiết kế vật lý - Layout - Place&Route; Tách tham số thời gian; Phân tích thời gian sau layout) – 2 LT

- Ý tưởng sản phẩm xuất phát từ:
  - Giám đốc
  - Khách hàng
  - Phòng phân tích thị trường
- Được mô tả bằng ngôn ngữ tự nhiên trong các văn bản như slides, bản đồ tư duy, bảng tính:
  - Mô tả chức năng cần thiết của vi mạch; các chuẩn công nghiệp cần tuân thủ
  - Ước lượng giá thành của vi mạch
  - Khách hàng và hệ thống ứng dụng mục tiêu (application scenarios)

#### 1.3.1 Specification

Với sự giúp đỡ của đội ngũ kỹ thuật cao cấp, các ý tưởng được cụ thể hóa thành các văn bản tài liệu specification trong đó mô tả cụ thể hơn các chỉ tiêu kỹ thuật cơ bản của mạch:

- Chức năng (hoạt động của mạch)  
Mô tả đặc điểm của đầu ra, đầu vào và hàm quan hệ giữa đầu ra và đầu vào.  
Mối quan hệ được mô tả dưới dạng hành vi (nếu ... thì, dùng hàm toán học trừu tượng, lưu đồ thuật toán đơn giản)  
Các chuẩn vào ra mạch cần tuân thủ (Chuẩn H264, Chuẩn Bluetooth 4.0, WLAN 802.11n)
- Diện tích của hệ thống (vi mạch) được tính toán từ một số thông số:
  - Giá thành mục tiêu
  - Công nghệ chế tạo (90nm, 180nm, 65nm,...): kích thước của tấm wafer
  - Yield: tỉ lệ vi mạch chế tạo có thể hoạt động
  - Non Recurrence Cost: Design cost, Mask cost, IP cost, Standard Cell Library Cost, ...
  - Giá thành chế tạo
- Hiệu năng (Tốc độ hoạt động-tần số đồng hồ MHz; thông lượng dữ liệu - Mbit/s): Được tính toán dựa trên tốc độ lấy mẫu và số bit lấy mẫu đầu vào hoặc dựa vào tốc độ của môi trường hoạt động của vi mạch, dựa vào chuẩn mà vi mạch cần tuân theo)
- Năng lượng tiêu thụ (W, mW, A, mA): Được tính toán dựa vào kích thước hệ thống để suy ra dung lượng pin)



thước hệ thống để suy ra dung lượng pin)

### 1.3.2. Architecture Design (Optional)

Ở bước thiết kế kiến trúc, đội ngũ thiết kế sẽ tạo ra tài liệu mô tả kiến trúc của hệ thống. Trong đó:

- lựa chọn thuật toán có thể đáp ứng được các yêu cầu của specification,
- ánh xạ các thao tác trong thuật toán vào các khối phần cứng,
- lựa chọn mô hình thông tin để kết nối các khối phần cứng với nhau (kiến trúc pipeline, kiến trúc bus)

Tài liệu mô tả kiến trúc sẽ bao gồm:

- Mô tả thuật toán
- Hình vẽ kiến trúc mạch (gồm các khối nối với nhau)
- Mô hình kết nối giữa các khối: Giao thức trao đổi dữ liệu giữa các khối, Sự đồng bộ giữa các khối, Tốc độ truyền dữ liệu giữa các khối
- Mô tả hoạt động của từng khối (Đầu vào/Đầu ra/Hàm truyền đạt của khối)

### 1.3.3 Test plan

Đồng thời với việc thiết kế kiến trúc, cần lên kế hoạch để kiểm chứng (verification) và kiểm tra vi mạch. Kết quả của bước này là một văn bản test plan bao gồm:

- Phương pháp và công cụ kiểm tra: (Dùng mô phỏng - simulation, dùng công cụ hình thức toán học - formal verification, dùng phỏng tạo trên phần cứng - hardware emulation, ...)
- Ngôn ngữ nào: SystemVerilog, C, Hardware-In-Loop
- Kiểm tra cái gì: Những chức năng của chip nào cần kiểm tra, những trường hợp hoạt động nào cần kiểm tra, ...

### 1.3.4. Design Description

Các kỹ sư thiết kế dựa vào thiết kế kiến trúc để mô tả hoạt động của từng khối trong vi mạch bằng ngôn ngữ mô tả phần cứng VHDL hoặc Verilog.

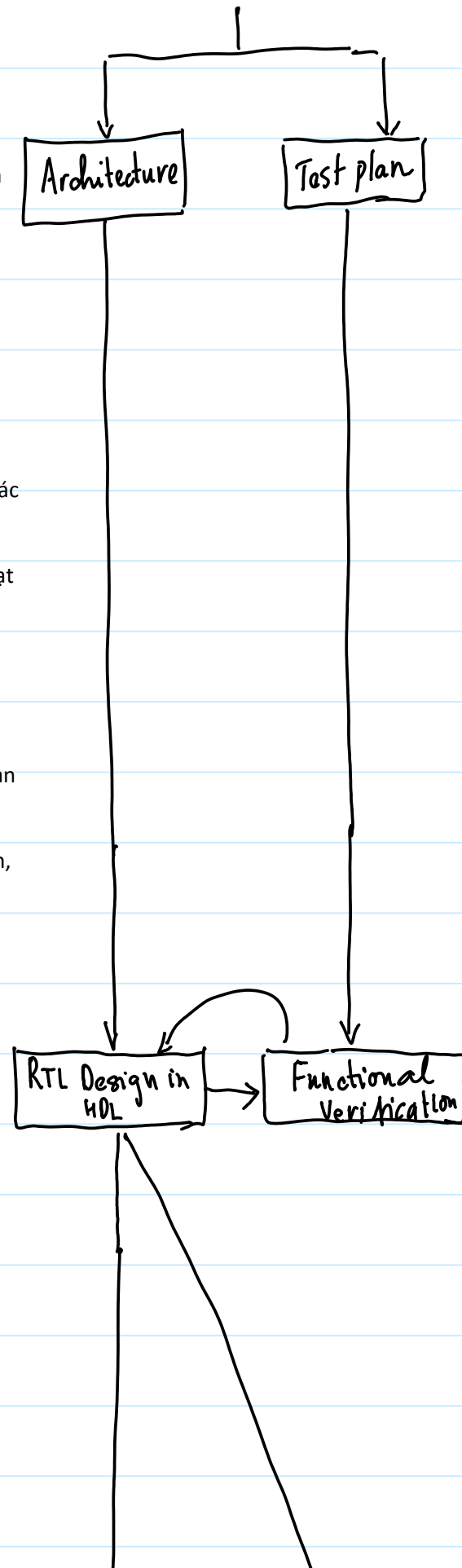
Kết quả là các mã HDL mô tả hoạt động của vi mạch.

### 1.3.5. Functional Verification

Các kỹ sư thiết kế & các kỹ sư kiểm chứng sử dụng tài liệu trong test plan để kiểm tra chức năng vi mạch nhằm đảm bảo: Vi mạch (khối) thực hiện đúng hoạt động đề ra ở specification và architecture.

Kết quả của bước này là:

- Mã HDL không còn lỗi (không còn tìm thấy lỗi sau thời gian



Kết quả của bước này là:

- Mã HDL không còn lỗi (không còn tìm thấy lỗi sau thời gian kiểm tra)
- Các báo cáo về lỗi, sự sửa chữa lỗi, báo cáo về chất lượng kiểm tra.

### 1.3.6. Synthesis (Tổng hợp)

Sử dụng phần mềm tổng hợp logic để chuyển đổi từ thiết kế HDL thành mạng cổng logic (logic gate netlist).

Đầu vào của phần mềm tổng hợp:

- Thiết kế HDL
- Giới hạn về chỉ tiêu thiết kế (Kích thước, Tốc độ)
- Thư viện công nghệ chế tạo (Standard Cell Library - ASIC, chip FPGA)

Đầu ra của phần mềm tổng hợp

- Mô tả thiết kế ở mức cổng logic (gate netlist)
- Báo cáo về việc đáp ứng (hoặc không đáp ứng) các yêu cầu thiết kế
  - Tốc độ
  - Diện tích

### 1.3.7. Equivalence Checking (Implementation Checking) - Kiểm tra triển khai, kiểm tra tương đương

Bước này đảm bảo bản thiết kế mạch ở mức cổng và bản thiết ở mức HDL là tương đương: Cùng đầu vào thì có đầu ra giống hệt nhau. Có thể dùng mô phỏng hoặc dùng formal verification

### 1.3.8. Floorplanning

Bước này thực hiện việc hoạch định (quy hoạch) diện tích vi mạch. Xác định kích thước chip, vị trí đặt các khối chức năng trong chip

Xác định các vùng đặt đầu vào, đầu ra, vùng cấp nguồn VDD, GND (power ring)

Floorplanning được thực hiện bằng tay dựa vào kết quả tổng hợp là kích thước của từng khối trên chip

### 1.3.9. Place & Route

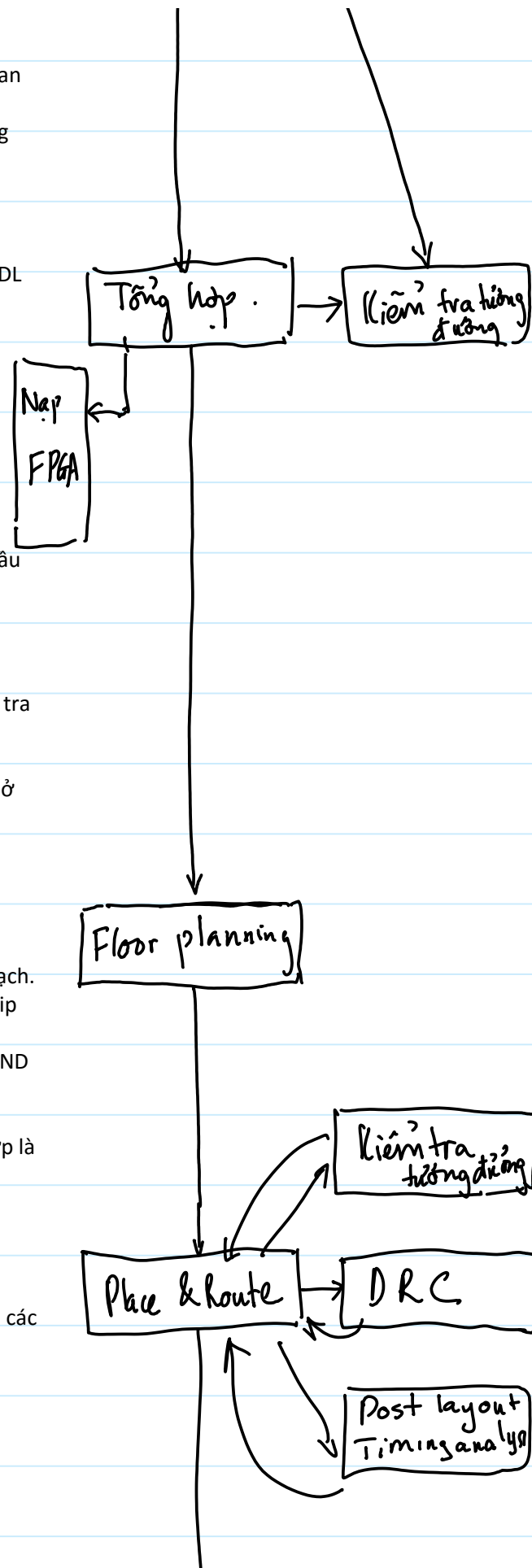
Bước này sử dụng phần mềm để tự động đặt các standard cell tương ứng với các cổng logic và các phần tử khác của mạch lên các diện tích đã được quy hoạch trên chip.

Đầu vào của phần mềm là:

- Mô hình gate netlist của mạch
- Thư viện standard cell
- Mô tả floor planning

Đầu ra của phần mềm là:

- Bản mô tả thiết kế layout của vi mạch



- vào từ floor planning

Đầu ra của phần mềm là:

- Bản mô tả thiết kế layout của vi mạch

1.3.10. Equivalence Checking: Đảm bảo sự tương đương của layout design và gate netlist

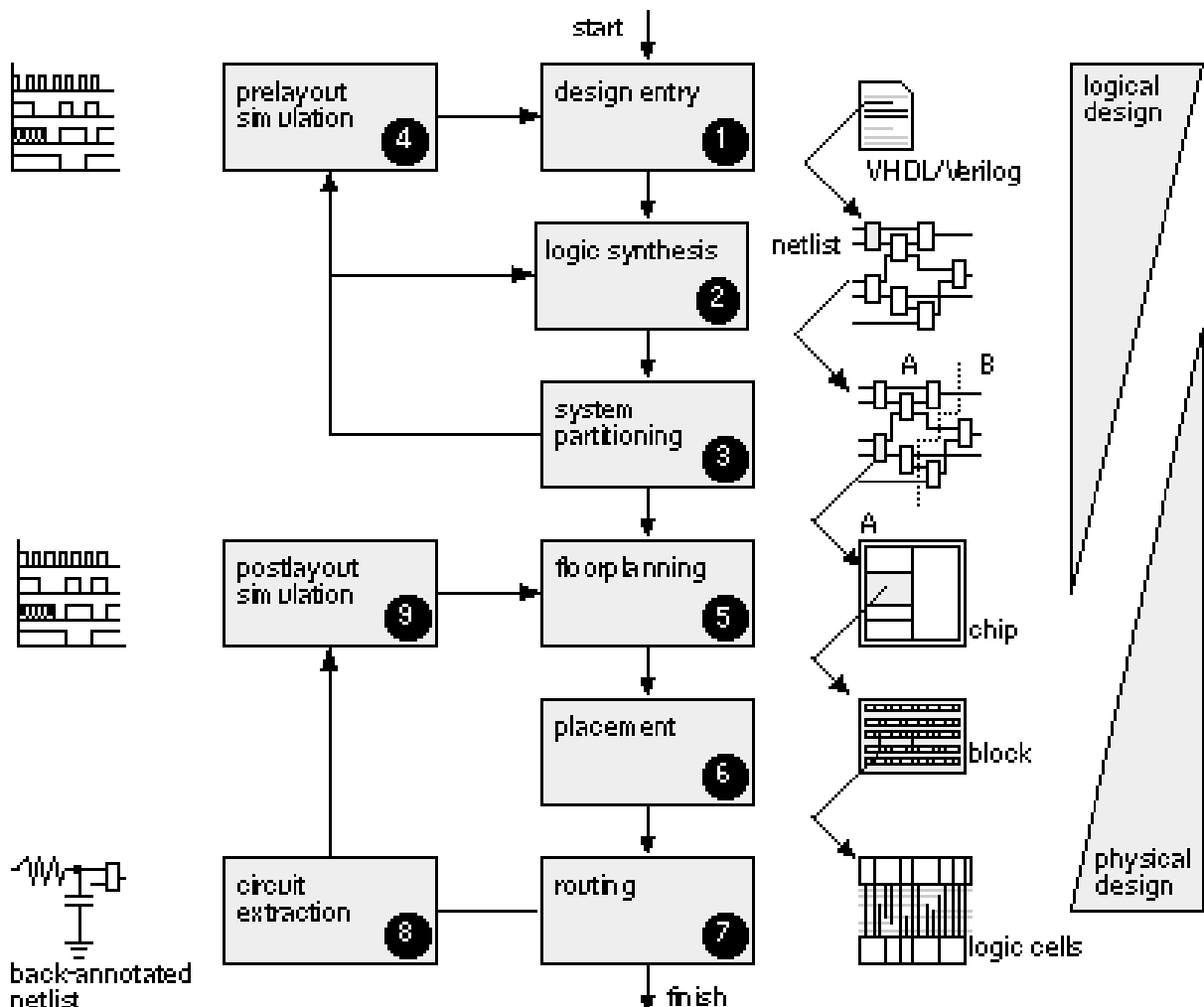
1.3.11. Design Rule Check: Đảm bảo các luật thiết kế không bị vi phạm

1.3.12. Post-layout Timing analysis: Phân tích lại tốc độ mạch

Sản xuất.

Tóm lại, quy trình thiết kế vi mạch số sử dụng HDL có thể phân chia làm 3 giai đoạn chính:

- 1) High level design: Idea, Specification, Architecture, Testplan. Sử dụng các công cụ
  - a. Powerpoint
  - b. Mindmap
  - c. Excel
  - d. Word
  - e. Ngôn ngữ bậc cao: C/C++, SystemC, Matlab/Simulink, ...
- 2) Logical Design (Front-end Design). Sử dụng các công cụ
  - a. Text-editor: emacs, gedit, eclipse
  - b. Simulator/Verifier: Modelsim, Synopsys VCS
  - c. Synthesizer: Synopsys Design Compiler, Xilinx ISE, Altera Quartus
- 3) Physical Design (Back-end Design). Sử dụng công cụ
  - a. Layout tool: Synopsys IC Compiler, ...



DDAS:

- Cung cấp kiến thức chủ yếu về: HDL Design, Functional Verification, Design Synthesis.
- Kiến thức (qua ví dụ) về: Specification, Architecture Design
- Kiến thức cơ bản về Equivalence Checking (qua ví dụ)

1.4. Giới thiệu về ngôn ngữ mô tả phần cứng HDL (Khái niệm; Ưu nhược điểm; Ví dụ đơn giản; Khái niệm về module; port; Chú thích; Khái niệm về mô hình cấu trúc, mô hình hành vi và mô hình dòng dữ liệu)

#### 1.4.1. Khái niệm

Hardware Description Language gồm 2 loại chính VHDL (Very Large Scale IC Hardware Description Language và Verilog) được sử dụng để mô tả hoạt động của các mạch số. HDL là ngôn ngữ chính xác gần ngôn ngữ tự nhiên được sử dụng để mô tả hoạt động, cho phép mô phỏng hoạt động của mạch và tổng hợp mạch thành dạng gate netlist.

- Mục đích: Ngôn ngữ HDL dùng để mô hình hóa và mô phỏng (có thời gian) thiết kế vi mạch số. Bản thiết kế vi mạch bằng ngôn ngữ HDL có thể tổng hợp thành mạch bằng các công cụ tổng hợp (Ví dụ: Synopsys Design Compiler; Altera Quatus; Xilinx ISE; Cadence, ...). Có thể mô tả mạch bằng ngôn ngữ gần với tự nhiên
- Phân loại:
  - Verilog: Mềm dẻo, sử dụng nhiều trong công nghiệp
  - VHDL: Sử dụng trong quốc phòng và thiết kế vi mạch trong ô tô; Hướng kiểu mạnh
- Ưu điểm:
  - Nâng cao năng suất thiết kế vi mạch
  - Có thể thực hiện được - có thể mô phỏng được
  - Một bộ phận của ngôn ngữ có thể tổng hợp được - có thể biến đổi thành mạch cổng logic => nâng cao năng suất thiết kế vi mạch
  - Một thiết kế HDL được chú giải tốt có thể sử dụng như một tài liệu bổ xung với specification và architecture design để mô tả hoạt động của vi mạch
  - Dễ dàng sửa đổi, gỡ lỗi, cập nhật
  - Dễ dàng tái sử dụng (mua, bán, chuyển giao)