



Tổng hợp luận lý vi mạch



Giới thiệu

- Môn học: Tổng hợp luận lý vi mạch
 - Mã số:
 - Số tín chỉ: 3
- Giảng viên:
 - Dr. Tran Ngoc Thinh
 - Email: tnthinh@cse.hcmut.edu.vn
 - Phone: 8647256 (5843)
 - Office: A3 building,
 - Office hours: Tuesdays, 9:30-11:00



Giới thiệu môn học

- Nội dung:
 - Giới thiệu về thiết kế vi mạch
 - Biến đổi từ các đặc tả ngôn ngữ HDL
 - Nắm quy trình tổng hợp, kiểm tra, kiểm thử các mạch 2 lớp và nhiều lớp
- Đối tượng:
 - Sinh viên cao học
 - Sinh viên đại học năm 3-4
- Đánh giá
 - Tiểu luận: 30%
 - Trình bày: 30%
 - Kiểm tra cuối kỳ: 40%



Nội dung môn học

- Tổng quan về thiết kế luận lý vi mạch
- Mạch tổ hợp 2 mức
- Tổng hợp mạch tổ hợp 2 mức
- Kiểm thử mạch tổ hợp 2 mức
- Mạch tổ hợp nhiều mức
- Tổng hợp mạch tổ hợp nhiều mức
- Kiểm thử mạch tổ hợp nhiều mức



Tài liệu tham khảo

- [1]. Logic Synthesis – Srinivas Devadas, Abhijit Ghosh, Kurt Keutzer
- [2]. Synthesis and Optimization of Digital Circuits – Giovanni De Micheli

Chương 1: Giới thiệu tổng quan

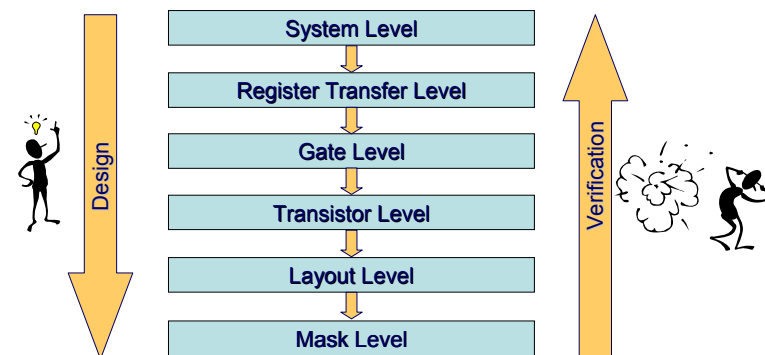


- Tổng quan
- Mạch tổ hợp 2 mức
- Mạch tổ hợp nhiều mức
- Kiểm tra (verification)
- Kiểm thử (testing)

Tổng quan

- Mạch số tích hợp
- Phương pháp thiết kế IC
- Bố trí mức transistor
- Mức cổng
- Initial use of Logic Optimizaion
- Emergence of Synthesis-Base Design
- A Logic Synthesis Design Methodology

Design of Integrated Circuits



System Level

- Abstract algorithmic description of high-level behavior
 - e.g. C-Programming language

```
Port*
compute_optimal_route_for_packet(Packet_t *packet,
                                Channel_t *channel)
{
    static Queue_t *packet_queue;

    packet_queue = add_packet(packet_queue, packet);
    ...
}
```

- abstract bởi vì nó không chứa bất cứ hiện thực chi tiết nào cho thời gian hoặc data
- Hiệu quả để đạt được một mô hình thực thi rút gọn ở giai đoạn đầu tiên của thiết kế
- Khó khăn để duy trì sự xuyên suốt dự án bởi vì không liên kết đến phần hiện thực



RTL Level

```
module mark1;
reg [31:0] m[0:8192];
reg [12:0] pc;
reg [31:0] acc;
reg [15:0] ir;

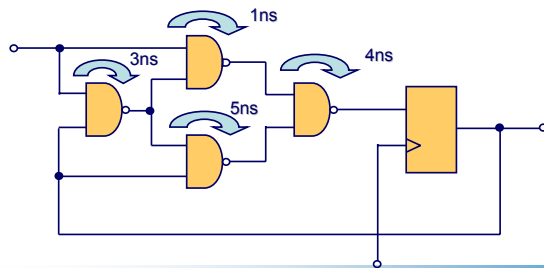
always
begin
    ir = m[pc];
    if (ir[15:13] == 3'b000)
        pc = m[ir[12:0]];
    else if (ir[15:13] == 3'b010)
        acc = -m[ir[12:0]];
    ...
end
endmodule
```

- Cycle accurate model “close” to the hardware implementation
 - Những kiểu dữ liệu bit-vector và cách hoạt động được trừu tượng hóa từ hiện thực bit-level
 - Cấu trúc tuần tự (e.g. if - then - else, while loops) để ủng hộ mô hình dòng điều khiển phức tạp.



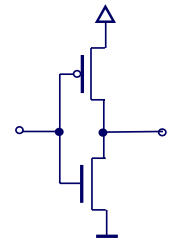
Gate Level

- Model on finite-state machine level
 - Những mô hình chức năng trong luận lý Boolean sử dụng registers và gates
 - Các mô hình delay khác nhau cho gates và wires



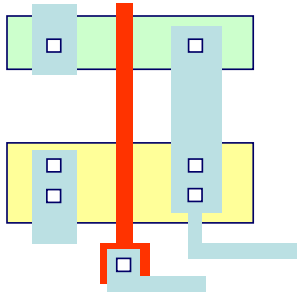
Transistor Level

- Model on CMOS transistor level
 - Được dùng trong kiểm tra sự tương đương chức năng
 - Hoặc phân tích chi tiết về timing



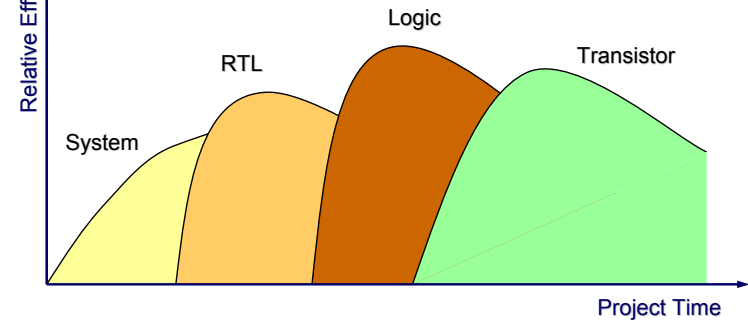
Layout Level

- Transistors và wires được đặt như các polygons trong các lớp khác nhau như diffusion, poly-silicon, metal, etc.



Design of Integrated Systems

- Design phases overlap to large degrees
- Parallel changes on multiple levels, multiple teams
- Tight scheduling constraints for product



Design Challenges

- Systems đang trở nên lớn hơn, thời gian thiết kế ngày càng phải rút ngắn
 - > 100 Mio gates càng phổ biến cho ASICs
 - > 0.4 Mio dòng của C-code để mô tả hành vi hệ thống
 - > 5 Mio dòng của RLT code
- Design teams rất lớn cho những projects lớn
 - Vài trăm người
 - Kỹ năng khác nhau
 - Làm việc đồng thời ở nhiều mức khác nhau
 - Việc quản lý độ phức tạp và liên lạc thiết kế là rất khó khăn
- Design tools đang ngày càng phức tạp hơn nhưng vẫn chưa đáp ứng yêu cầu
 - Thông thường người thiết kế phải chạy khoảng 50 tools cho mỗi component
 - tools có nhiều lỗi, interfaces chưa thân thiện etc.

Design Challenges

- Quyết định về design point là rất khó
 - performance / costs / time-to-market
 - Quyết định phải được thực hiện 2-3 years trước khi hoàn thành thiết kế before design finished
 - Các thời điểm thiết kế rất khó đoán trước nếu không làm thật sự
 - Xác định vòng đời sản phẩm
- Functional verification
 - Mô phỏng vẫn là công cụ chính cho kiểm tra chức năng nhưng không đủ bởi vì kích thước của không gian thiết kế
 - Các lỗi trong released hardware thì rất đắt để khắc phục (khác với software)

Design Challenges

- Tradeoffs cơ bản giữa các mức mô hình hóa khác nhau:
 - **modeling detail and team size to maintain model**
 - high-level models có thể được kiểm soát bởi 1 hoặc 2 người
 - detailed models cần được phân hoạch mà kết quả là sự khó khăn trong việc liên lạc giữa các khối
 - **modeling accuracy versus modeling compactness**
 - compact models bỏ qua các chi tiết và đưa ra chỉ những ước lượng sơ khai cho hiện thực
 - detailed models thì kéo dài và khó khăn để thích nghi với những thay đổi trong thiết kế
 - **simulation speed versus hardware performance**
 - high-level models có thể mô phỏng nhanh nhưng không thể hiện thực hiệu quả với những phương tiện tự động
 - low-level models có thể hiện thực nhanh chóng nhưng khó mô phỏng

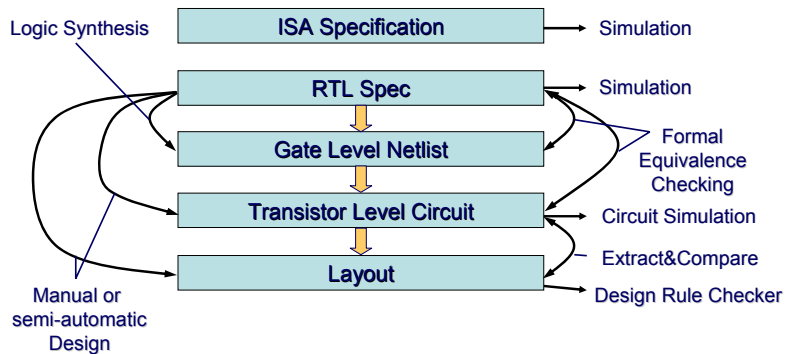


Full Custom Design Flow

- Application: những thiết kế hiệu suất cực cao
 - general-purpose processors, DSPs, graphic chips, internet routers, games processors etc.
- Target: thị trường lớn và lợi nhuận cao
 - e.g. PC business
- Complexity: rất cao và đòi hỏi lab hiện đại
 - Đội ngũ lớn, đầu tư cao và rủi ro cũng cao
- Vai trò Logic Synthesis:
 - Hạn chế chỉ cho components không chuẩn hóa về hiệu suất hoặc có thể thay đổi sau đó trong chu trình thiết kế
 - non-critical data paths logic và control logic
 - Một lượng lớn data-path components và fast control logic được thiết kế thủ công để tối ưu hiệu suất



Full Custom Design Flow

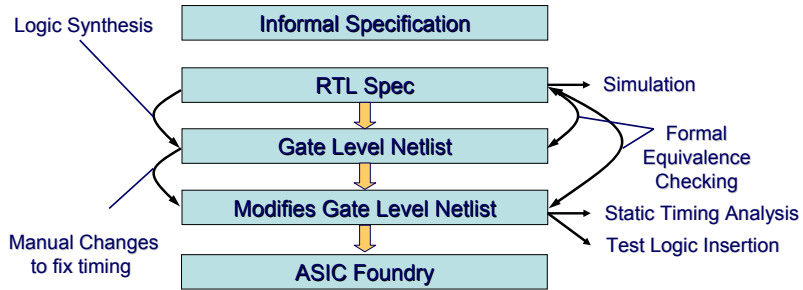


ASIC Design Flow

- Application: thị trường IC thông dụng
 - peripheral chips in PCs, toys, handheld devices etc.
- Target: thị trường trung bình và nhỏ, thời gian thiết kế hạn chế
 - e.g. consumer electronics
- Complexity of design: kiểu thiết kế chuẩn, đoán trước được
 - standard flows, standard off-the-shelf tools
- Vai trò Logic Synthesis:
 - Được sử dụng tỉ lệ cao trong thiết kế ngoại trừ những khối đặc biệt như RAM's, ROM's, analog components



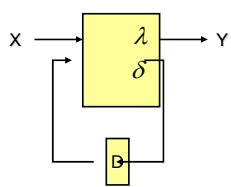
ASIC Design Flow



What is Logic Synthesis?

- **Definition:** To design a logic circuit such that it meets the specifications and can be economically manufactured:
 - Performance – đạt những yêu cầu về delay hoặc tối thiểu delay.
 - Cost – dùng hardware ít nhất, diện tích chip nhỏ nhất, số lượng gates hoặc transistors nhỏ nhất.
 - Power – đạt các yêu cầu về power, tiêu thụ power ít nhất.
 - Testability – không có dư thừa logic và dễ kiểm tra.

What is Logic Synthesis?



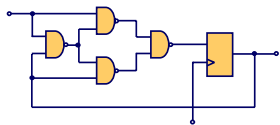
Given: Finite-State Machine $F(X, Y, Z, \lambda, \delta)$ where:

- X: Input alphabet
- Y: Output alphabet
- Z: Set of internal states
- $\lambda: X \times Z \rightarrow Z$ (next state function)
- $\delta: X \times Z \rightarrow Y$ (output function)

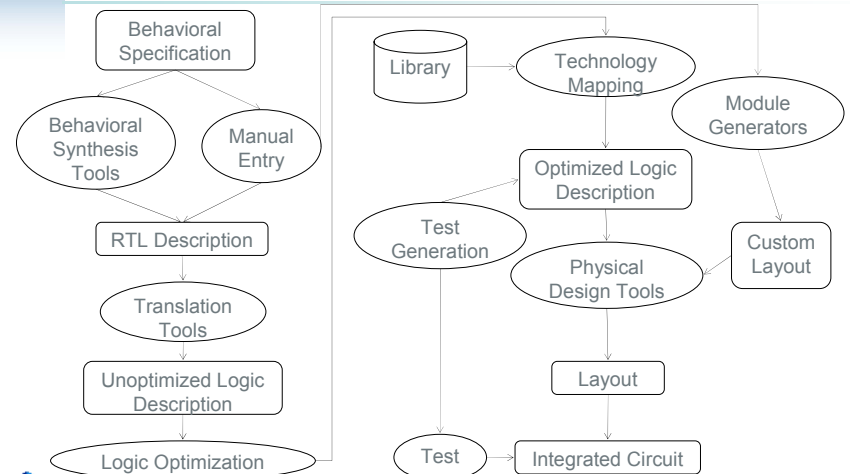


Target: Circuit $C(G, W)$ where:

- G: set of circuit components $g \in \{ \text{Boolean gates, flip-flops, etc} \}$
- W: set of wires connecting G



Quy trình thiết kế tổng hợp thông thường



Mục tiêu của Synthesis

- Minimize area
 - số lượng cell, register, etc.
- Minimize power
 - Hoạt động chuyển mạch trong individual gates, deactivated circuit blocks, etc.
- Maximize performance
 - Tối ưu clock frequency của synchronous systems, throughput của asynchronous systems
- Kết hợp các giải pháp trên
 - Kết hợp với những trọng số khác nhau
 - Công thức hóa như một vấn đề ràng buộc
 - “minimize area for a clock speed > 300MHz”
- More global objectives
 - feedback from layout
 - actual physical sizes, delays, placement and routing



Constraints on Synthesis

- Given implementation style:
 - Hiện thực hai mức (PLA, CAMs)
 - Hiện thực nhiều mức, FPGAs
- Given performance requirements
 - Yêu cầu tối thiểu của clock speed
 - Yêu cầu tối thiểu latency, throughput
- Given cell library
 - Tập các cells trong cell library
 - fan-out constraints (tối đa số gates được nối với gate khác)



Brief History of Logic Synthesis

- 1960s: first work on automatic test pattern generation used for Boolean reasoning
 - D-Algorithm
- 1978: Formal Equivalence checking introduced at IBM in production for designing mainframe computers
 - SAS tool based on the DBA algorithm
- 1979: IBM introduced logic synthesis for gate array based main frame designed
 - LSS, next generation is BooleDozer
- End 1986: Synopsys founded
 - first product “remapper” between standard cell libraries
 - later extended to full blown RTL synthesis
- 1990s other synthesis companies enter the marker
 - Ambit, Compass, Synplicity, Magma, Monterey, ...



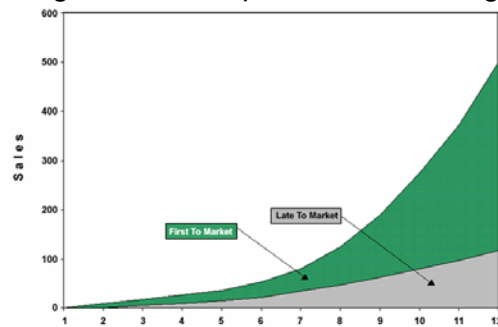
Why learning about Logic Synthesis?

- Logic synthesis là hạt nhân của các CAD tools ngày nay cho thiết kế IC và system.
 - Bao gồm nhiều giải thuật sử dụng rộng rãi trong CAD tools
 - Cơ bản cho các kỹ thuật tối ưu khác, e.g. embedded software
 - Cơ bản cho các kỹ thuật kiểm tra chức năng
- Phần lớn giải thuật là khó tính toán



Phương pháp thiết kế IC

- Hai tham số ràng buộc quá trình thiết kế IC:
 - Chất lượng của mạch
 - Thời gian đưa sản phẩm ra thị trường



Mạch số tích hợp

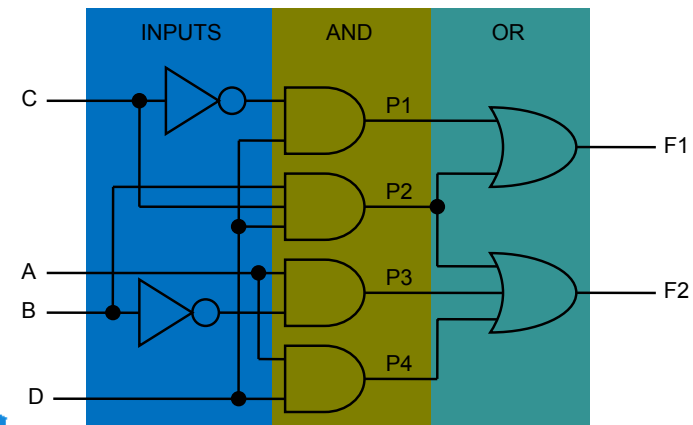
- Integrated circuit
 - Silicon compiler
- Transistor-level
 - Một tập các transistor và các kết nối được cấu hình trong một IC
 - Nor cell
- Gate-level
 - Được sử dụng từ những năm 1970

Mạch tổ hợp 2 mức

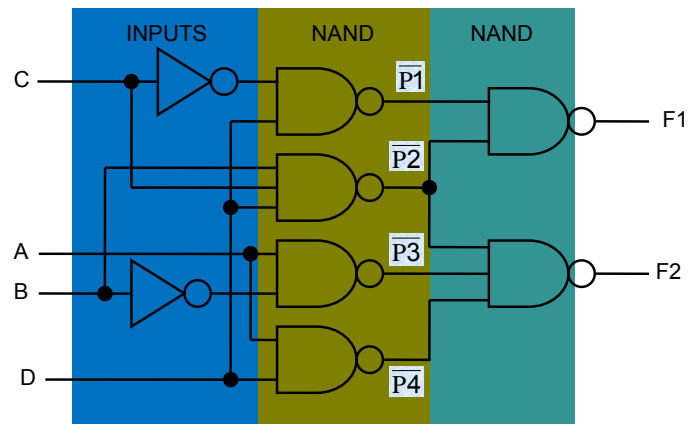
- Có nhiều dạng mạch logic 2 mức:
 - Sum-of-Product (AND-OR)
 - Product-of-Sum (OR-AND)
 - NOR-NOR
 - NAND-NAND
 - AND-XOR
 - ...

Two-Level AND-OR Implementation

- Technology-independent circuit.



NAND-NAND Implementation



Mạch tổ hợp 2 mức

- Ưu điểm của mạch 2 mức
 - PLA và mạch logic 2 mức có thể hiện thực hiệu quả các khối điều khiển logic.
 - Việc tối ưu dạng Sum-of-Product thường được sử dụng trong giai đoạn đầu tiên của quá trình tổng hợp nhiều mức
- Nhược điểm:
 - Nhiều hàm quy tắc có chung một dạng tối giản 2 mức
 - Có kích thước tăng theo cấp số nhân khi tăng số ngõ nhập

Mạch tổ hợp nhiều mức

- Thường được sử dụng nhiều hơn các mạch tổ hợp 2 mức
- Nhanh hơn và nhỏ hơn so với việc hiện thực của mạch 2 mức
- Mạch tổ hợp nhiều mức thường được thể hiện ở dạng mạng nhiều mức của các cổng (multi-level networks of logic gate) Cho phép tự do hơn trong việc thiết kế
 - Tối ưu diện tích
 - Delay
 - Thỏa mãn các ràng buộc (ví dụ: các yêu cầu về thời gian trên các đường xuất nhập khác nhau)
- Nhược điểm
 - Khó mô hình hóa và tối ưu mạng nhiều mức (multi-level networks)

Kiểm tra

- Kiểm tra tính chính xác của bản thiết kế
 - Được thực hiện bởi quá trình mô phỏng (simulating) hoặc kiểm tra hình thức (formal methods)
 - Mô phỏng là một kỹ thuật thường dùng nhất của quá trình kiểm tra
- Kiểm tra tính consistency giữa đặc tả hành vi và đặc tả RTL

Kiểm thử

- Nguyên nhân: Quá trình sản xuất sản xuất phức tạp và không hoàn hảo
- Mục đích:
 - Phát hiện lỗi của mạch
 - Phân loại chất lượng trước khi đưa vào thị trường
 - Phát hiện những kiểm khuyết trong quá trình sản xuất
- Các loại kiểm thử:
 - Fault detection
 - Fault Diagnosis



Kiểm thử

- Quy trình: Sử dụng các mẫu kiểm thử và so sánh kết quả
- Các loại lỗi
 - Lỗi vật lý (physical fault):
 - Lỗi tĩnh (static/logic fault)
 - Lỗi động (dynamic/parametric fault)



Kiểm thử

- Các mô hình lỗi (fault model): các mô hình được sử dụng trong quá trình sinh mẫu kiểm thử:
 - Static fault model: single stuck-at, multiple stuck-at, bridging fault
 - Dynamic fault model: gate delay fault, transistor stuck-open fault, path delay fault
 - Comprehensive fault model: các mô hình có chỉ định vị trí của lỗi.
 - Debug lỗi mạch
 - Sửa (fix) lỗi quy trình sản xuất



Tham khảo

- **Chapter 1, Logic Synthesis – Srinivas Devadas, Abhijit Ghosh, Kurt Keutzer**

