CHƯƠNG 6: TỔNG HỢP PHẦN CỨNG VÀ PHẦN MỀM

Bài 12: Công nghệ thiết kế

Tổng quan

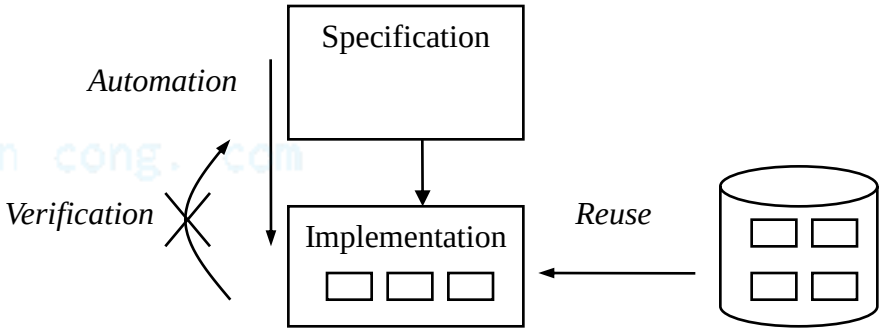
- Tự động: tổng hợp
- Kiểm thử: đồng mô phỏng phần cứng/phần mềm
- Sử dụng lại: nền tảng dựa trên sở hữu trí tuệ intellectual property (IP)
- Mô hình quá trình thiết kế

Giới thiệu

- Nhiệm vụ thiết kế
  - Định nghĩa chức năng của hệ thống
  - Biến đổi các chức năng thành việc thực hiện vật lý, trong khi phải
    - Đảm bảo các thông số ràng buộc
    - Tối ưu các thông số thiết kế khác
- Thiết kế hệ thống nhúng là một việc khó
  - Phức tạp về chức năng
    - Hàng triệu điều kiện làm việc khác nhau
    - Nhiều ràng buộc
  - Khoảng cách về tính sản xuất
    - Khoảng 10 dòng code hoặc 100 transistors được sản xuất mỗi ngày

Cải thiện tính sản xuất

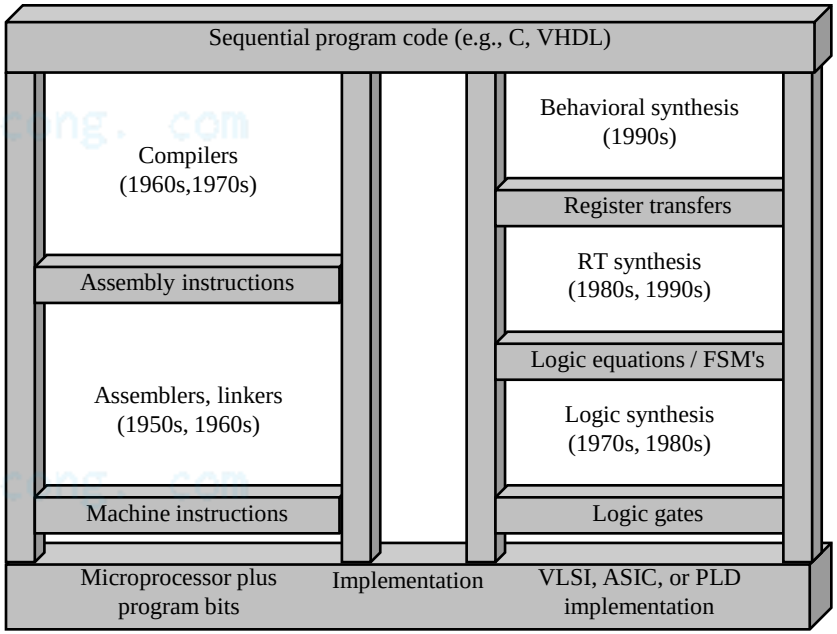
- Thiết kế các công nghệ để tăng tính sản xuất
- Chúng ta tập trung vào các công nghệ để đồng thiết kế phần cứng/phần mềm
  - Tự động
    - Các chương trình thay thế cho việc thiết kế thủ công
    - Tổng hợp
  - Tái sử dụng
    - Các bộ phần được thiết kế trước
    - Các lõi (Cores)
    - Bộ xử lý chức năng đơn và chức năng chung trên cùng một IC
  - Kiểm thử
    - Đảm bảo tính đúng đắn, tính hoàn thiện của mỗi bước thiết kế
    - Đồng mô phỏng phần cứng/phần mềm



Tự động: tổng hợp

- Các thiết kế trước chủ yếu là phần cứng
- Độ phức tạp về phần mềm tăng cùng với sự ra đời của bộ xử ý chức năng chung
- Các kỹ thuật khác nhau cho thiết kế phần cứng và thiết kế phần mềm
  - Tạo ra sự phân biệt giữa hai lĩnh vực
- Lĩnh vực thiết kế phần cứng và phần mềm tái hợp lại
  - Cả hai có thể được bắt đầu từ mức mô tả trạng thái của hệ thống nhúng
  - Quá trình này gọi là đồng thiết kế

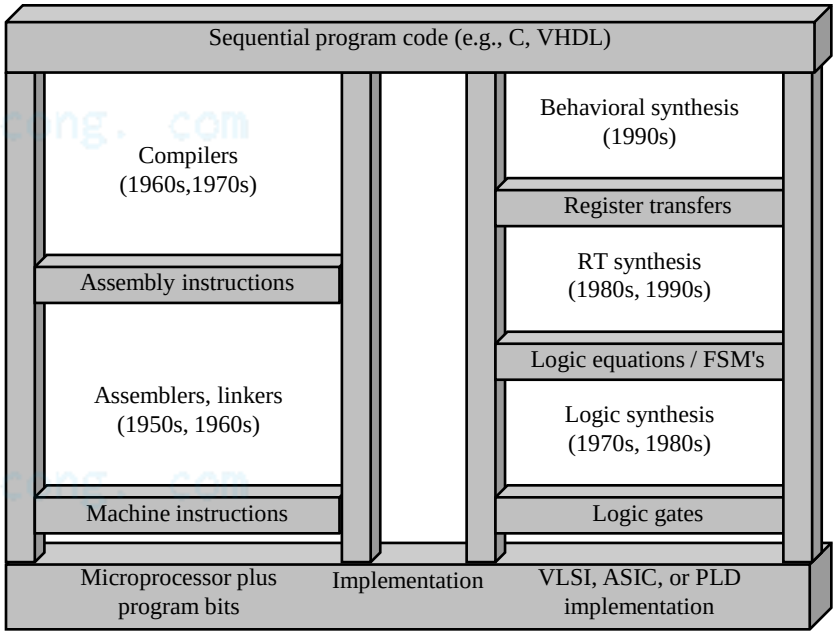
Đồng thiết kế



Tiến hóa song song của phần cứng và phần mềm

- Tiến hóa thiết kế phần mềm
  - Các lệnh máy
  - Assemblers
    - Biến đổi chương trình assembly thành mã máy
  - Compilers
    - Biến đổi chương trình tuần tự sang assembly
- Tiến hóa thiết kế phần cứng
  - Các cổng logic được kết nối
  - Tổng hợp logic
    - Biến đổi phương trình logic thành các cổng
  - Tổng hợp mức chuyển đổi thanh ghi (Register-transfer: RT)
    - Biến đổi FSMDs thành FSMs, phương trình logic, các thành phần RT được thiết kế trước (thanh ghi, bộ công, vv…)
  - Tổng hợp trạng thái
    - Biến đổi chương trình tuần tự thành FSMDs

Đồng thiết kế



Tổng hợp logic

- Trạng thái mức logic sang cấu trúc thực hiện
  - Phương trình logic và/hoặc FSM sang các cổng logic
- Tổng hợp logic tổ hợp
  - Tối ưu hai mức (Tổng các tích/tích các tổng)
    - Chất lượng tốt nhất có thể
      - Tuyến dài nhất = 2 cổng (cổng AND + cổng OR/cổng OR + cổng AND)
    - Kích cỡ tối thiểu
      - Tổi thiểu đầu vào/ra
  - Tối ưu nhiều mức
    - Bù chất lượng và kích thước
    - Tối ưu
      - Tìm kiếm tối ưu
- Tổng hợp FSM
  - Tối ưu trạng thái
  - Mã hóa trạng thái

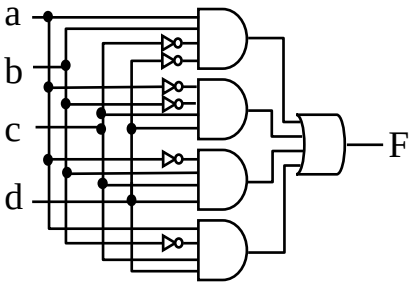
Tối ưu hai mức

- Diễn tả hàm logic dạng tổng của tích (hoặc tích của tổng)
  - Cổng AND cho mỗi tích
  - Cổng OR cho mỗi tổng
- Chất lượng tốt nhất có thể
  - Trễ tối đa qua hai cổng
- Mục tiêu: tối thiểu kích thước
  - Tối thiểu số cổng AND (tổng của các tích)
  - Tối thiểu số đầu vào của mỗi cổng AND (tổng của các tích)

Tổng các tích

F = abc'd' + a'b'cd + a'bcd + ab'cd

Thực hiện trực tiếp



4 4-input AND gates and
1 4-input OR gate

→ 40 transistors

Tối thiểu: Phương pháp Karnaugh

- Bản đồ Karnaugh (K-map)
  - 1 diễn tả minterm
  - Vòng tròn diễn tả các nhóm
- Tối thiểu
  - Hình bên

K-map: sum of products

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	0	0	0
10	0	0	1	0

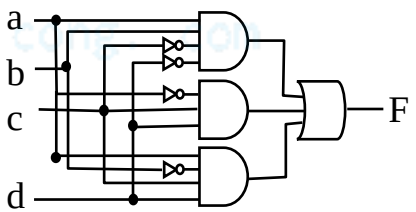
K-map: minimum cover

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	0	0	0
10	0	0	1	0

Minimum cover

F=abc'd' + a'cd + ab'cd

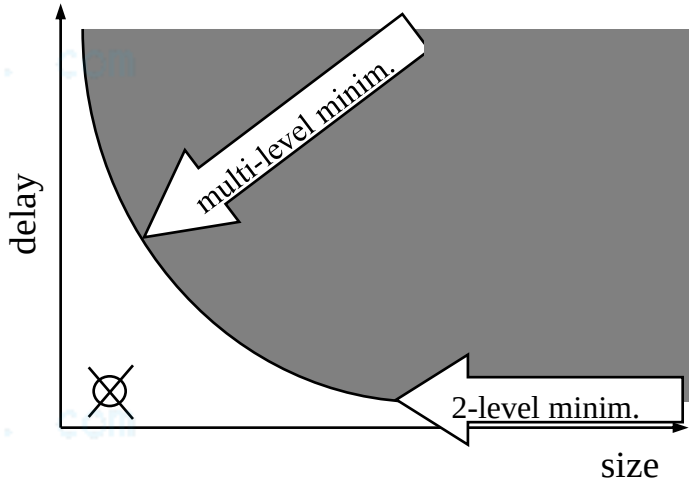
Minimum cover implementation



2 4-input AND gate
1 3-input AND gates
1 4 input OR gate
→ 28 transistors

Tối ưu logic nhiều mức

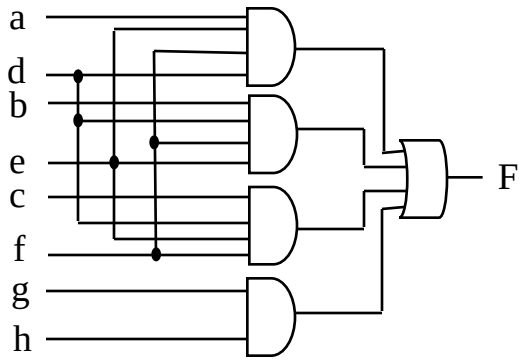
- Cân bằng chất lượng và kích thước
  - Vùng màu xám diễn tả các giải pháp có thể thực hiện
  - Hình tròn với dấu X diễn tả giải pháp tối ưu
    - Thông thường không thể thực hiện được
  - Thiết kế hai mức đạt chất lượng tốt nhất
    - Trễ cực đại = 2 gates
    - Giải bài toán kích thước nhỏ nhất
  - Tối ưu nhiều mức
    - Trễ nhỏ nhất với kích thước cho trước
    - Kích thước nhỏ nhất với trễ cho trước



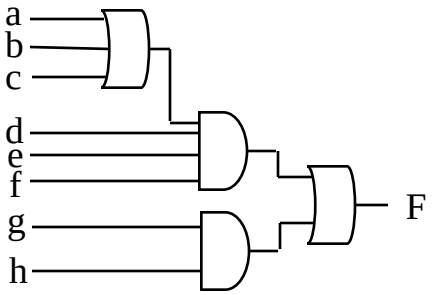
Ví dụ

- Tối thiểu hàm logic hai mức:
  - F = adef + bdef + cdef + gh
  - Yêu cầu 5 cổng với tổng số 18 đầu vào cổng
    - 4 AND và 1 OR
- Sau khi biến đổi đại số:
  - F = (a + b + c)def + gh
  - Chỉ yêu cầu 4 cổng với tổng số 11 đầu vào cổng
    - 2 AND và 2 OR
  - Ít đầu vào/cổng hơn
  - Giả sử mỗi đầu vào cổng = 2 transistors
    - Giảm được 14 transistors
      - 36 (18 * 2) xuống còn 22 (11 * 2)
  - Hy sinh chất lượng để đạt kích thước
    - Đầu vào a, b, và c có mức trễ 3 cổng

2-level minimized



multilevel minimized



Tổng hợp FSM

- Chuyển FSM sang cổng
- Tối thiểu trạng thái
  - Giảm số trạng thái
    - Nhận dạng và ghép các trạng thái tương đương
      - Cùng đầu ra, trạng thái tiếp theo đối với tất cả các đầu vào
      - Dùng bảng
        - Lập bảng tất cả các cặp trạng thái có thể
        - Nếu có n trạng thái, bảng có n^2 tham số
- Mã hóa trạng thái
  - Tuần tự bit duy nhất cho mỗi trạng thái
  - Nếu có n trạng thái, cần $\log_2(n)$ bits
  - n! phương pháp mã hóa

Mô phỏng

- Tạo ra mô hình máy tính cho thiết kế
  - Cung cấp đầu vào mẫu
  - Kiểm tra đầu ra
- Ví dụ kiểm tra tính đúng đắn
  - ALU
    - Cung cấp tất cả các đầu vào có thể (kết hợp)
    - Kiểm tra tính đúng đắn của kết quả đầu ra
- Ví dụ về tính hoàn thiện
  - Cửa thang máy đóng khi di chuyển
    - Cung cấp tất cả các đầu vào tuần tự có thể
    - Kiểm tra cửa luôn đóng khi thang máy di chuyển

Tái sử dụng: Sử dụng các sản phẩm sở hữu trí tuệ (intellectual property – IP)

- Các thành phần có sẵn (COST)
  - Được thiết kế trước, IC được đóng gói trước
  - Thực hiện GPP hoặc SPP
  - Giảm thời gian thiết kế
- Hệ on-chip (SOC)
  - Tất cả các thành phần của hệ được thực hiện trên một chip đơn
  - Làm tăng khả năng của IC

Các thách thức với nhà cung cấp bộ xử lý

- "Lõi đã làm thay đổi đáng kể mô hình kinh doanh
  - Mô hình giá
    - Quá khứ
      - Các nhà sản xuất bán sản phẩm IC cho người thiết kế
      - Nhà thiết kế phải mua thêm các bản copy
    - Hiện tại
      - Các nhà sản xuất bán sản phẩm là các IP
      - Nhà thiết kế có thể tạo nhiều bản copy
    - Các nhà sản xuất có thể dùng mô hình giá khác nhau
      - Mô hình dựa trên bản quyền
        - Tương tự mô hình IC cũ
        - Nhà thiết kế trả tiền cho các model bổ sung
      - Mô hình giá cố định
        - Giá duy nhất cho IP và các bản copy cần thiết khác
      - Nhiều mô hình khác

Các thách thức với người sử dụng bộ xử lý

- Đàm phán về cấp phép
  - Không đơn giản như mua IC
  - Mô hình giá và bảo vệ IP
    - Yêu cầu trợ giúp của pháp luật, bản quyền
- Nỗ lực thiết kế bổ sung
  - Đặc biệt với soft cores
    - Phải được tổng hợp và kiểm tra
    - Sự thay đổi nhỏ trong thiết kế có thể gây vấn đề lớn
- Yêu cầu kiểm tra khó khăn hơn
  - Extensive testing for synthesized soft cores and soft/firm cores mapped to particular technology
    - Ensure correct synthesis
    - Timing and power vary between implementations
  - Early verification critical
    - Cores buried within IC
    - Cannot simply replace bad core

Tóm tắt

- Công nghệ thiết kế nhằm giảm khoảng cách giữa khả năng của IC và tính sản xuất
- Quá trình tổng hợp đã thay đổi thiết kế số
- Tăng khả năng của IC có nghĩa các thành phần sw/hw cùng tồn tại trên một chip
- Quá trình thiết kế dịch chuyển sang thiết kế dựa trên "lõi"
- Mô phỏng là cần thiết nhưng rất khó
- Thiết kế quá trình xoắn ốc là phổ biến