



Chương 1: Embedded Computing



- Dẫn nhập
- Quy trình thiết kế hệ thống nhúng

Định nghĩa hệ thống nhúng (embedded system)

- Hệ thống tính toán bao gồm phần mềm và phần cứng kết hợp chặt chẽ với nhau để thực hiện một chức năng đặc biệt.
- Có thể bao gồm thêm phần cơ khí và các bộ phận khác
- PC có phải là 1 hệ thống nhúng?
- Còn gọi là ubiquitous system hoặc pervasive system

Định nghĩa hệ thống nhúng

- “Nhúng” (embedded): là 1 bộ phận của một hệ thống lớn hơn (embedding system)
- Có thể có nhiều hệ thống nhúng trong 1 hệ thống lớn
- Thực chất không có định nghĩa chính xác

Ví dụ hệ thống nhúng

- Dân dụng
 - Hầu hết các thiết bị điện tử
- Công nghiệp
 - Trong các máy tự động
- Giải trí
 - Video game console
 - Đồ chơi trẻ em
- Quân sự
 - Javelin
 - Tên lửa

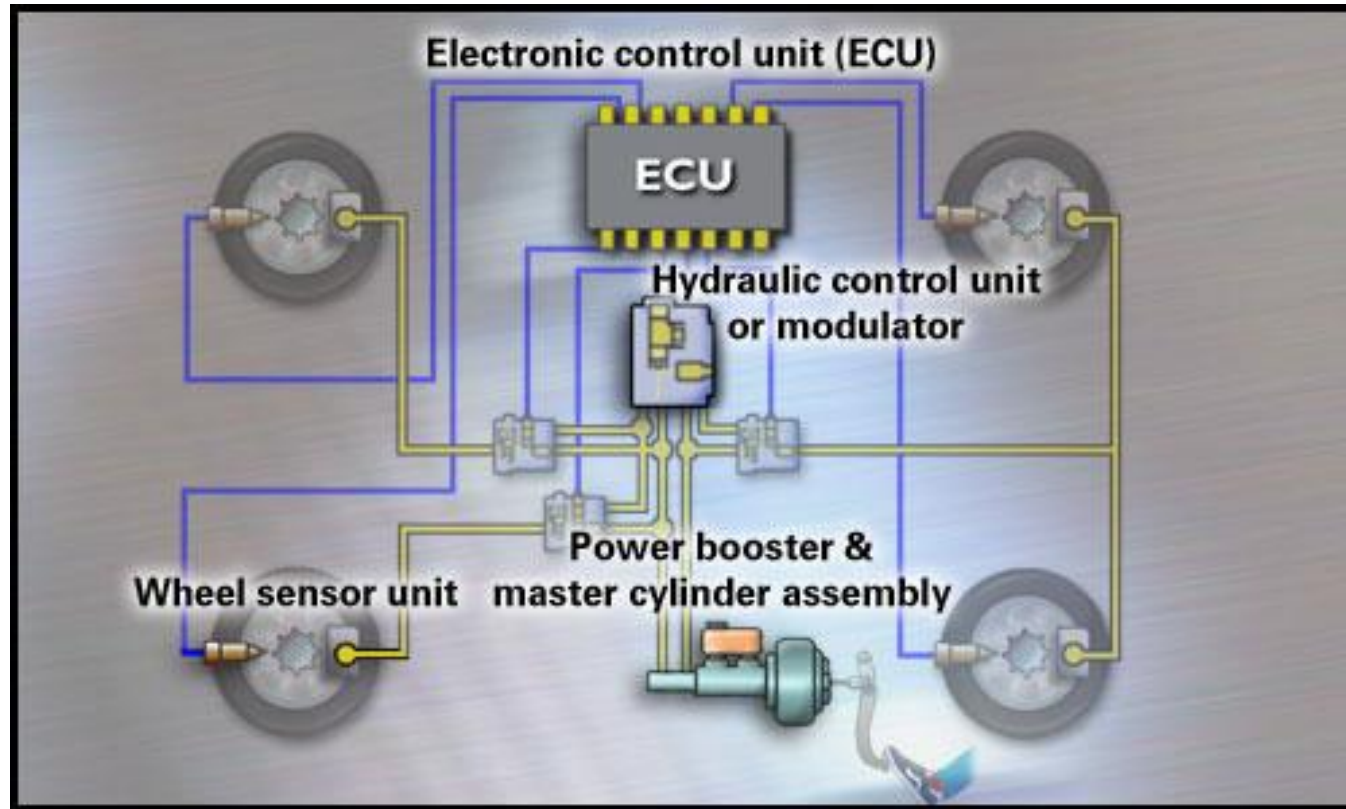
From Computer Desktop Encyclopedia
Reproduced with permission.
© 2007 Nintendo



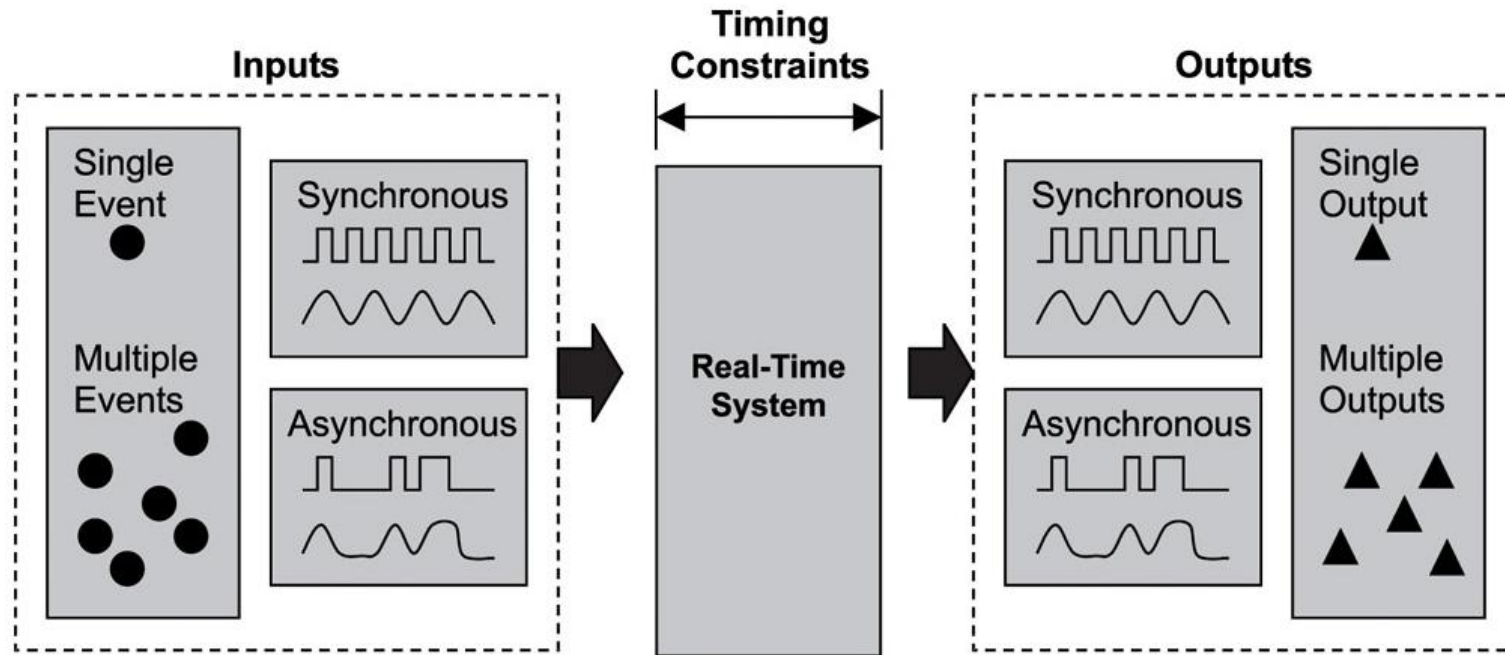
Peter Russo & Brendan Wyich / Stanford University 2009



ABS (anti-lock brake system)

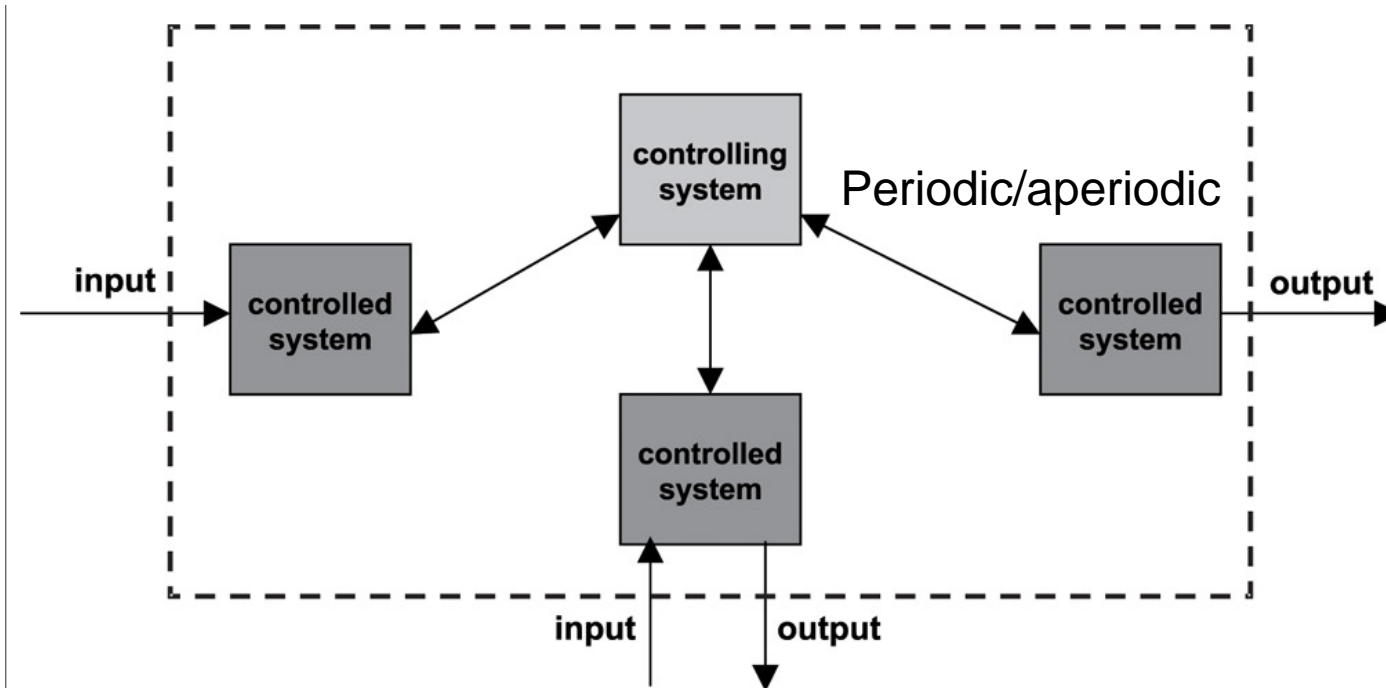


Hệ thống real-time



- Hệ thống có khả năng tính toán hoặc ra quyết định trong (đáp ứng với các sự kiện bên ngoài) 1 khoảng thời gian nhất định biết trước
- Có “deadline”. Trễ deadline cũng tương đương cho ra kết quả sai
- Ví dụ: hệ thống điều khiển bay, ABS
- 2 tính chất quan trọng
 - Functional/logical correctness
 - Timing correctness

Hệ thống real-time



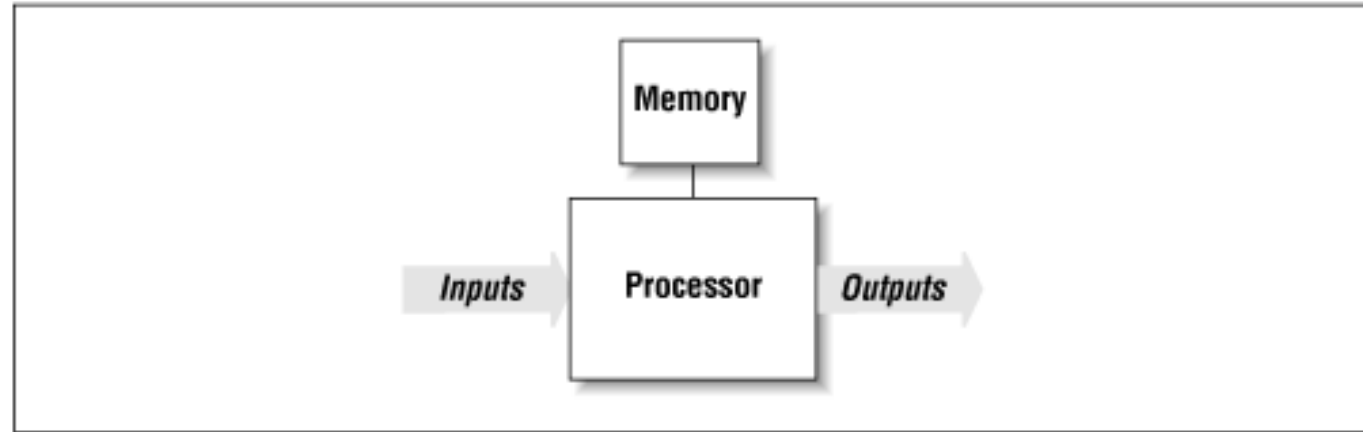
- Hệ thống tên lửa đánh chặn
 - Radar
 - Command & Decision system (C&D)
 - Weapon firing control system
- DVD player
 - Remote
 - DVD player
- Hệ thống đo đạc
 - Sensor
 - Hệ thống tính toán

Hệ thống real-time

- Hard real-time system
 - Phải cho ra kết quả trước deadline
 - Trễ deadline có thể gây ra hậu quả nghiêm trọng
 - Ví dụ: ABS, tên lửa đánh chặn
- Soft real-time system
 - Trễ deadline không gây ra hậu quả nghiêm trọng
 - Ví dụ: DVD player, máy in

Tính chất chung của hệ thống nhúng

- Bộ nhớ
 - RAM
 - ROM/flash
- Vi xử lý
 - Tốc độ thấp
 - SoC
- Input/Output
- Cross-platform development
- Các yếu tố cần quan tâm
 - Sức mạnh xử lý
 - Năng lượng tiêu tốn
 - Bộ nhớ nhiều hay ít
 - Chi phí phát triển
 - Số lượng sản phẩm bán ra
 - Thời gian sống của sản phẩm
 - Tính ổn định của sản phẩm



Mars explorer

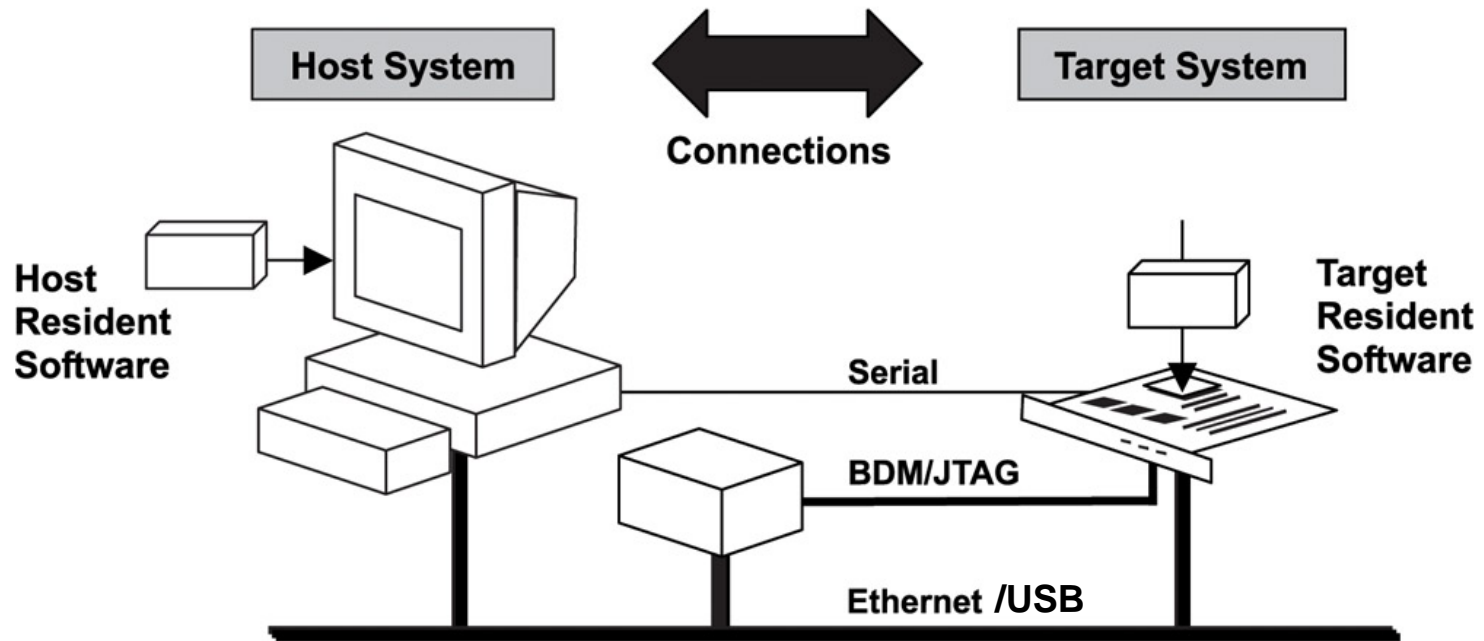
- Viking 1976
 - 2 tàu không người thu thập đất đá trên sao hỏa, phân tích gửi dữ liệu về trái đất
 - 34 triệu dặm, 5 năm
- Pathfinder 1997
 - Tàu đổ bộ: 32-bit CPU, 128 MB RAM
 - Xe tự hành: 8-bit CPU, 512 KB RAM



Thách thức trong thiết kế hệ thống nhúng

- Phần cứng mạnh như thế nào
 - Vi điều khiển, bộ nhớ, ngoại vi
 - Yêu cầu của ứng dụng vs. chi phí
- Đáp ứng yêu cầu ứng dụng như thế nào
 - Nâng cấp phần cứng dẫn tới tăng chi phí
 - Tăng tốc độ CPU có thể không tăng tốc hệ thống
- Giảm thiểu năng lượng tiêu thụ
 - Thời gian sử dụng pin
 - Năng lượng \Leftrightarrow nhiệt
- Khả năng nâng cấp
 - Thêm tính năng bằng cách đổi software
- Độ ổn định
 - Thời gian ra sản phẩm vs. thời gian phát triển
- Nếu viết ứng dụng trên PC giống như lắp ghép máy móc trên ghế thì viết ứng dụng trên hệ thống nhúng giống như thực hiện trên xe hơi

Cơ bản về phát triển hệ thống nhúng



- Ngôn ngữ lập trình thường là C
 - Chạy trên CPU từ 8 bit đến 64 bit
 - Không phụ thuộc CPU → tập trung vào giải thuật
 - Cho phép truy xuất phần cứng bên dưới
- Có thể dùng C++ để tận dụng tính chất hướng đối tượng

Các yếu tố tạo ra thách thức

- Kiểm thử khó khăn
 - Phải đưa vào thiết bị thực tế để tạo ra input đúng
- Khó khăn trong việc quan sát và điều khiển
 - Không có bàn phím và màn hình
 - Phải quan sát tín hiệu bằng oscilloscope
 - Đối với các hệ thống real-time rất khó để debug
- Môi trường phát triển hạn chế
 - Các công cụ phát triển không nhiều
 - Khó khăn khi download chương trình, debug

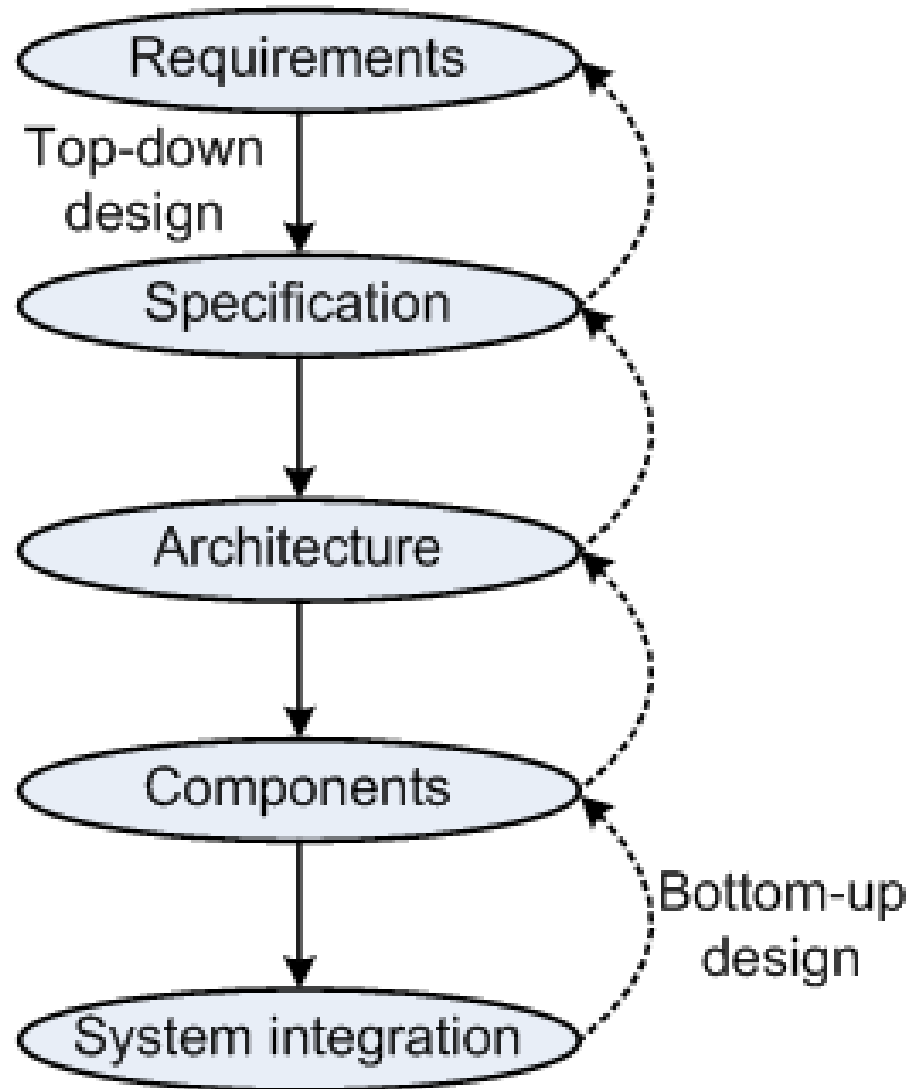
Xu hướng phát triển của hệ thống nhúng

- Chu kì phát triển sản phẩm giảm còn 6-9 tháng
- Quá trình toàn cầu hóa làm mở rộng thị trường và tạo ra nhiều sản phẩm mới
- Các thiết bị điện tử càng ngày càng phức tạp hơn
- Tất cả các sản phẩm đòi hỏi phải có kết nối có dây hoặc không dây
- Các ứng dụng ngày càng dựa trên nền tảng networking nhiều hơn
- Sức mạnh của CPU tuân theo định luật Moore

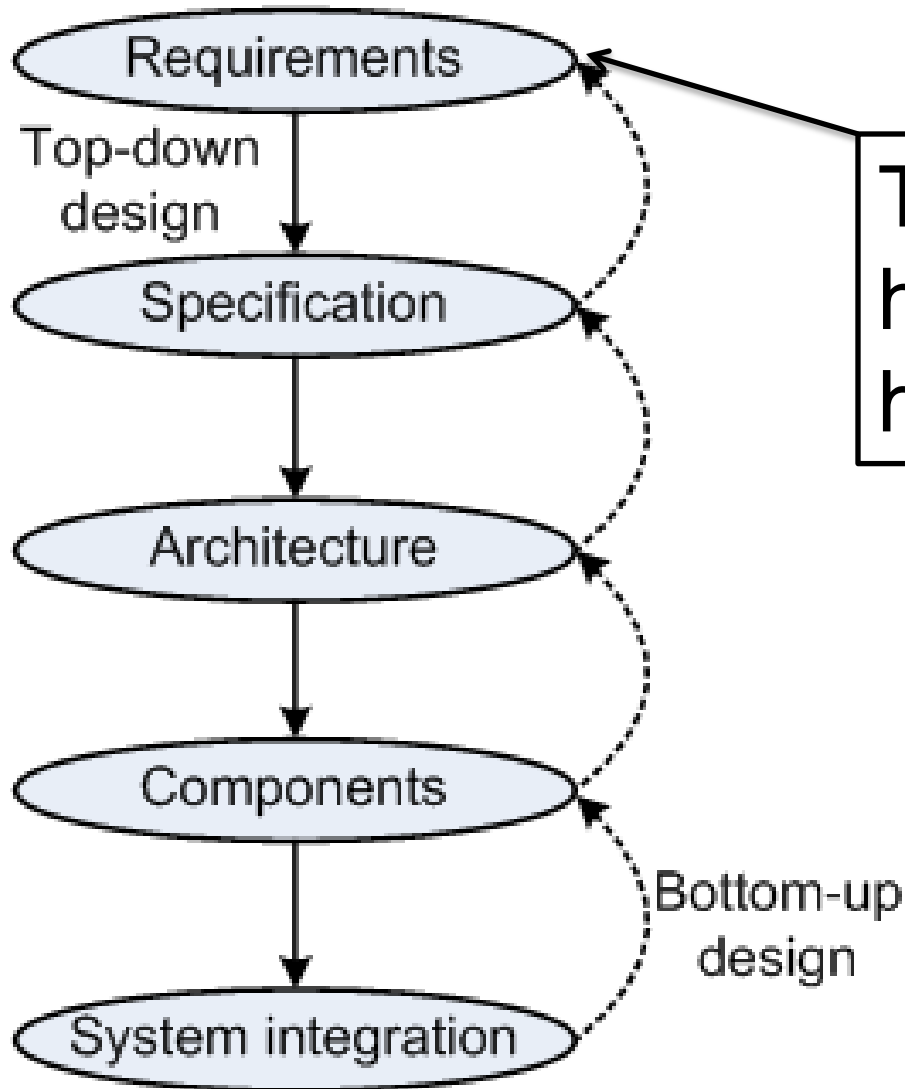
Quy trình thiết kế hệ thống nhúng

- Sự cần thiết của phương pháp luận thiết kế (design methodology)
 - Cho phép xác định được mục tiêu cần thiết
 - Đã hoàn thành hết các chức năng của hệ thống chưa
 - Tối ưu hóa hệ thống, kiểm thử ...
 - Cho phép phát triển các công cụ CAD
 - Chia nhỏ quá trình thiết kế
 - Phát triển các công cụ tự động cho từng bước
 - Hiện tại có nhiều công cụ được hãng thứ 3 phát triển
 - Dễ dàng cho các thành viên trong nhóm giao tiếp với nhau
 - Thành viên biết được mục tiêu của toàn hệ thống
 - Biết được đầu vào, đầu ra của công việc được giao

Quy trình thiết kế hệ thống nhúng

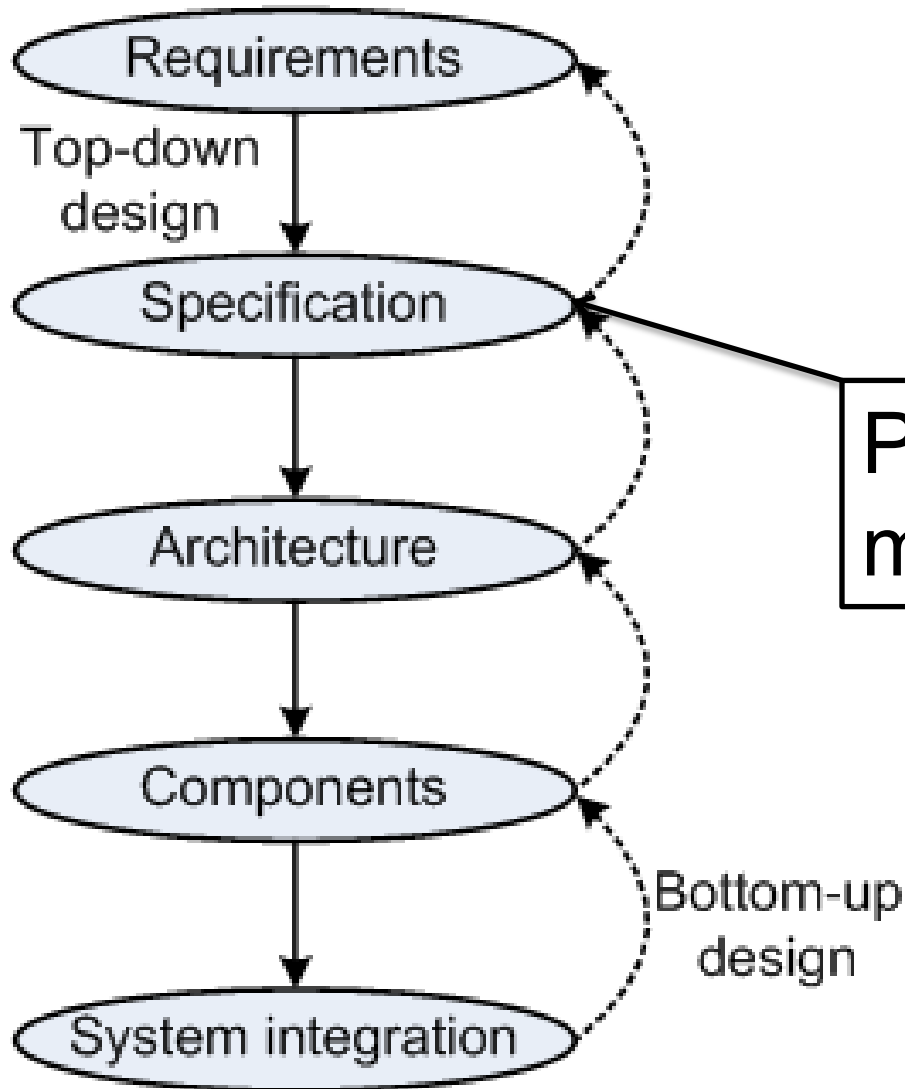


Quy trình thiết kế hệ thống nhúng



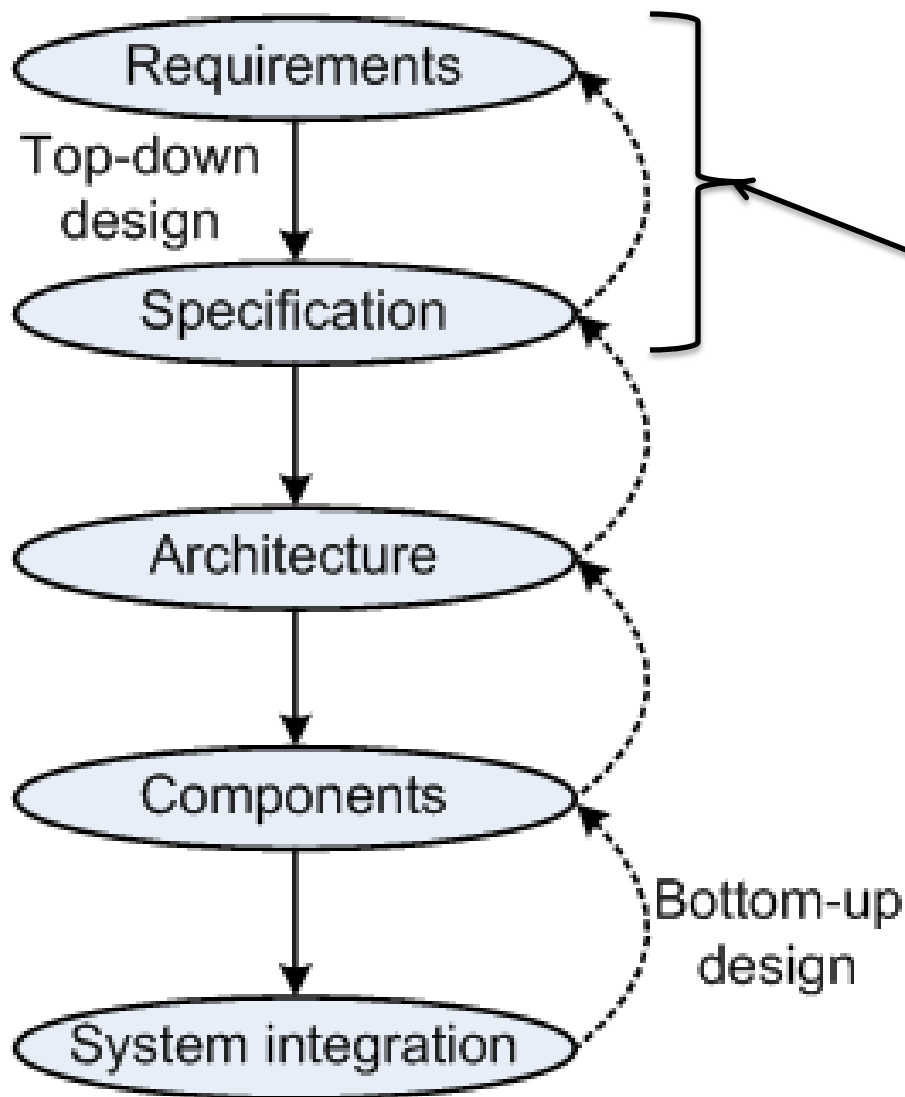
Thu thập thông tin
hệ thống từ khách
hàng

Quy trình thiết kế hệ thống nhúng



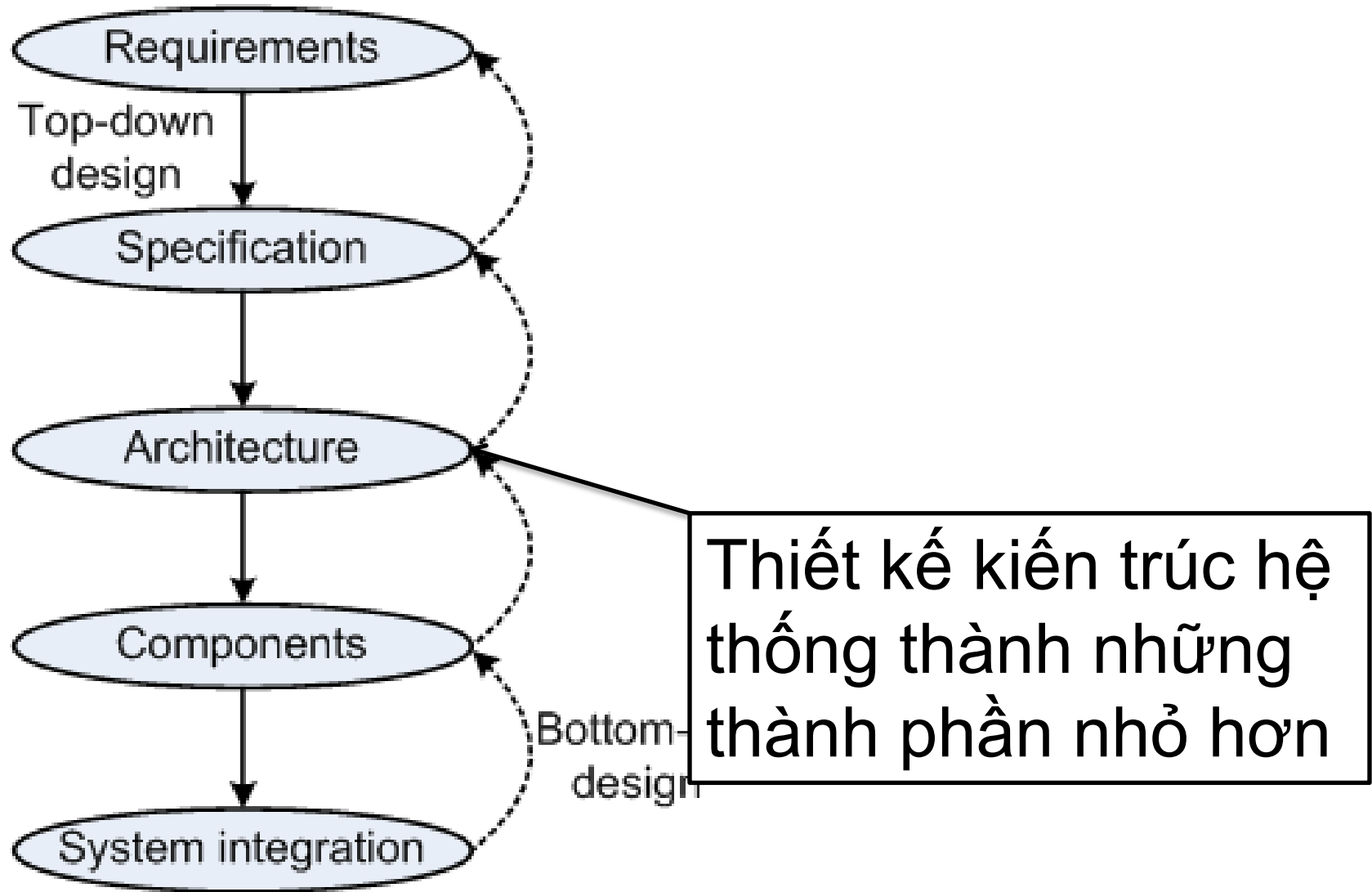
Phân tích thành bản mô tả chi tiết hơn

Quy trình thiết kế hệ thống nhúng

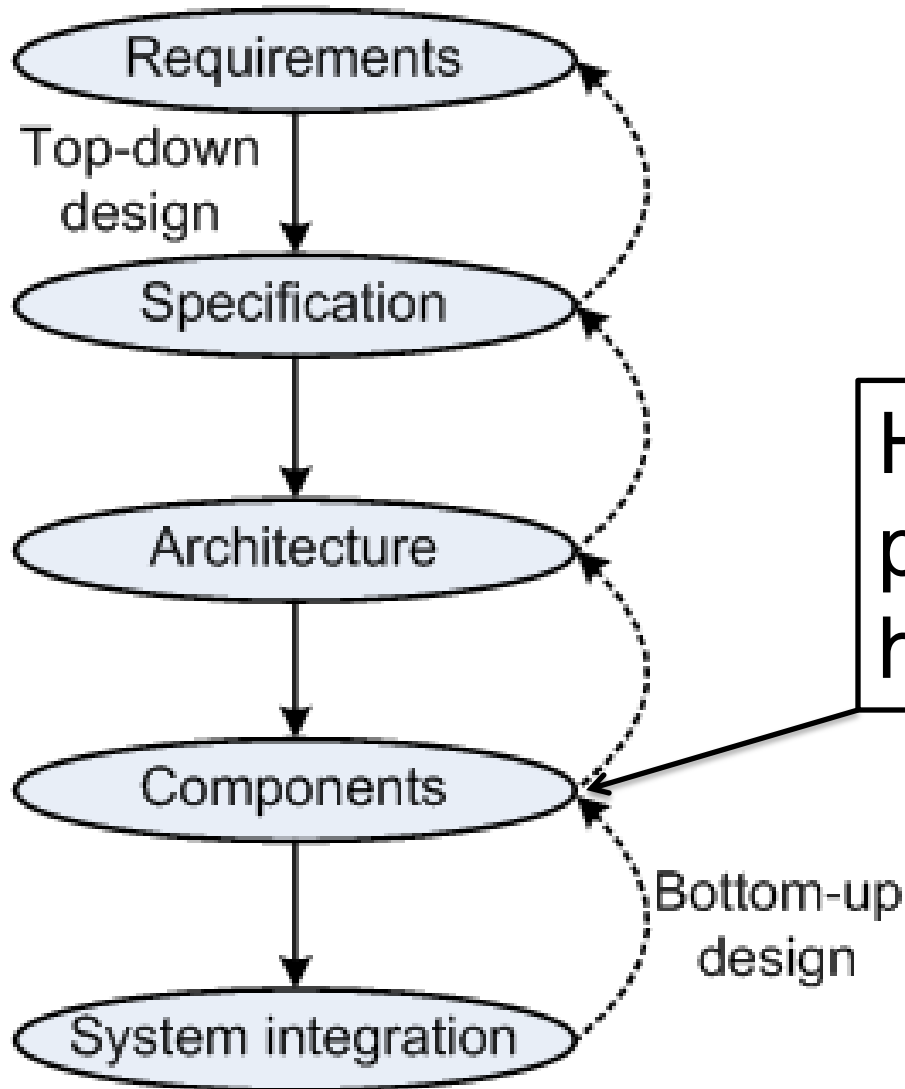


Chỉ phân tích hành vi của hệ thống hoạt động như thế nào, chưa quan tâm hệ thống được hiện thực như thế nào

Quy trình thiết kế hệ thống nhúng

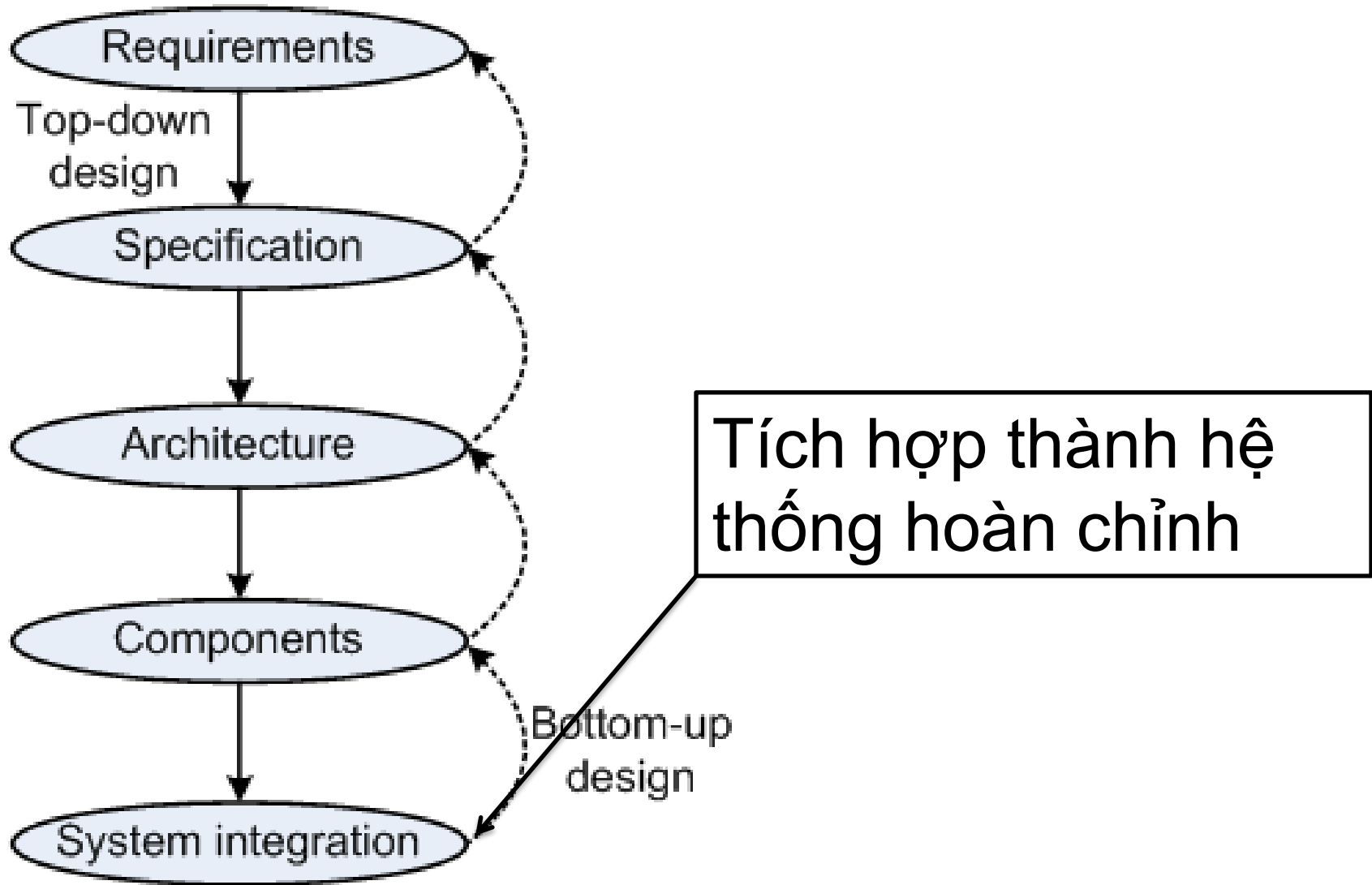


Quy trình thiết kế hệ thống nhúng

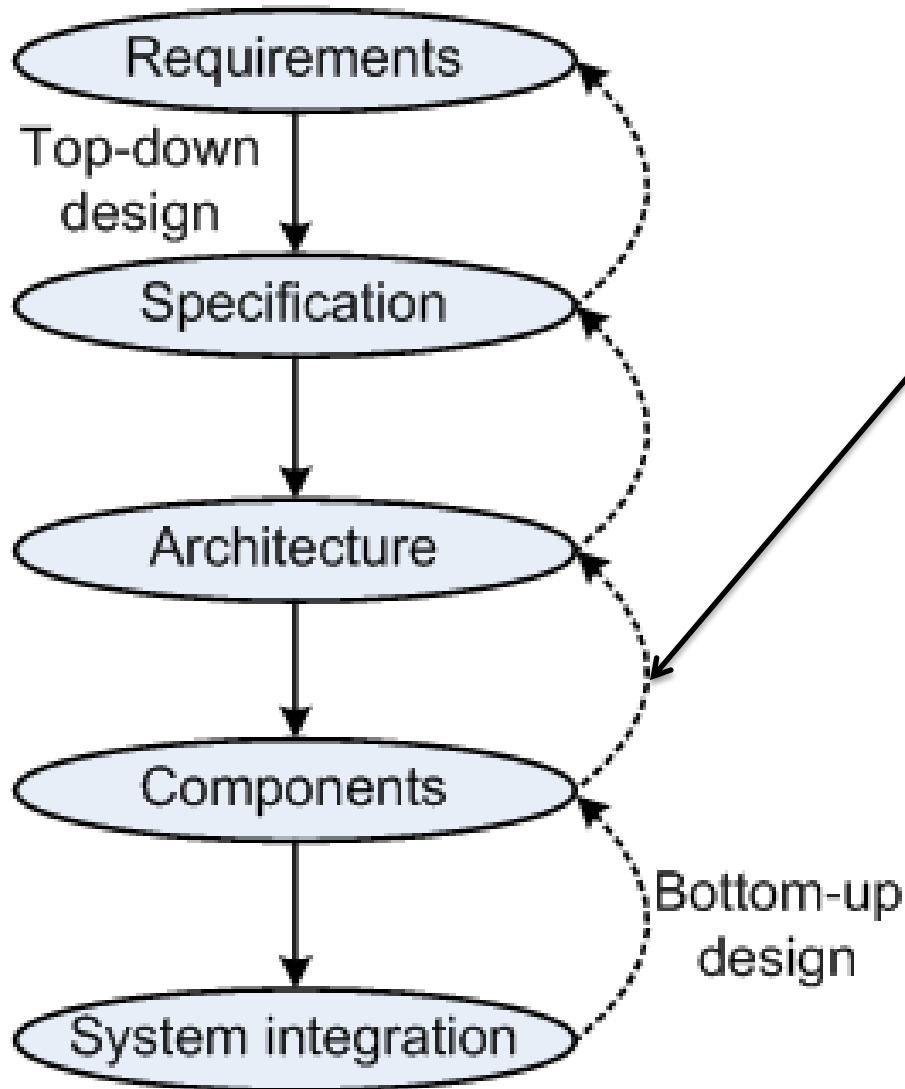


Hiện thực các thành phần hệ thống:
hardware, software

Quy trình thiết kế hệ thống nhúng



Quy trình thiết kế hệ thống nhúng



Feedback để tinh
chế, cải thiện hệ
thống

Requirements

- Thu thập thông tin từ khách hàng
- Khách hàng chỉ mô tả hệ thống ở góc nhìn người sử dụng
 - Mong muốn có thể không thực tế
 - Cách diễn đạt có thể khác với người thiết kế
- Yêu cầu chức năng và phi chức năng
 - Hiệu suất
 - Usability vs. cost
 - Aproximate time vs. hard deadline
 - Chi phí
 - Chi phí sản xuất (manufacturing cost): các bộ phận của hệ thống
 - Chi phí NRE (nonrecurring engineering cost): nhân sự, chi phí thiết kế
 - Kích thước, khối lượng
 - Ảnh hưởng nhiều đến thiết kế
 - Năng lượng tiêu thụ
 - Quan trọng
 - Thường được mô tả theo thời gian sử dụng pin (người sử dụng không quan tâm đến công suất bao nhiêu W)

Form thu thập thông tin đơn giản

- Tên (name)
 - Dễ dàng trao đổi với người khác
 - Làm nổi bật chức năng của hệ thống
- Mục đích (purpose)
 - Mô tả ngắn gọn hệ thống sẽ được dùng làm gì
- Inputs/Outputs
 - Loại dữ liệu (tín hiệu analog, dữ liệu số, cơ học...)
 - Tính chất dữ liệu (dữ liệu định kỳ, người dùng nhập vào, bao nhiêu bit...)
 - Loại thiết bị I/O (nút nhấn, ADC, màn hình...)
- Chức năng (functions)
 - Nên đi từ inputs đến outputs
- Hiệu suất (performance)
- Chi phí sản xuất (manufacturing cost)
 - Chủ yếu là chi phí cho các thành phần cấu tạo nên hệ thống
 - Ảnh hưởng nhiều đến thiết kế
 - Quy luật đơn giản: giá bán ra cao gấp 4-5 lần giá sản xuất
- Năng lượng (power)
- Kích thước, khối lượng

Requirements: ví dụ thiết bị dẫn đường GPS

- Thiết bị được sử dụng cho đường bộ, phải hiển thị các con đường và cột mốc quan trọng có trong cơ sở dữ liệu
- Màn hình tối thiểu 400x600, có ít hơn 3 nút nhấn, sử dụng menu để chọn lựa
- Bản đồ phải cuộn mượt, không bị giật hình. Khi khởi động thiết bị, màn hình phải hiển thị trong vòng 1s, vị trí phải được xác định trong vòng 15s
- Giá bán thiết bị < \$500
- Thiết bị phải cầm vừa trong lòng bàn tay
- Thiết bị phải sử dụng được tối thiểu 8 tiếng với 4 pin AA



Requirements: ví dụ thiết bị dẫn đường GPS

- Tên: thiết bị dẫn đường GPS
- Mục đích: thiết bị dẫn đường dân dụng khi lái xe
- Inputs: 1 nút nguồn, 2 nút điều khiển
- Outputs: màn hình LCD 400x600
- Chức năng: cho phép người dùng lựa chọn 3 độ phân giải, luôn hiển thị kinh độ, vĩ độ
- Hiệu suất: cập nhật màn hình trong vòng 0.25s khi di chuyển
- Chi phí sản xuất: \$100
- Công suất tiêu thụ: 100mW
- Kích thước, khối lượng: nhỏ hơn 2"x6", 350g

Kiểm tra requirements

- Correctness
 - Phải diễn đạt đúng ý khách hàng
 - Tránh dư thừa (overrequiring)
- Unambiguousness
 - Phải rõ ràng, chỉ hiểu được theo 1 nghĩa
- Completeness
 - Tất cả yêu cầu của khách hàng
- Verifiability
 - Phải có cách xác định sản phẩm có thỏa yêu cầu hay không
 - Ví dụ: “attractive”
- Consistency
 - Các yêu cầu không mâu thuẫn lẫn nhau
- Modifiability
 - Có thể thay đổi dễ dàng mà không làm ảnh hưởng đến các điều kiện trên
- Nên xây dựng một mô hình hoặc phiên bản prototype để dễ trao đổi với khách hàng hơn.

Specification

- Phân tích chi tiết của requirement
- Chuyển đổi ý khách hàng thành định dạng người thiết kế có thể sử dụng để xây dựng kiến trúc hệ thống
- Là bước quan trọng trong quá trình thiết kế hệ thống
- Specification phải
 - Dễ hiểu để có thể kiểm tra hệ thống có đạt yêu cầu khách hàng hay không
 - Rõ ràng để có thể xây dựng kiến trúc hệ thống

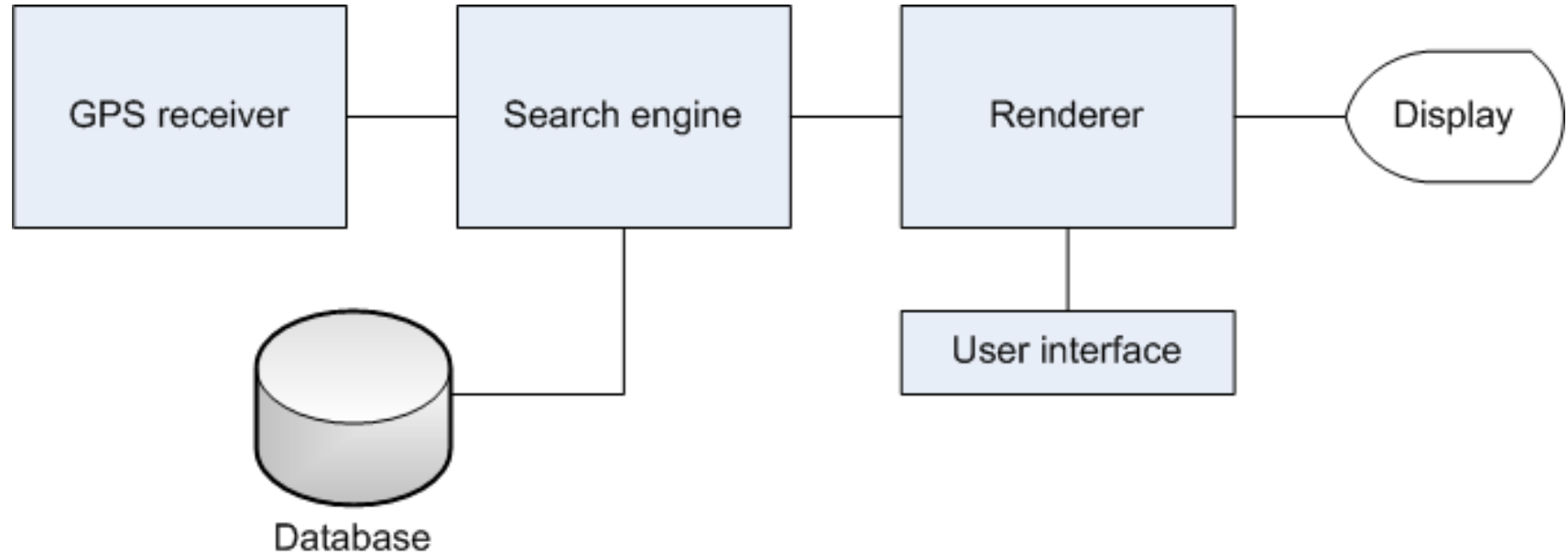
Specification: ví dụ thiết bị dẫn đường GPS

- Dữ liệu thu từ bộ thu GPS
- Dữ liệu bản đồ
- Giao diện người dùng
- Các chức năng phải thực hiện để đáp ứng yêu cầu người dùng
- Các chức năng bên trong để duy trì hoạt động của hệ thống.

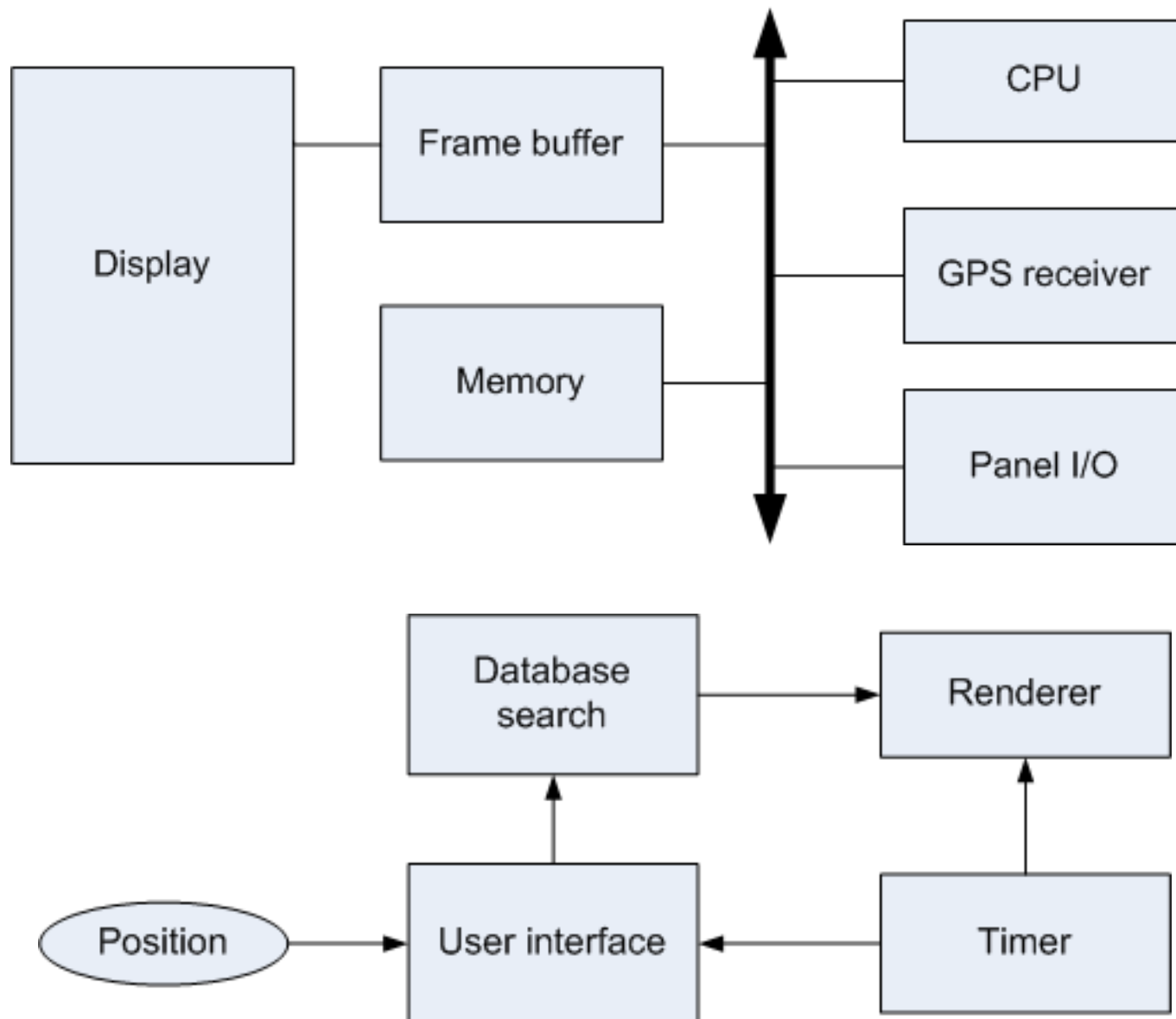
Thiết kế kiến trúc hệ thống

- Là bước đầu tiên của quá trình thiết kế
- Phải thỏa mãn yêu cầu chức năng lẫn không chức năng
 - Bắt đầu từ kiến trúc tổng quát sau đó tinh chế dần trong các bước thiết kế hardware, software
 - Tập trung yêu cầu chức năng trước sau đó mới đến yêu cầu không chức năng
 - Phải biết ước lượng hệ thống → kinh nghiệm

Kiến trúc: ví dụ thiết bị dẫn đường GPS



Kiến trúc: ví dụ thiết bị dẫn đường GPS



Thiết kế thành phần hệ thống

- Thành phần có sẵn
 - GPS receiver, CPU
 - Dữ liệu bản đồ
- Tự thiết kế
 - PCB kết nối các thành phần
 - Lập trình
- Lưu ý các yêu cầu không chức năng

Tích hợp hệ thống

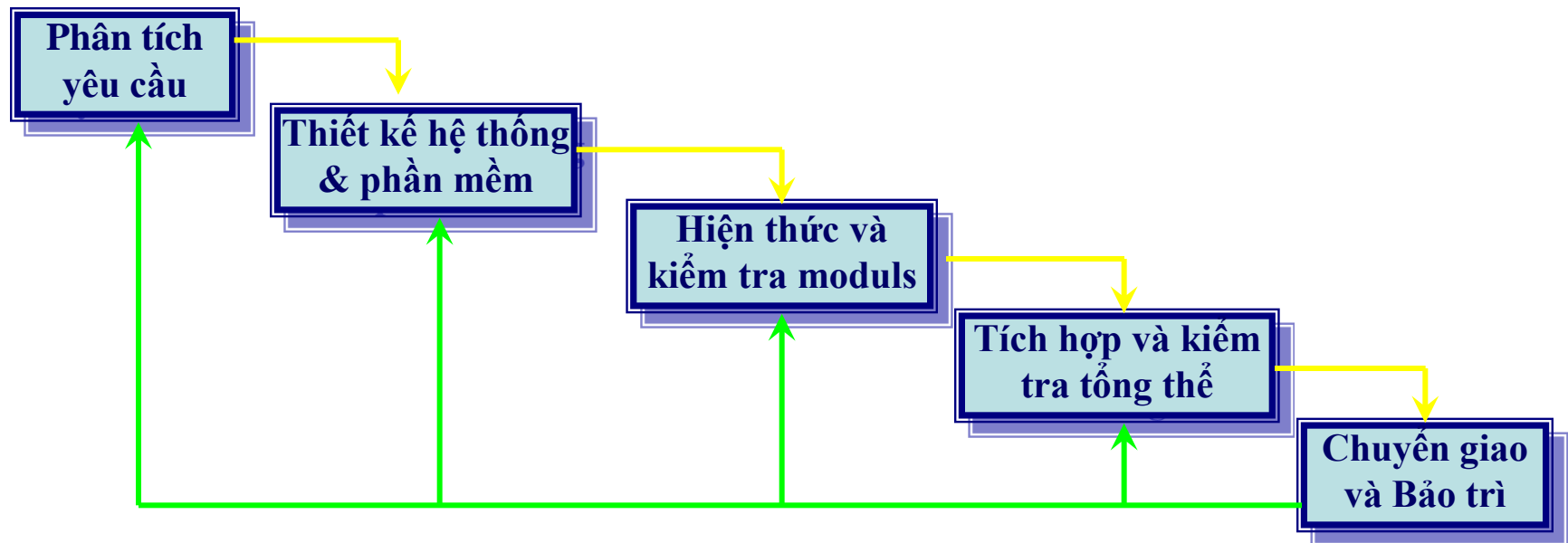
- Kết nối các thành phần hệ thống
- Debug
- Khó khăn

Phương pháp luận thiết kế

- Nhiều người có thể làm việc với nhau
- Cần thiết để xây dựng một hệ thống hoạt động đúng
 - Thỏa các yêu cầu chức năng, phi chức năng
 - Time to market
 - Chi phí cho việc thiết kế
 - Chất lượng sản phẩm
 - Bug có thể gây nguy hiểm tính mạng con người
 - Mất tàu do thám sao Hỏa năm 1999
 - Toyota

Waterfall model

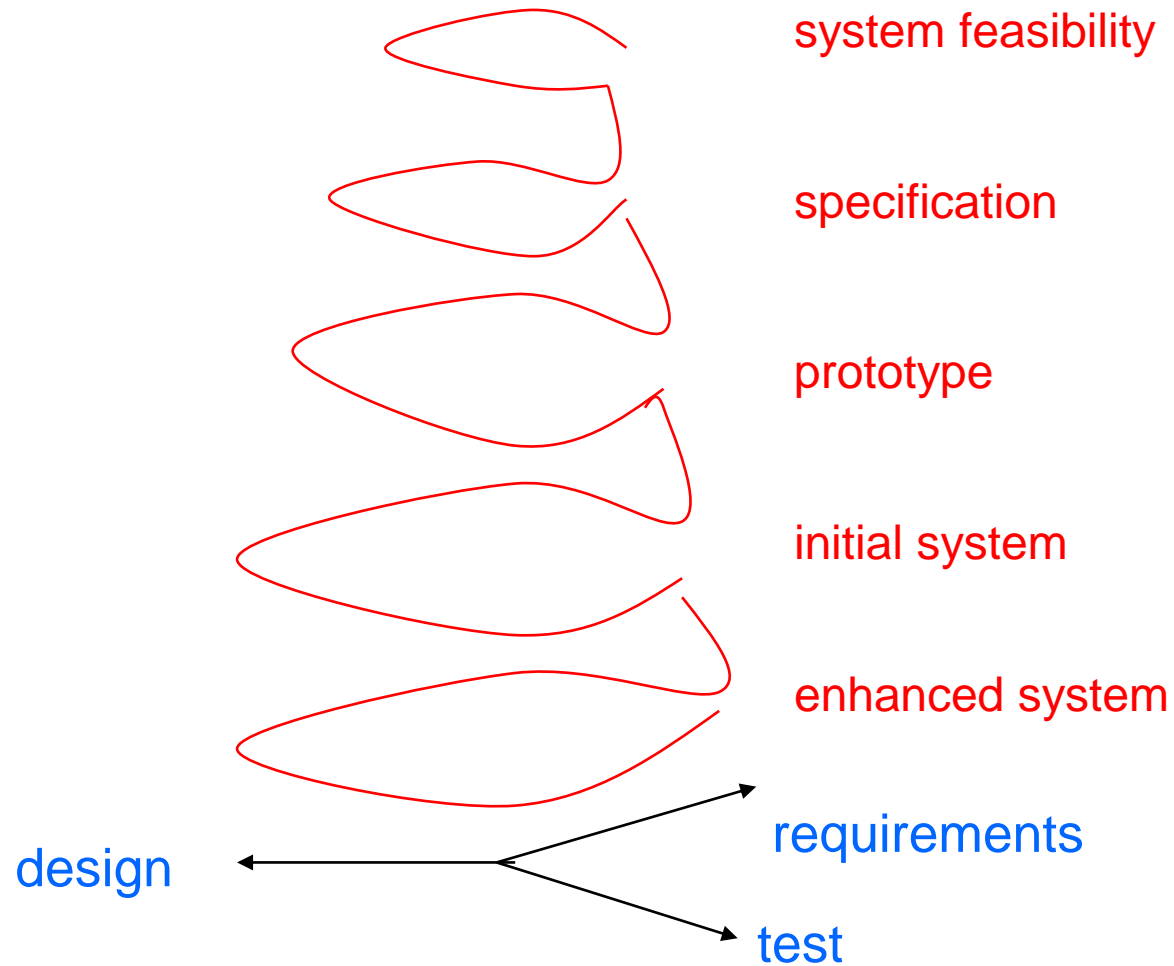
- Mô hình phát triển phần mềm đầu tiên
- Các công việc tiếp nối nhau một cách tuần tự
- Đặt nền móng cho các phương pháp phân tích, thiết kế, kiểm tra...



Waterfall model

- Bản chất của phát triển phần mềm là quá trình lặp đi lặp lại chứ không phải tuần tự
- Các bước thực chất không tách biệt hoàn toàn mà có chồng lấn và tham khảo lại
- Bắt buộc khách hàng đặc tả tất cả yêu cầu một cách chính xác và đầy đủ ngay từ ban đầu
- Khách hàng thường phải chờ đợi rất lâu để thấy được phiên bản đầu tiên của sản phẩm
- Tồn tại “delay” tích lũy trong nhóm làm việc -> dự án thường bị trễ.
- Chỉ phù hợp cho dự án nhỏ, đơn giản.

Spiral model



Hierarchical design flow

