

TDDI11: Embedded Software

Embedded Systems Design

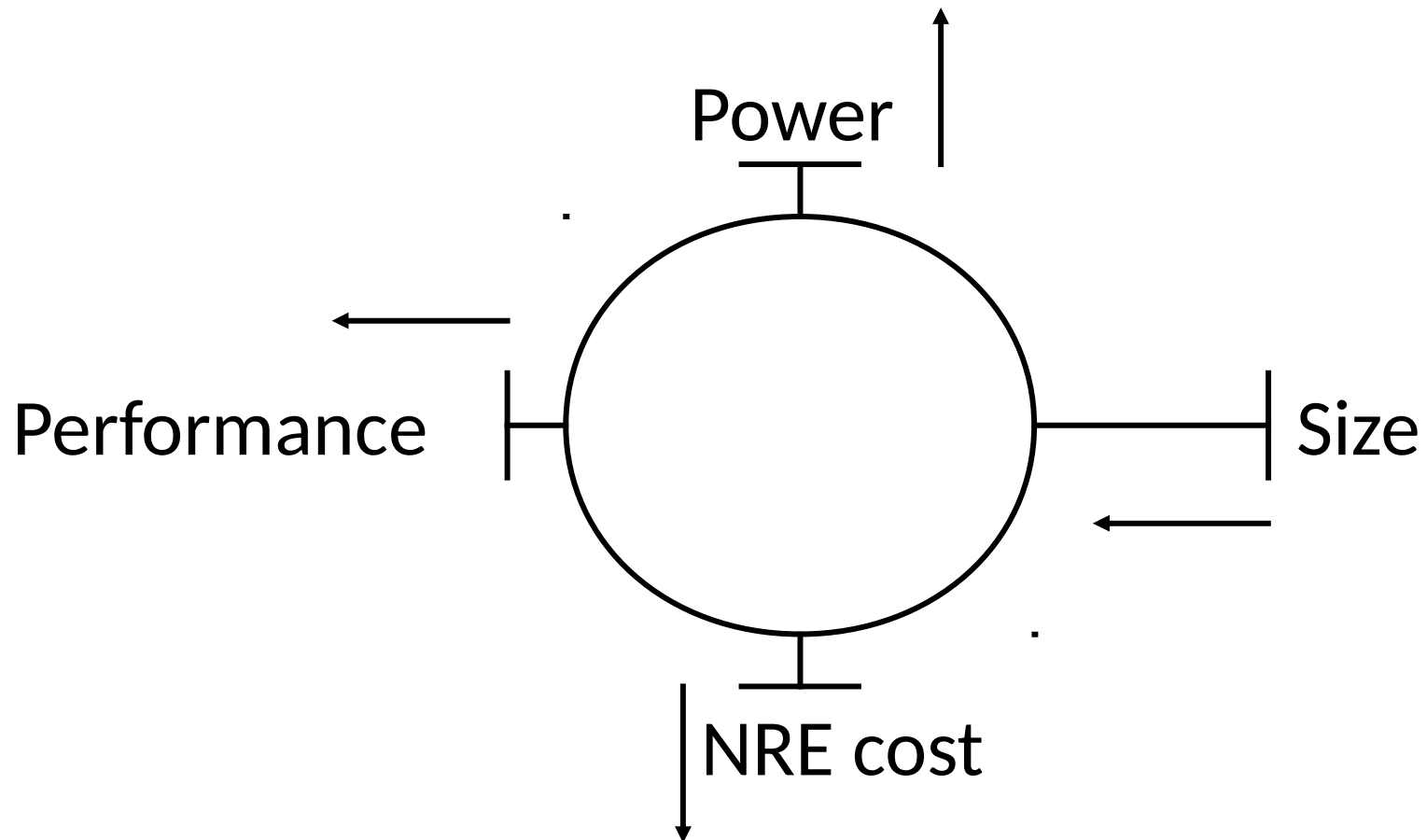
Design challenge – optimizing design metrics

- Common metrics
 - **Unit cost:** the monetary cost of manufacturing each copy of the system, excluding NRE cost
 - **NRE cost** (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
 - **Size:** the physical space required by the system
 - **Performance:** the execution time or throughput of the system
 - **Power:** the amount of power consumed by the system
 - **Flexibility:** the ability to change the functionality of the system without incurring heavy NRE cost

Design challenge – optimizing design metrics

- Common metrics (continued)
 - **Time-to-prototype:** the time needed to build a working version of the system
 - **Time-to-market:** the time required to develop a system to the point that it can be released and sold to customers
 - **Maintainability:** the ability to modify the system after its initial release
 - **Safety:** absence of catastrophic consequences on the user(s) and the environment.

Design metric competition: improving one may worsen others



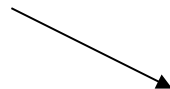
Design methodology, design flow

- **Design methodology:** the process of creating a system
 - Goal: optimize competing design metrics
 - Time-to-market
 - Design cost
 - Manufacturing cost
 - Quality, etc.
- **Design flow:** sequence of steps in a design methodology.
 - May be partially or fully automated.
 - Use tools to transform, verify design.
- Design flow is one component of design methodology.
Methodology also includes management, organization, etc.

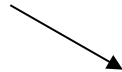
Waterfall model

- Early model for software development:

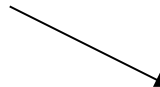
**requirements/
specification**



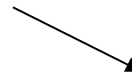
architecture



coding



testing

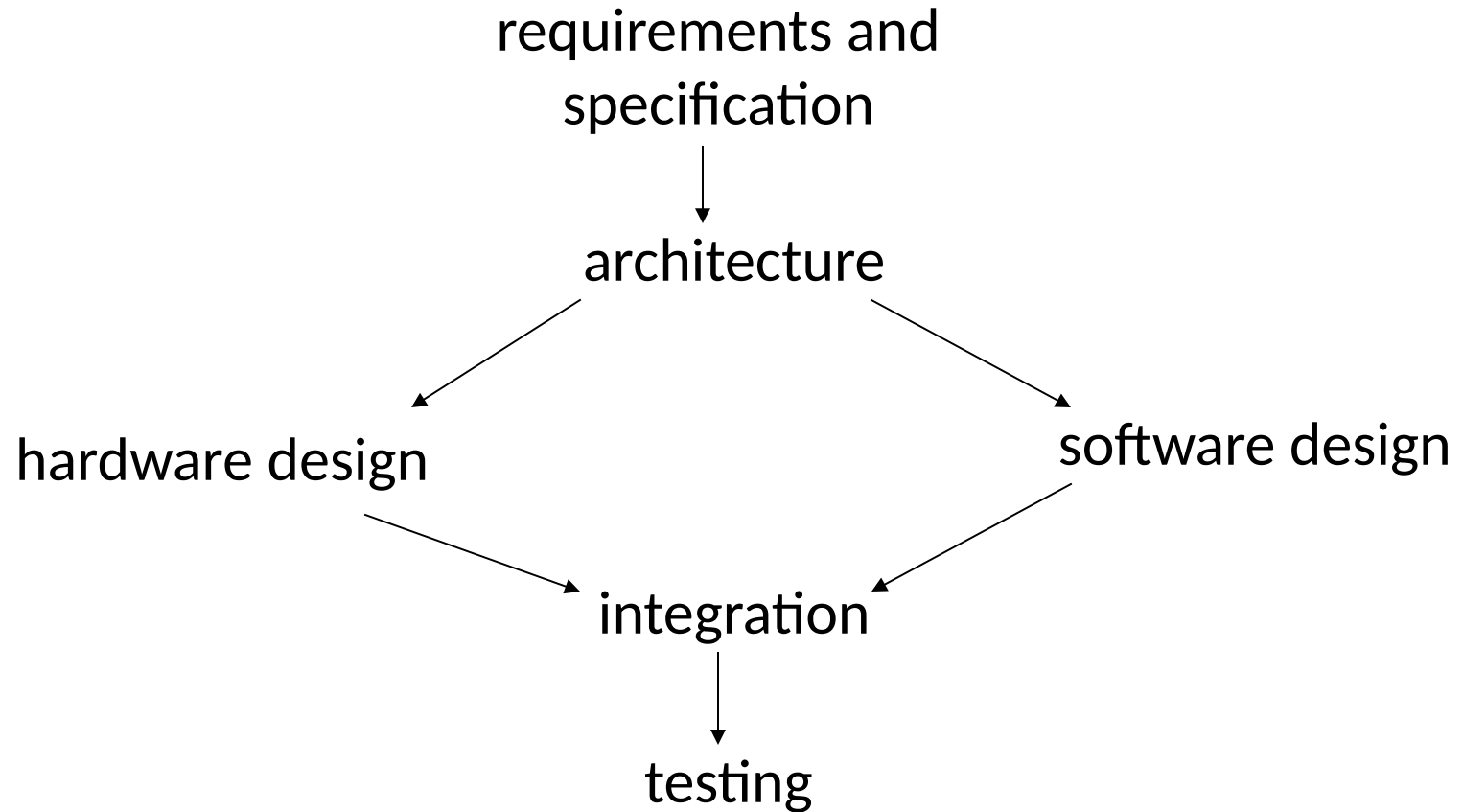


maintenance

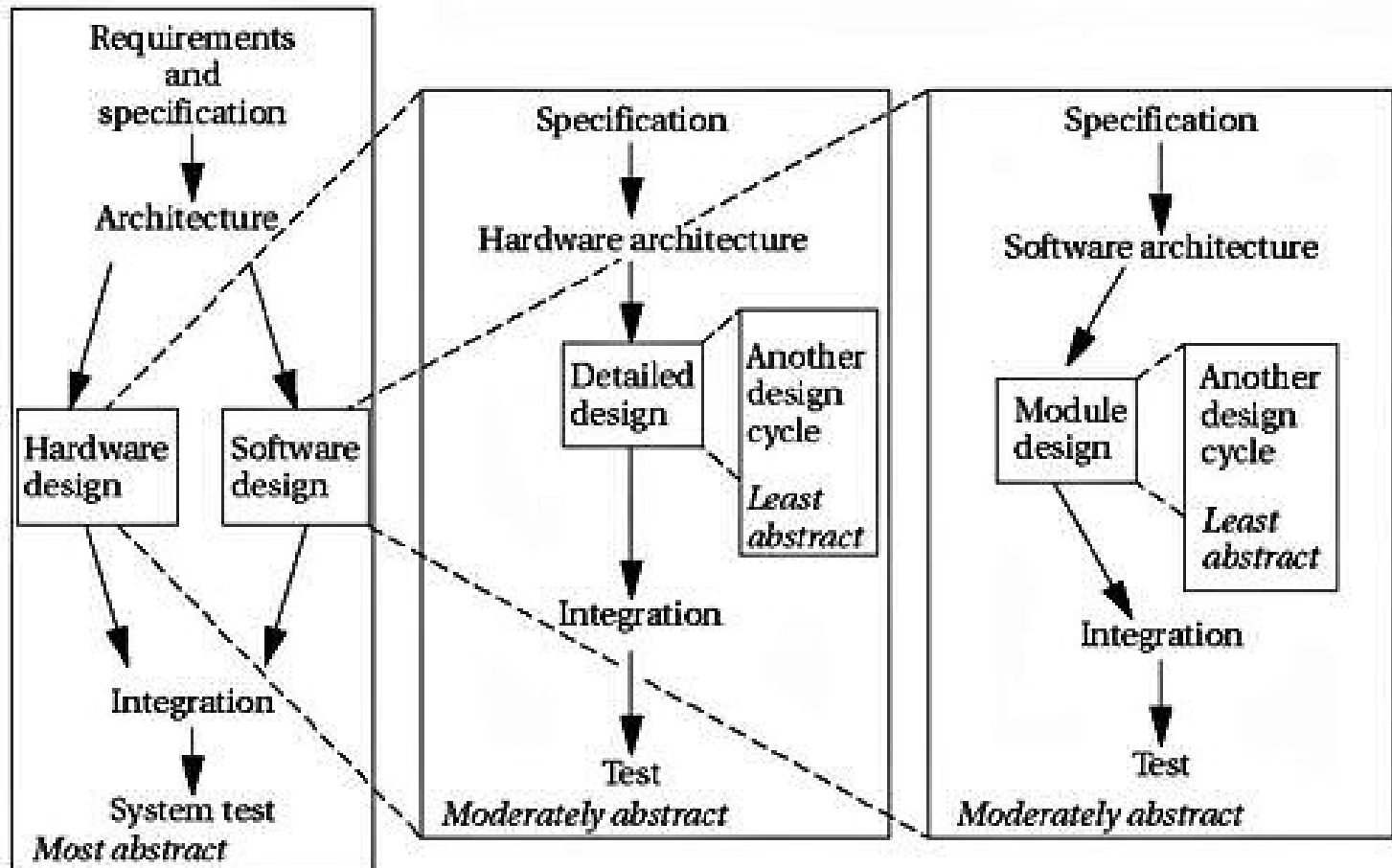
Design flows for embedded systems

- Embedded systems need design of hardware and software
- Even if you don't design hardware, you still need select the correct boards, plug together several hardware components, and write code

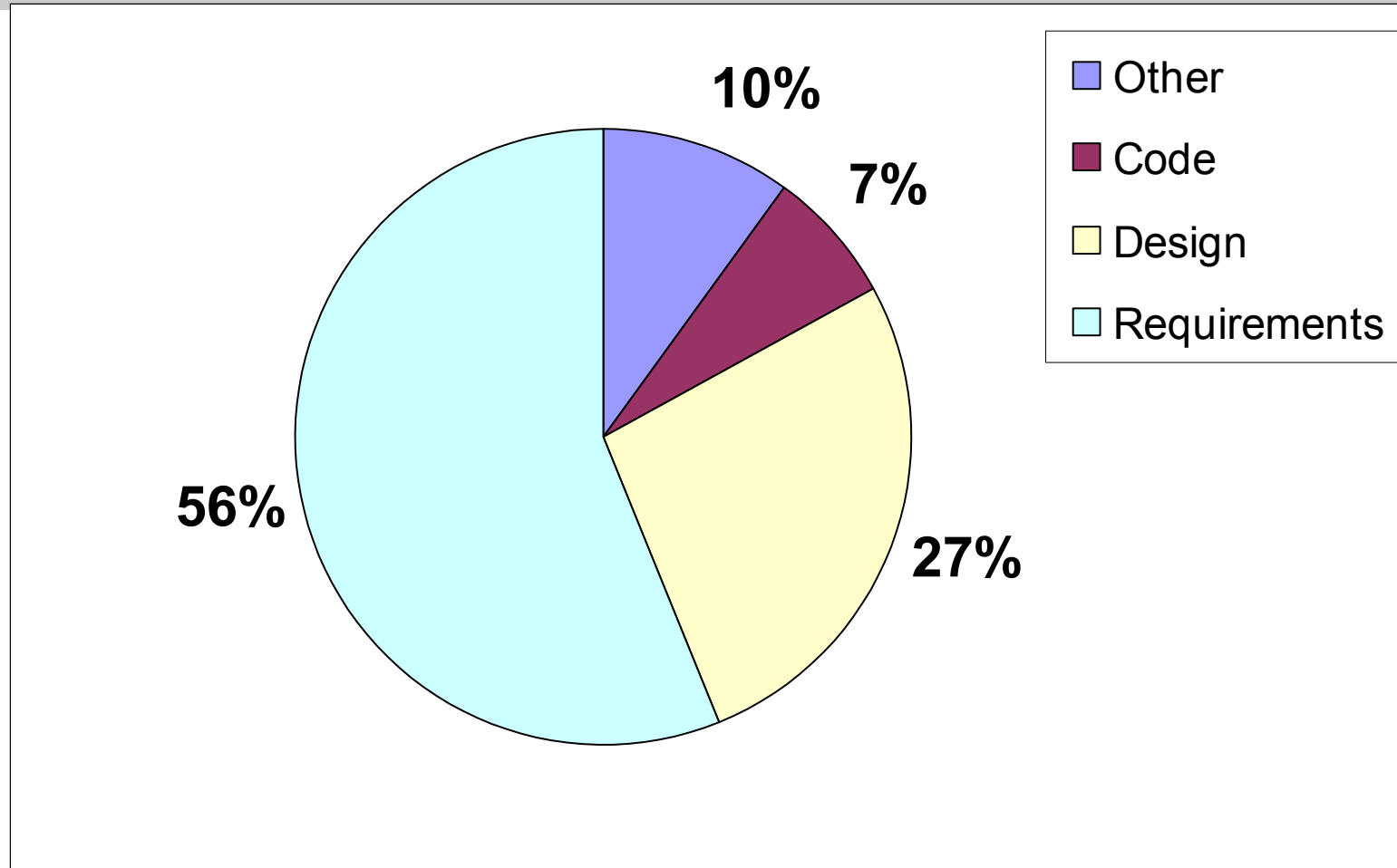
Hardware/software design flow



Hierarchical design



Frequency of faults



[Jim Cooling 2003, cited from DeMarco78]

Specification

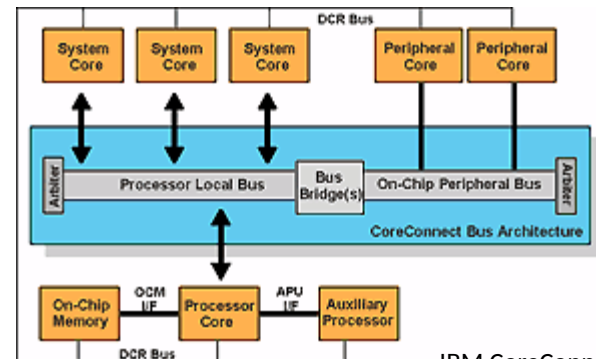
- Describing embedded system's behavior
 - Can be extremely difficult
 - Complexity increasing with increasing IC capacity
 - Desired behavior often not fully understood in beginning
- English (or other natural language) common starting point
 - Precise description difficult to impossible
- Überlingen crash example

Models and languages

- How can we (precisely) capture behavior?
 - We may think of languages (C, C++), but *computation model* is the key
- Common computation models:
 - Sequential program model
 - Statements, rules for composing statements, semantics for executing them
 - State machine model
 - For control dominated systems, monitors control inputs, sets control outputs
 - Dataflow model
 - For data dominated systems, transforms input data streams into output streams

Reuse: platforms

- A partial design:
 - for a particular type of system;
 - includes embedded processor(s);
 - may include embedded software;
 - customizable to a customer's requirements:
 - software;
 - component changes.



IBM CoreConnect

Why platforms?

- Any given space has a limited number of good solutions to its basic problems.
- A platform captures the good solutions to the important design challenges in that space.
- A platform reuses architectures.

Standards and platforms

- Many high-volume markets are standards-driven:
 - wireless;
 - multimedia;
 - networking.
- Standard defines the basic I/O requirements.

MPEG Tampere
meeting



bluetooth.com

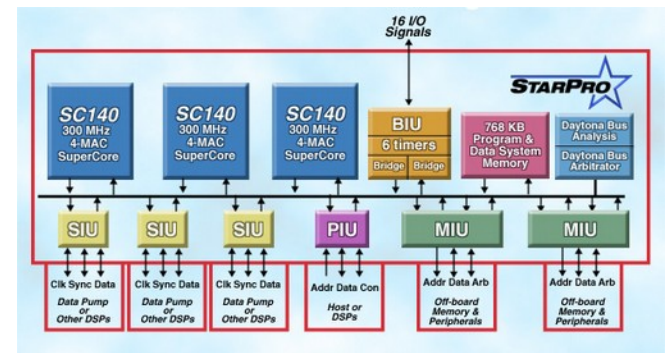
Standards and platforms, cont'd.

- Systems house chooses implementation of standards functions:
 - improved quality, lower power, etc.
- Product may be differentiated by added features:
 - cell phone user interface.
- Standards encourage platform-based design.

Platforms and embedded computing

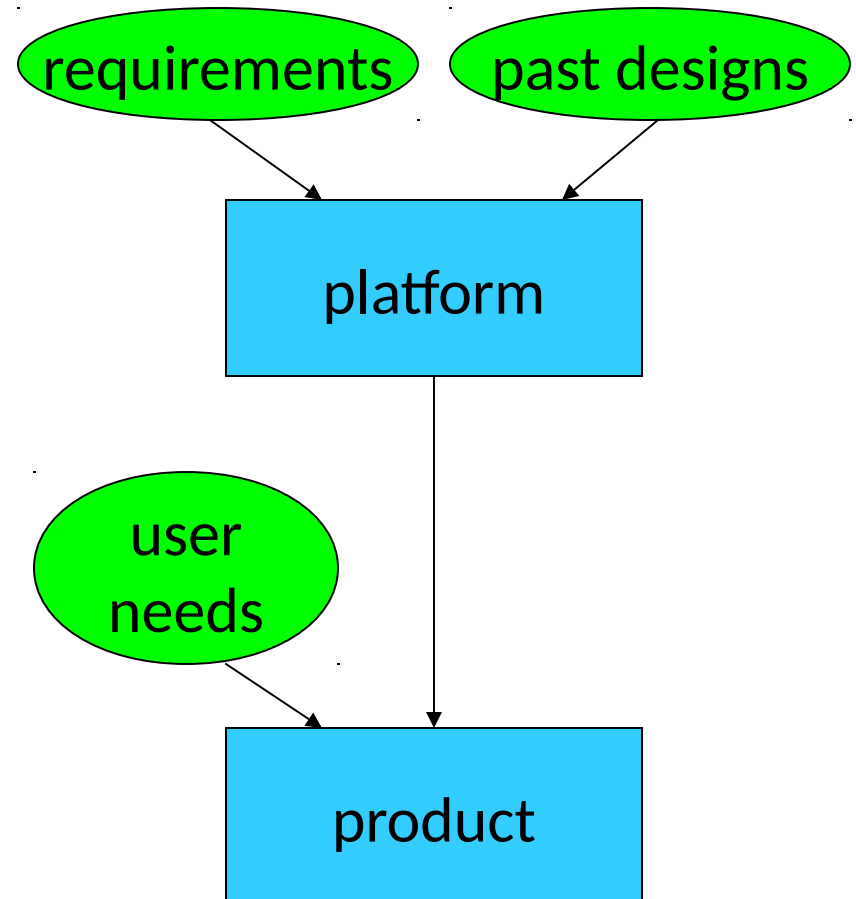
- Platforms rely on embedded processors:
 - can be customized through software;
 - can put considerable design effort into the CPU.
- Many platforms are complex heterogeneous multiprocessors.

Agere StarPro



Two phases of platform-based design

- Design the platform.
- Use the platform.



Design example