Nested Processors, Virtual Memory, Hardware & Virtual Platforms

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Jan 2017

Approach

Hardware platform and virtual hardware platform

Processor, memory, bus without subcomponents

Open discussion question:

Need for distinction between platform and virtual platform?

Summary/Actions

- No nested processors, memory, buses: composition via system
- No virtual x as subcomponent of x: handled by bindings
- Virtual platform with connectivity between virtual bus, virtual processor, virtual memory
- Composition through system component: no new platform category
- Virtual memory and binding points: work examples to validate need for both

Processor and Virtual Processor

Separate Hardware and Software concerns

Processor: specification of hardware aspects (chip, core, multi-threads)

Virtual Processor: (OS, partitions, schedulers)

Processor binding points: for cycles (and storage as processors have memory subcomponents)

V3: Virtual Processor no longer contained in a processor

Virtual Processor

Logical resource with capacity/budget

Containment region

“Nested” Processor and hardware platform

V3: processor as system (platform) subcomponent

Properties to describe hardware constraints (endianness)

Processors with “Internal” Memory

Explicit Memory Model

System with processor and memory components

Implicit Memory Model

Expressed via processor binding point resource type

Bus & Virtual Bus

- Separation of physical and logical concerns
  - Binding of logical to physical
- Bus
  - Physical entity for transferring data (discrete logical entities)
  - ~~Bus for continuous flow between physical features~~
- Virtual bus
  - Logical channel resource with resource capacity/budget
  - Containment region for security/safety
  - Protocol with wrapper overhead
- Bus as system (platform) subcomponents
  - Network architecture with AADL bus to bus connectivity

- Virtual bus implementation
  - Realization of virtual bus protocol abstraction
    - Protocol wrapping/unwrapping
    - En/Decryption

Memory & Virtual Memory

- Separation of physical and logical concerns
  - Binding of logical to physical
- Memory
  - Storage with binding points
  - Need for representing different section of memory addresses: binding points have properties to indicate base address and range (size)
  - Memory binding point on devices can model device registers without requiring memory subcomponents.
- Virtual memory roles
  - Represent logical addresses that are mapped to addresses in different components in the platform
  - Logical resource with capacity/budget
  - Logical containment regions
  - Represent segments of address space
- Memory as system (platform) subcomponents
  - Subcomponents as binding points
  - Memory system architecture with connectivity via bus
  - Platform with memory and processor

Can we separate the issue of address mapping from containment region?

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

Platform & Virtual platform

- System as platform abstraction
  - Provided resource expressed through binding points
  - Physical HW system model
  - Memory, bus, processor as leaves
  - Parameterized platform configuration
- Virtual platform
  - Connectivity between virtual buses, virtual processors, virtual memory
  - Virtual platform and elements as resource providers
    - via binding to physical resources
- Physical platform refinement
  - Simple platform model: processor, memory, network
  - Refinement: Processor is actually a system of internally networked processor (e.g., multi-core chip)
  - Refinement: an avionics network is actually a switch with internal bus, memory, and processor units
  - Use processor, bus, memory implementation to flesh out realization and configure into the model