Provably Correct Memory Management

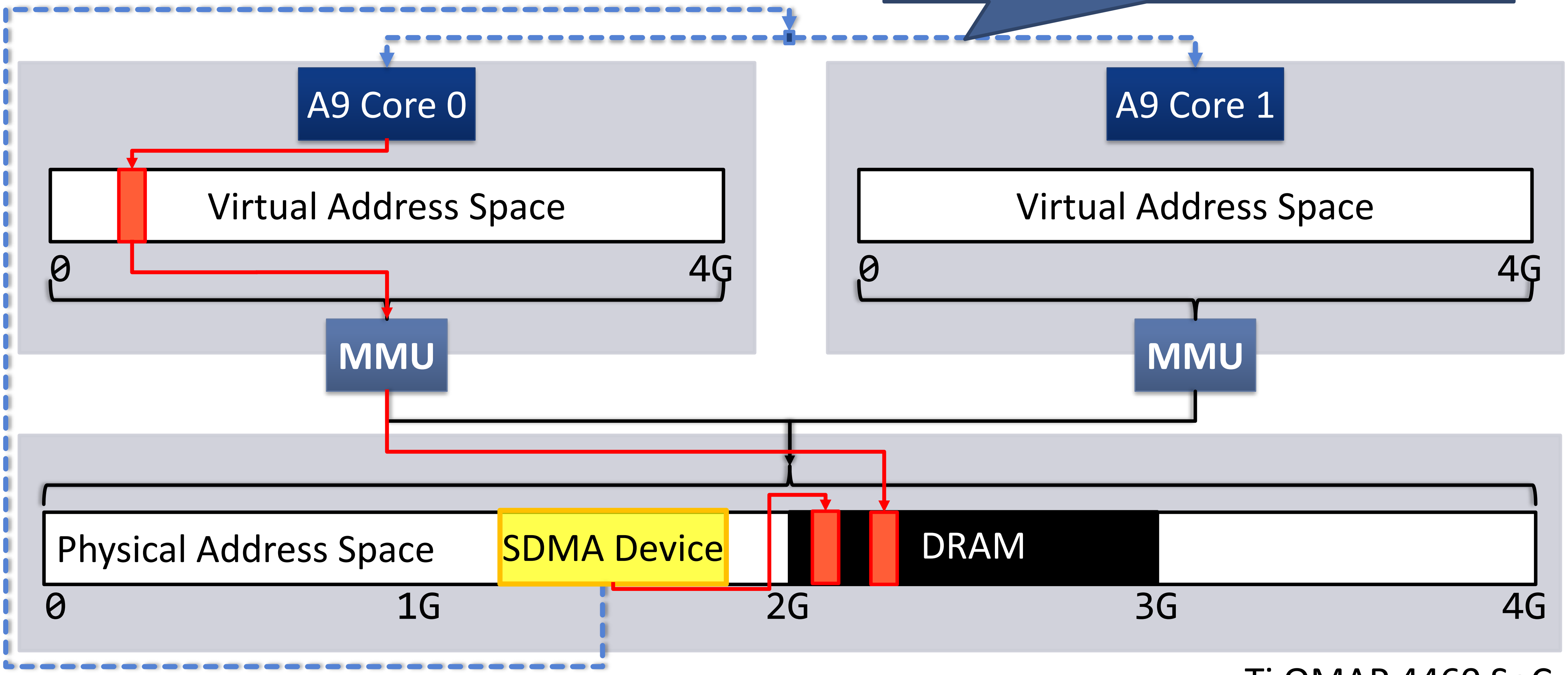
Reto Achermann

11th EuroSys Doctoral Workshop (EuroDW’17), Belgrade

Systems Group, Department of Computer Science, ETH Zurich

The naïve view of today's systems

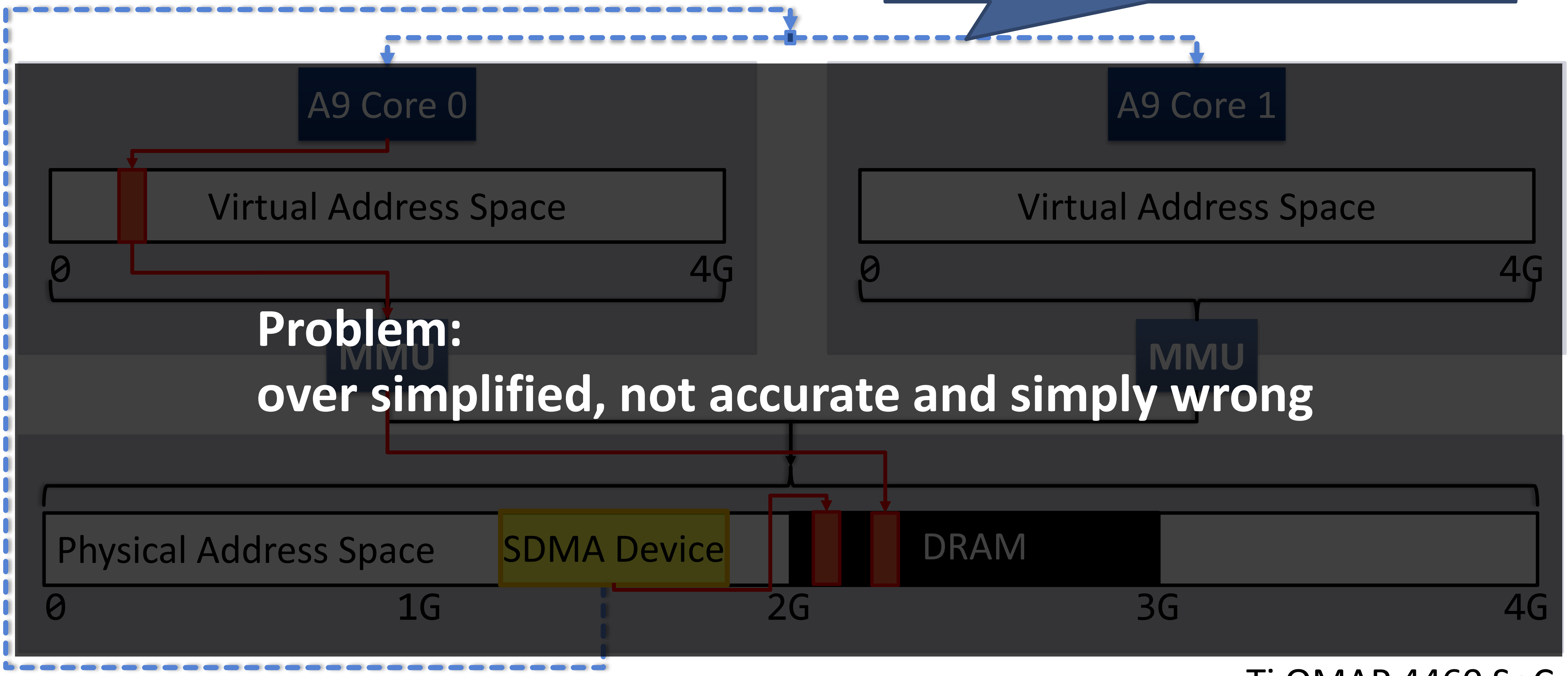
Lukas Humbel "Formalizing Interrupt Routing"



Ti OMAP 4460 SoC

The naïve view of today's systems

Lukas Humbel "Formalizing Interrupt Routing"



**Problem:
over simplified, not accurate and simply wrong**

Ti OMAP 4460 SoC

Reality: The devil is in the details

Your mobile phone… 5-10 years ago!

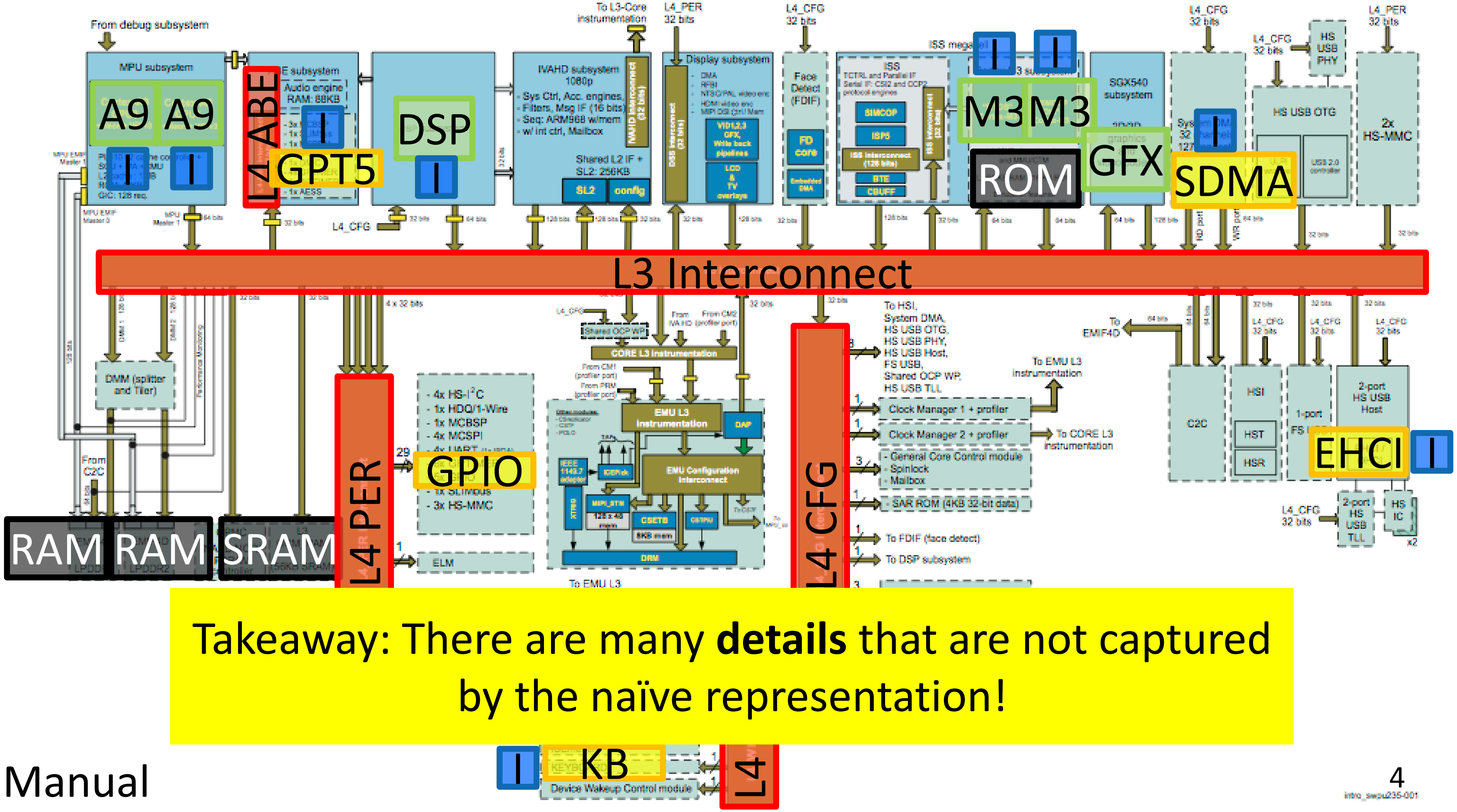
6+ heterogeneous cores

Private and shared memory

5+ Interconnects

Devices attached to different interconnects

Complex interrupt subsystem



Takeaway: There are many **details** that are not captured by the naïve representation!

OMAP 4460 SoC,
Technical Reference Manual

Reality: The devil is in the details

Your mobile phone... 5-10 years ago!

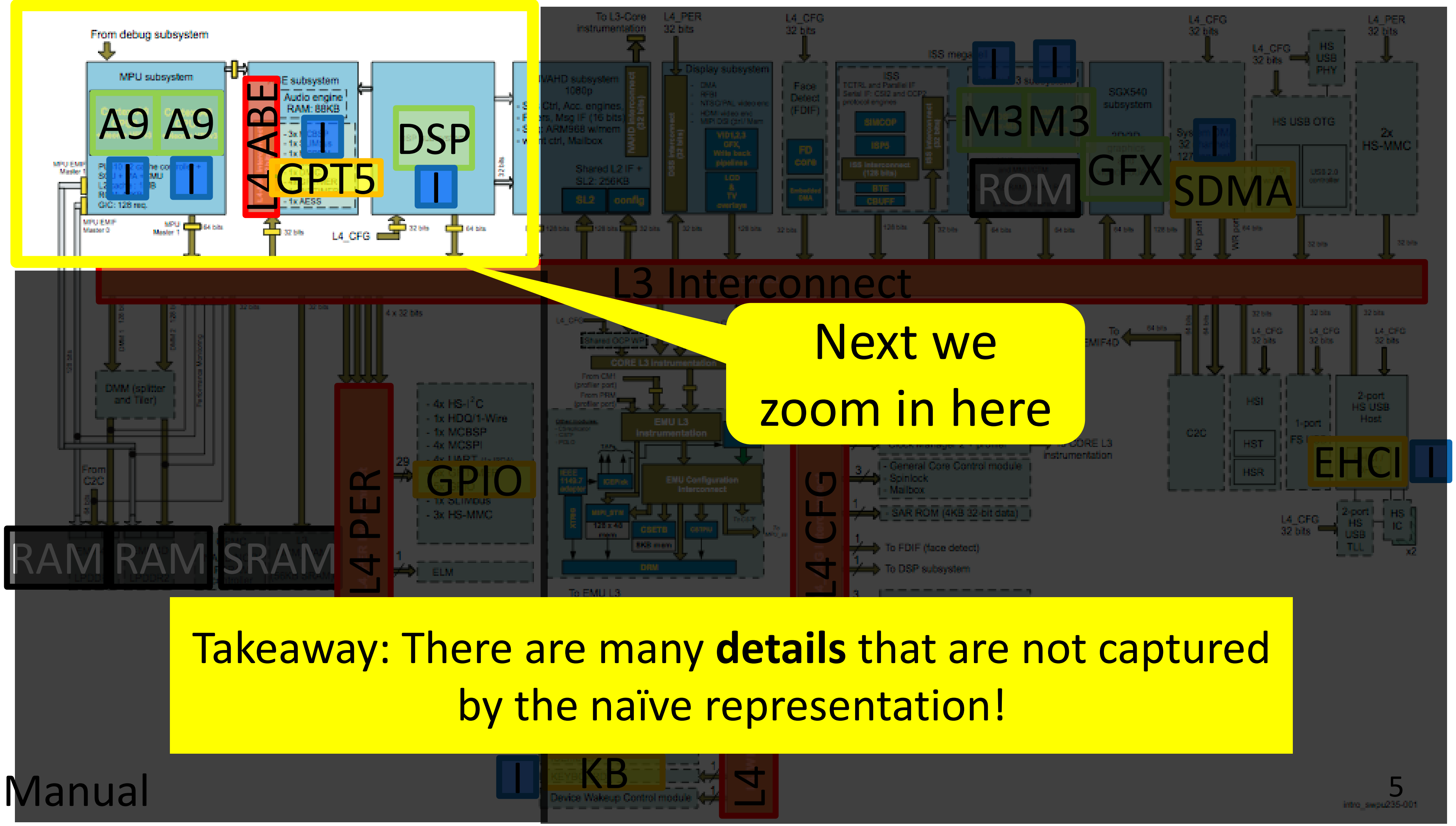
6+ heterogeneous cores

Private and shared memory

5+ Interconnects

Devices attached to different interconnects

Complex interrupt subsystem



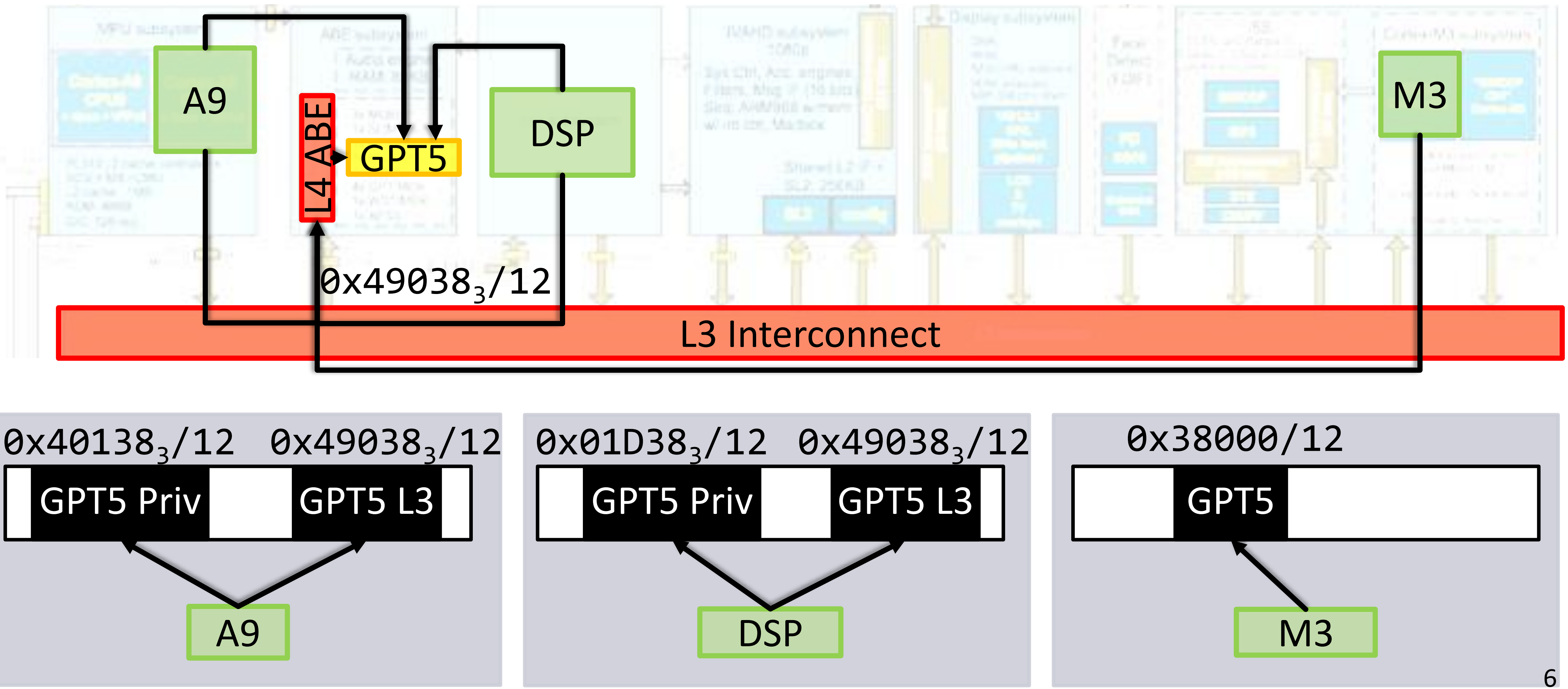
Next we zoom in here

Takeaway: There are many **details** that are not captured by the naïve representation!

OMAP 4460 SoC, Technical Reference Manual

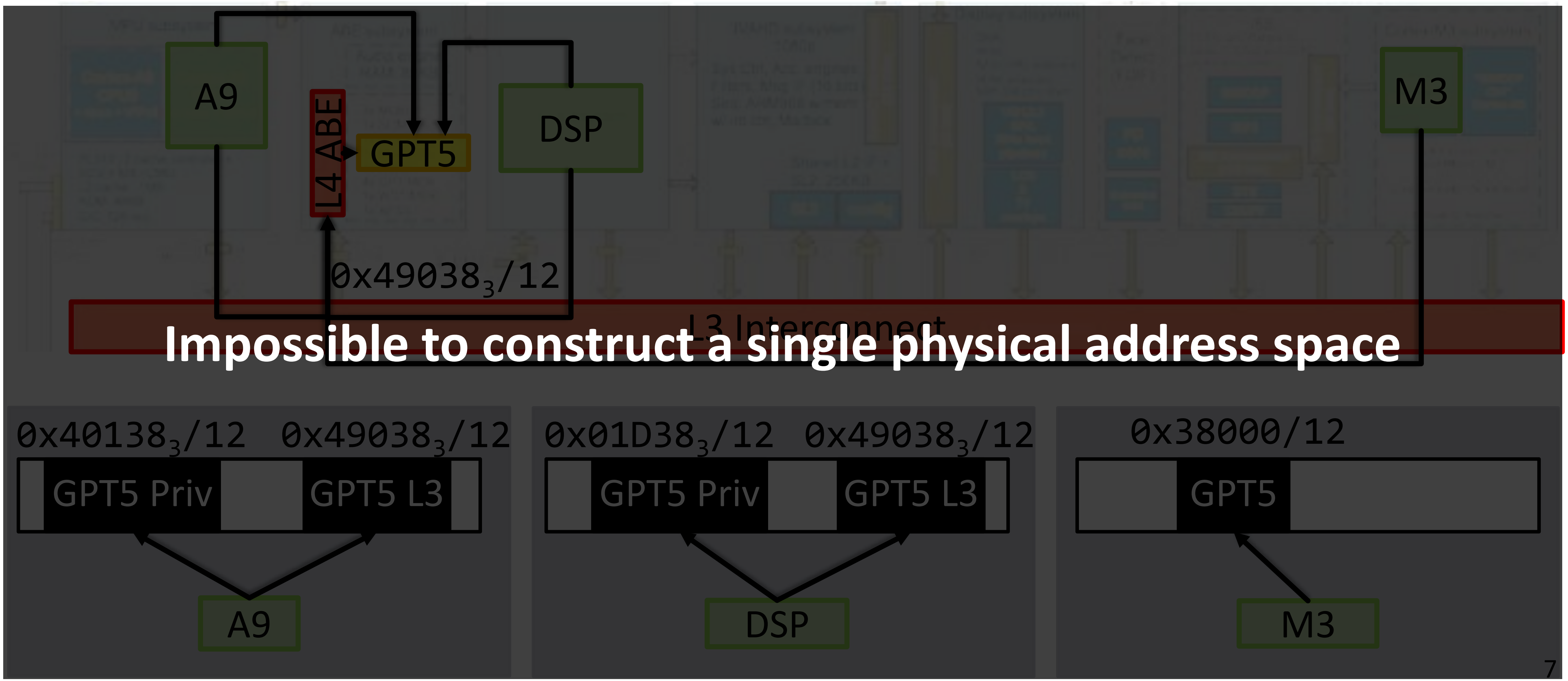
There is NO uniform view of the system

OMAP 4460 SoC



There is NO uniform view of the system

OMAP 4460 SoC

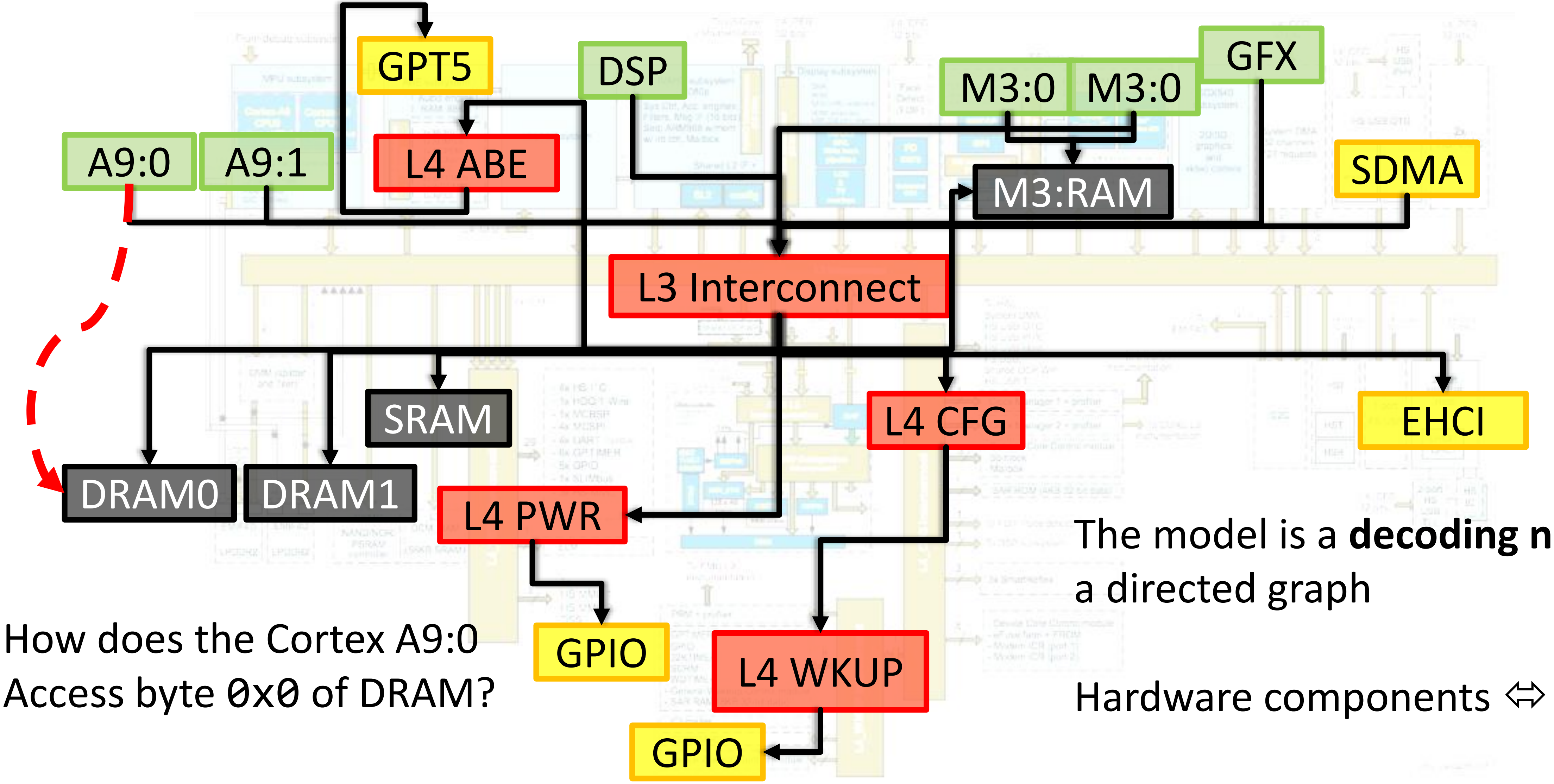


Impossible to construct a single physical address space

Why do we need a formal model for memory accesses ?

- We **build** systems and want to write **correct** systems code
- Experience from the Barrelfish operating system:
  dealing with this complexity every day.
  e.g. PCI programming, heterogeneity, resources, devices, new platforms
- Problem:
  - Current abstractions make the **wrong** assumptions
  - System software verification requires a sound system hardware description

A partial decoding net for the OMAP4460

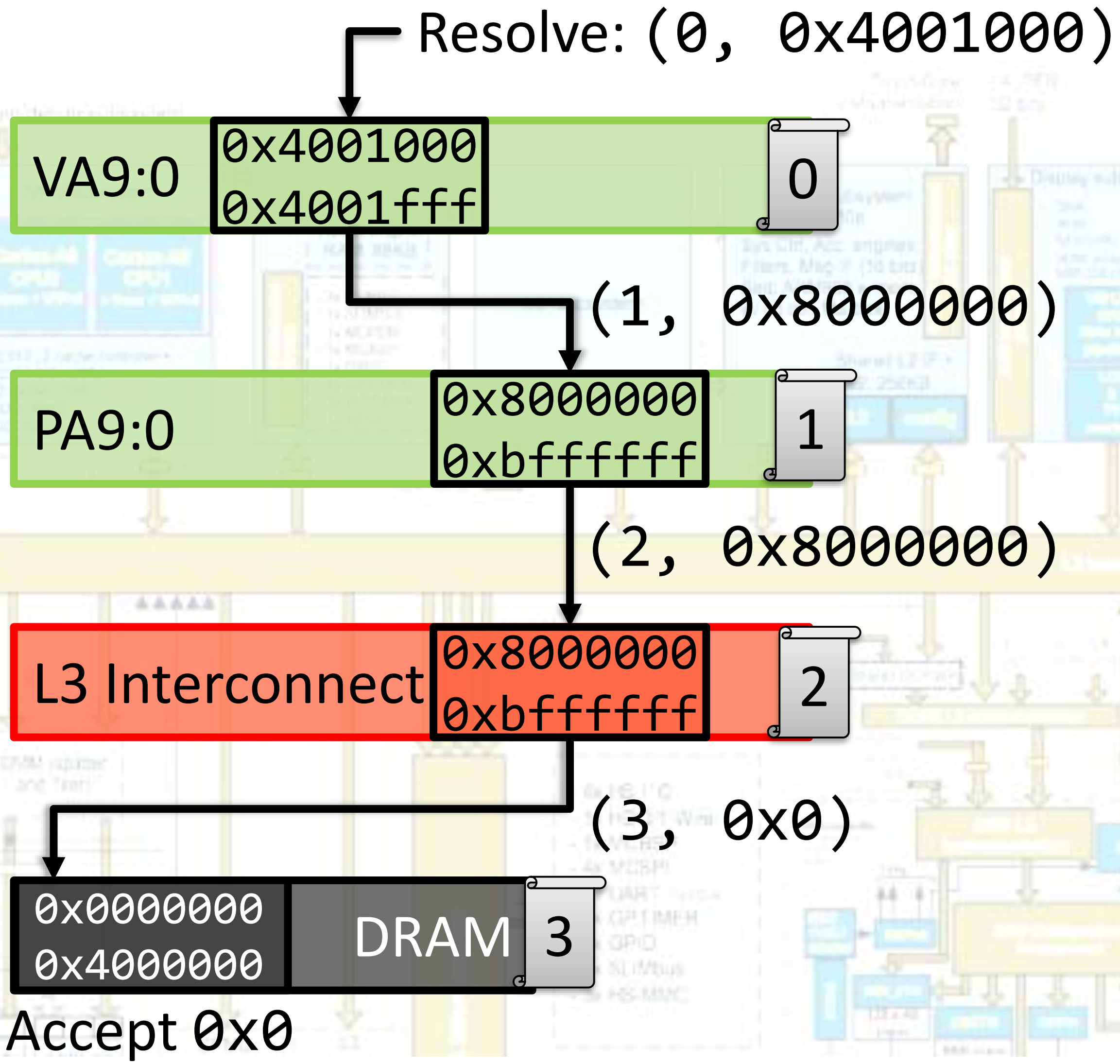


How does the Cortex A9:0 Access byte `0x0` of DRAM?

The model is a **decoding net,** a directed graph

Hardware components ⇔ **nodes**

Modelling the access to byte 0x0 of DRAM from an A9 core



Resolve: (0, 0x4001000)

VA9:0 0x4001000 0x4001fff 0

(1, 0x8000000)

PA9:0 0x8000000 0xbffffff 1

(2, 0x8000000)

L3 Interconnect 0x8000000 0xbffffff 2

(3, 0x0)

0x0000000 0x4000000 DRAM 3

Accept 0x0

Each node has a label

Resolve a **name** (node, address)

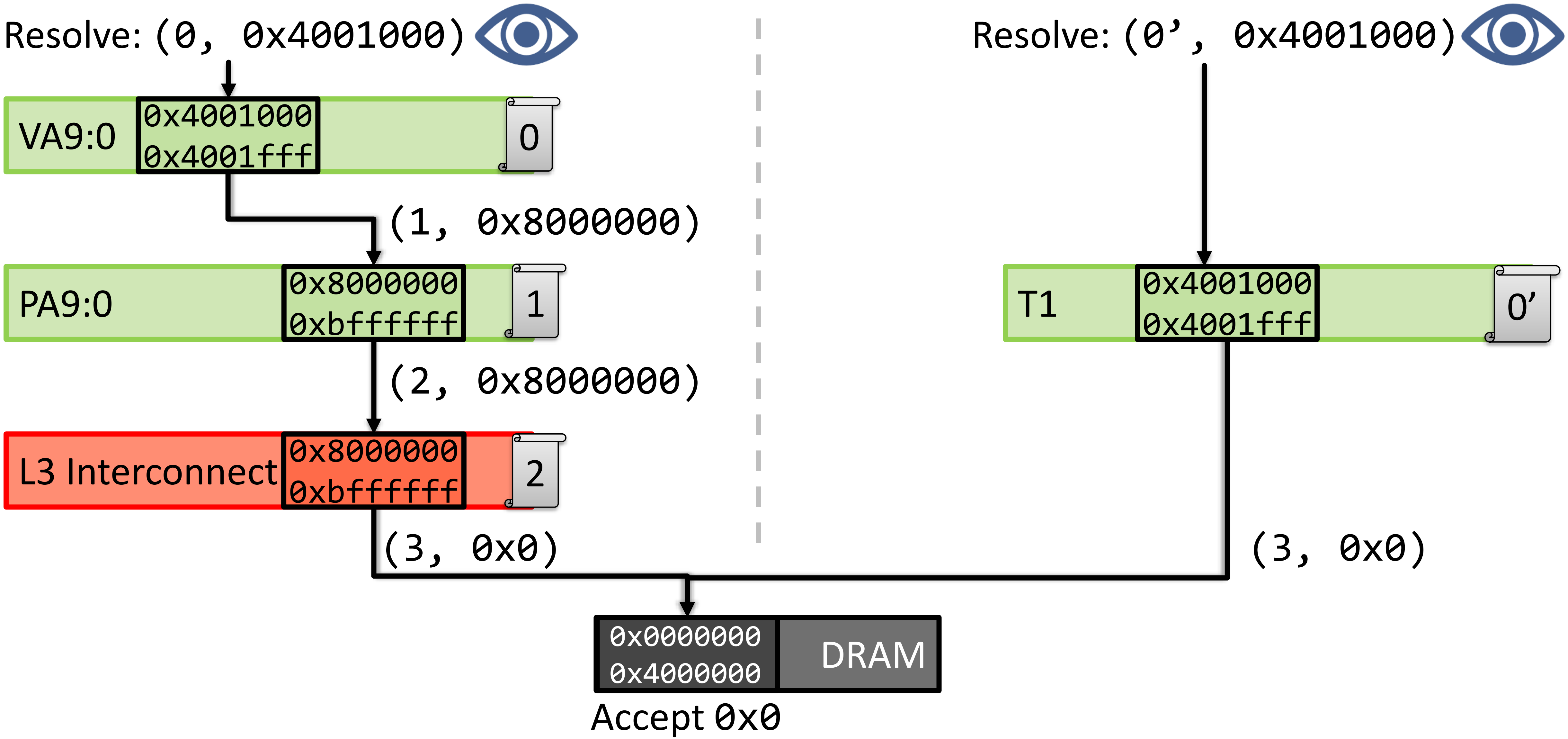
Model of one particular, static configuration state

Nodes have **two properties**:

$accept$: $node \rightarrow \{\mathbb{N}\}$

$translate$: $node \rightarrow \mathbb{N} \rightarrow \{name\}$

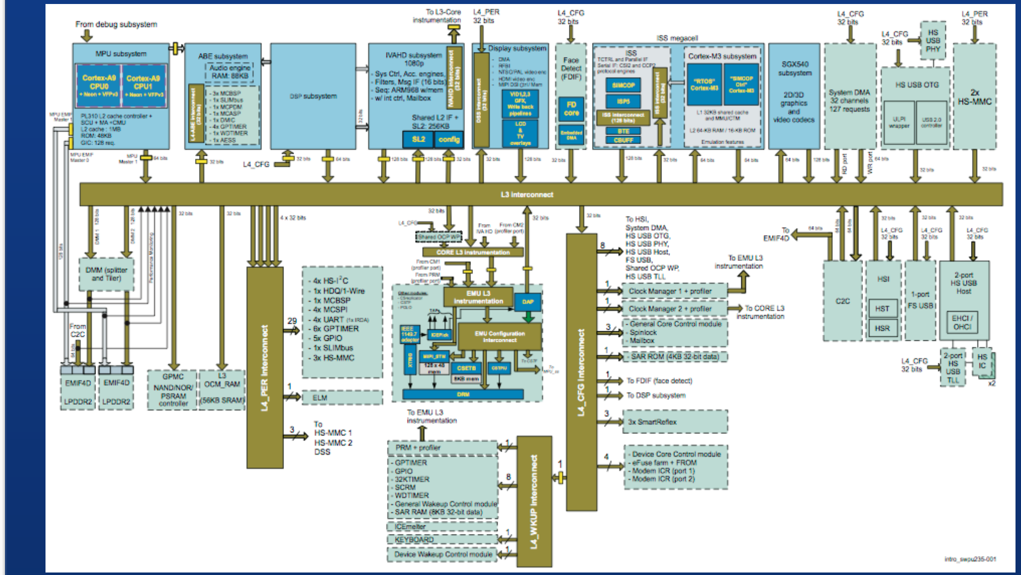
Flattening using view equivalence preserving operations

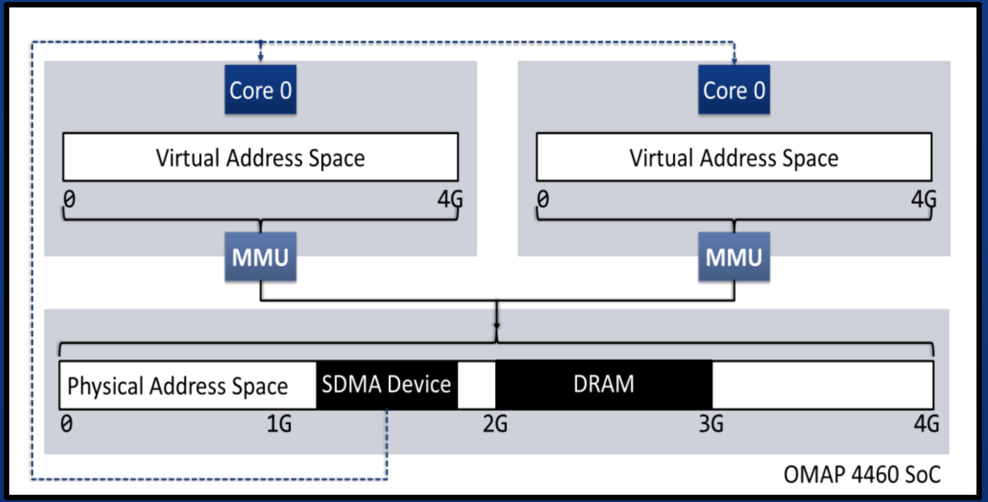


Flattening using view equivalence preserving operations

Resolve: (0, 0x4001000)

Resolve: (0’, 0x4001000)



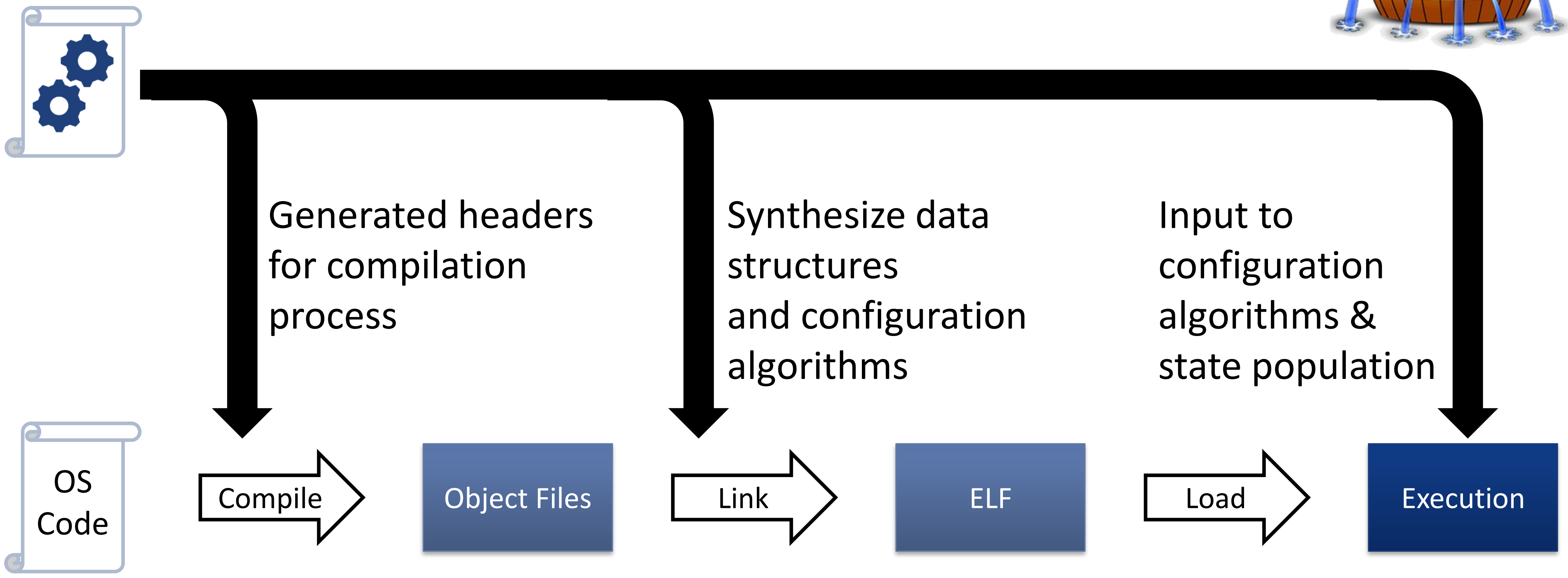


For ONE observer the flattened representation is equivalent to the textbook abstraction

Accept 0x0

Ongoing work: Using model output at compile and run time

Model

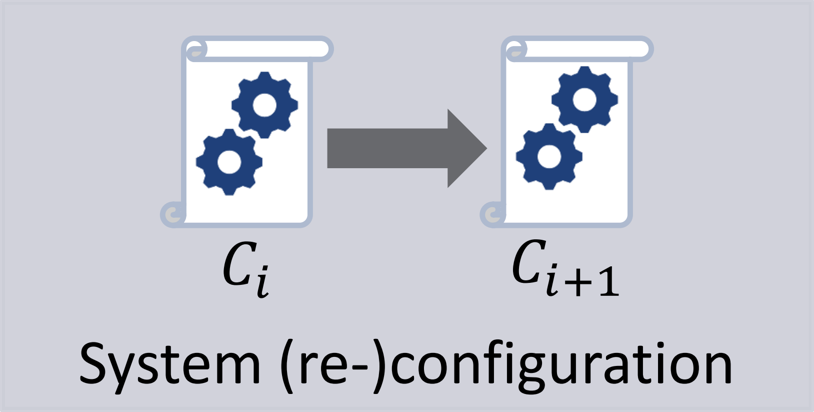


Generated headers for compilation process

Synthesize data structures and configuration algorithms

Input to configuration algorithms & state population

Ongoing Work: Model applications



- Generate system configuration from the model:
  - Kernel page tables
  - Initial capabilities
- **Synthesize** configuration algorithms
- **Transition** between configurations without violation of **invariants**
- **Constraints** on memory accesses

Future work: Model refinements

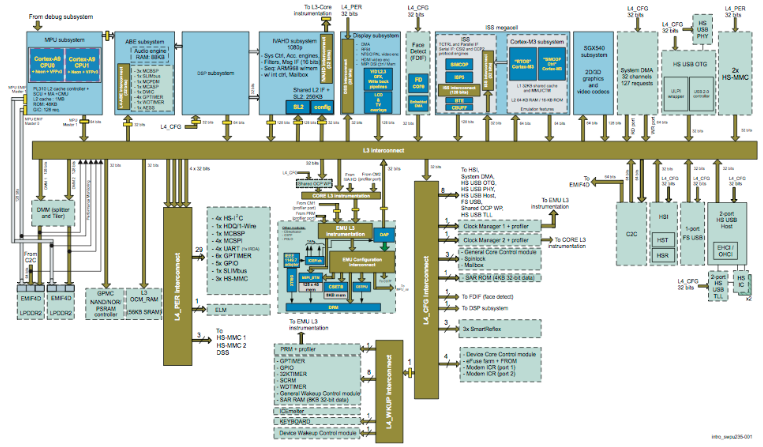
Distinction of Read/Write accesses

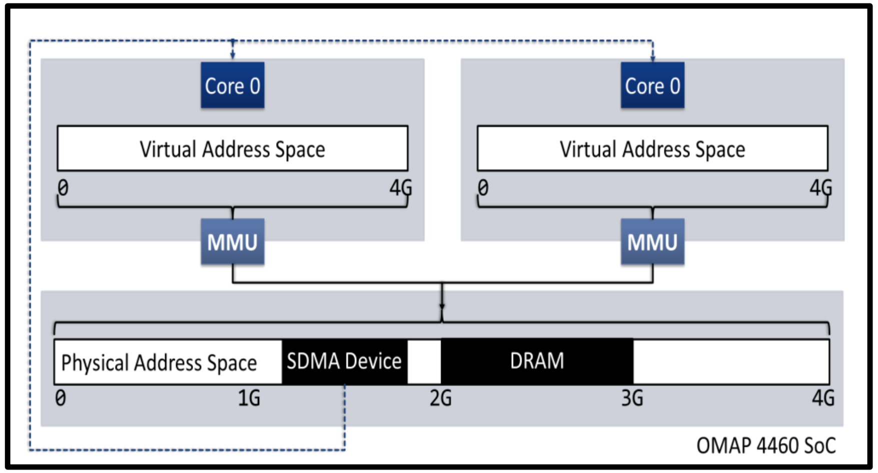
- Reads / writes have different semantics
- Write only / read only regions

Expressing performance characteristics

- Basis for a performance model.
- Resource allocation & scheduling

Summary





$V_{A9:0}$ **is map** $[\mathtt{20000}_3/12$ **to** $P_{A9:0}$ **at** $\mathtt{80000}_3]$

$P_{A9:0}, P_{A9:1}$ **are map** $[\mathtt{40138}_3/12$ **to** GPT **at** $\mathtt{0}]$ **over** $L3$

P_{DSP} **is map** $[\mathtt{1d3e}_3/12$ **to** GPT **at** $\mathtt{0}]$ **over** $L3$

V_{M3}, V_{M3} **are over** $L1_{M3}$

RAM_{M3} **is accept** $[\mathtt{55020}_3/16]$

ROM_{M3} **is accept** $[\mathtt{55000}_3/14]$

$V_{A9:1}$ **is map** $[\mathtt{20000}_3/12$ **to** $P_{A9:1}$ **at** $\mathtt{80000}_3]$

V_{DSP} **is over** P_{DSP}

$L2_{M3}$ **is map** $[\mathtt{0}_{30}$ **to** $L3$ **at** $\mathtt{80000}_3]$

$L1_{M3}$ **is map** $[\mathtt{0}_{28}$ **to** $MIF]$

$L4$ **is map** $[\mathtt{49038}_3/12$ **to** GPT **at** $\mathtt{0}]$

GPT **is accept** $[\mathtt{0}/12]$

MIF **is map** $[\mathtt{0} - \mathtt{5ffffff}$ **to** $L2_{M3}, \mathtt{55000}_3/14$ **to** $RAM_{M3}, \mathtt{55020}_3/16$ **to** $ROM_{M3}]$

$L3$ **is map** $[\mathtt{49000}_3/24$ **to** $L4$ **at** $\mathtt{40100}_3, \mathtt{55000}_3/12$ **to** $MIF]$ **accept** $[\mathtt{80000}_3/30]$

Configuration