Imperial College London

SynADT: Dynamic Data Structures in High Level Synthesis

Zeping Xue, David Thomas
zeping.xue10@imperial.ac.uk

FCCM 2016, Washington DC
2 May 2016

What are ADTs?

Abstract Data Types(ADTs)

Interface + functionality

e.g. Dictionary, Queue, Stack

Concrete Data Types(CDTs)

Code + Memory layout

e.g. Linked List, Binary Tree

ADT	CDTs are used to realise an ADT
Dictionary	AVL Tree, Red-Black Tree, Hash Table

Why ADTs?

ADTs abstract away representation and implementation

Make it possible to easily change underlying CDT

Where ADTs?

Any application needing more than a basic array…

Is it possible to do ADTs in HLS?
Is it efficient to do ADTs in HLS?

A linked list example in HLS?

malloc() is not supported.

```
void insert(node_t ** head, int data) {
    node_t * node = malloc(sizeof(node_t));
    node->data = data;
    node->next = *head;
    *head = node;
}
```

Pointer to pointer is restricted.

Memory address space mismatch?

Which memory?

Stack for recursions?

What is a null pointer?

Our Proposed SynADT

Hardware memory allocator

~~malloc() is not supported.~~

```
void insert(node_t ** head, int data) {
    node_t * node = malloc(sizeof(node_t));
    node->data = data;
    node->next = *head;
    *head = node;
}
```

~~Pointer to pointer is restricted.~~

Use offsets instead of pointers

~~Memory address space mismatch?~~

~~Which memory?~~ **Pass components as arguments**

~~Stack for recursions?~~ **LIFO based on SynADT linked list**

~~What is a null pointer?~~ **User-defined illegal address**

Is it possible to do ADTs in HLS? ✓
Is it efficient to do ADTs in HLS?

Besides SynADT..

BenchADT

A benchmark that allows cross-platform evaluations.

Is it possible to do ADTs in HLS? ✓

Is it efficient to do ADTs in HLS? ✓ ✗

Messages

Why SynADT? ADTs are possible and usable in HLS

Approach Offset-based structures allow heterogeneity

Benchmark Benchmark allows cross-platform evaluation

Related Works

Work	How they differ from SynADT
Custom Implementation [1]	Dedicated HDL design.
MATCHUP [2]	Code Transformation. No dynamic memory usage. Specific memory layout requirement.
CoRAM++ [3]	Traverses data structures by tracing pointer. Not aiming to create data structures.

SynADT aims to provide software-like abstractions and use true dynamic memory management.

[1]J.Xu, Design and synthesis of a high-speed hardware linked-list for digital image processing
[2]F.Winterstein, MATCHUP: Memory Abstractions for Heap Manipulating Programs
[3]G.Weisz, CoRAM ++: Supporting Data-StructureSpecific Memory Interfaces for FPGA Computing

SynADT – Main Problems and Solutions

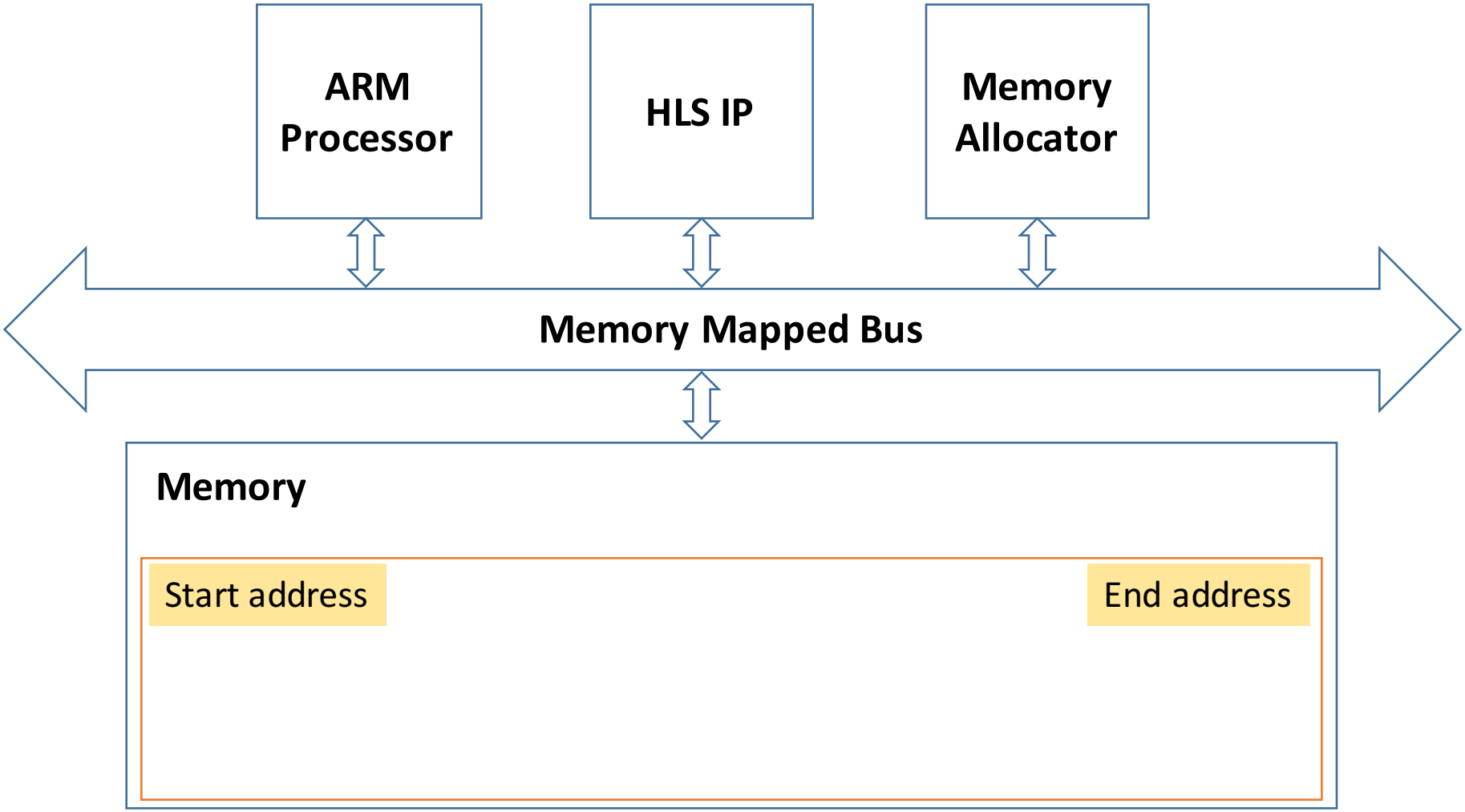
Shared Dynamic Memory Allocation?

Pluggable Hardware Allocator (SysAlloc)

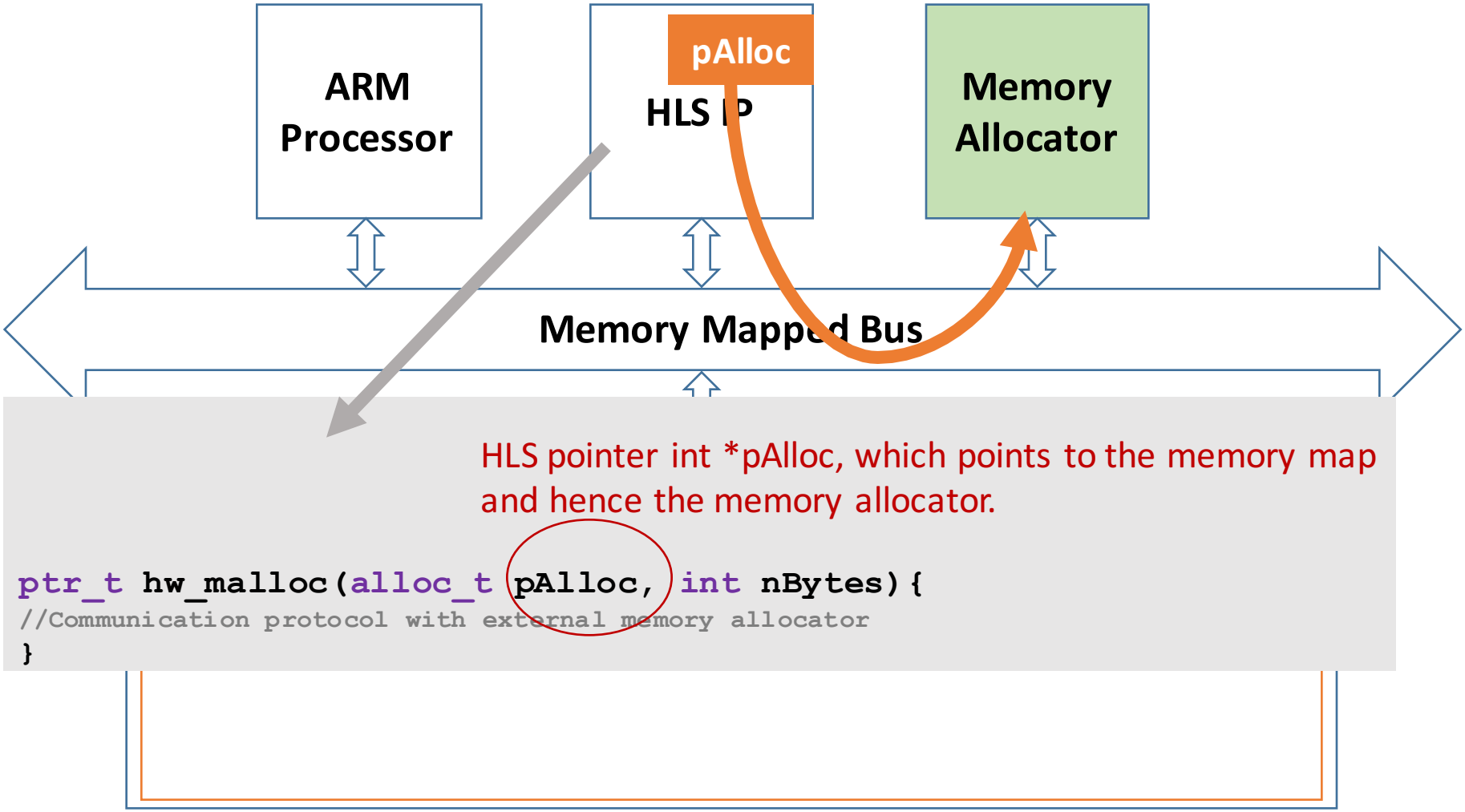
Pointers and Memory Address Space Mismatch?

Offset-based data structures

System Diagram



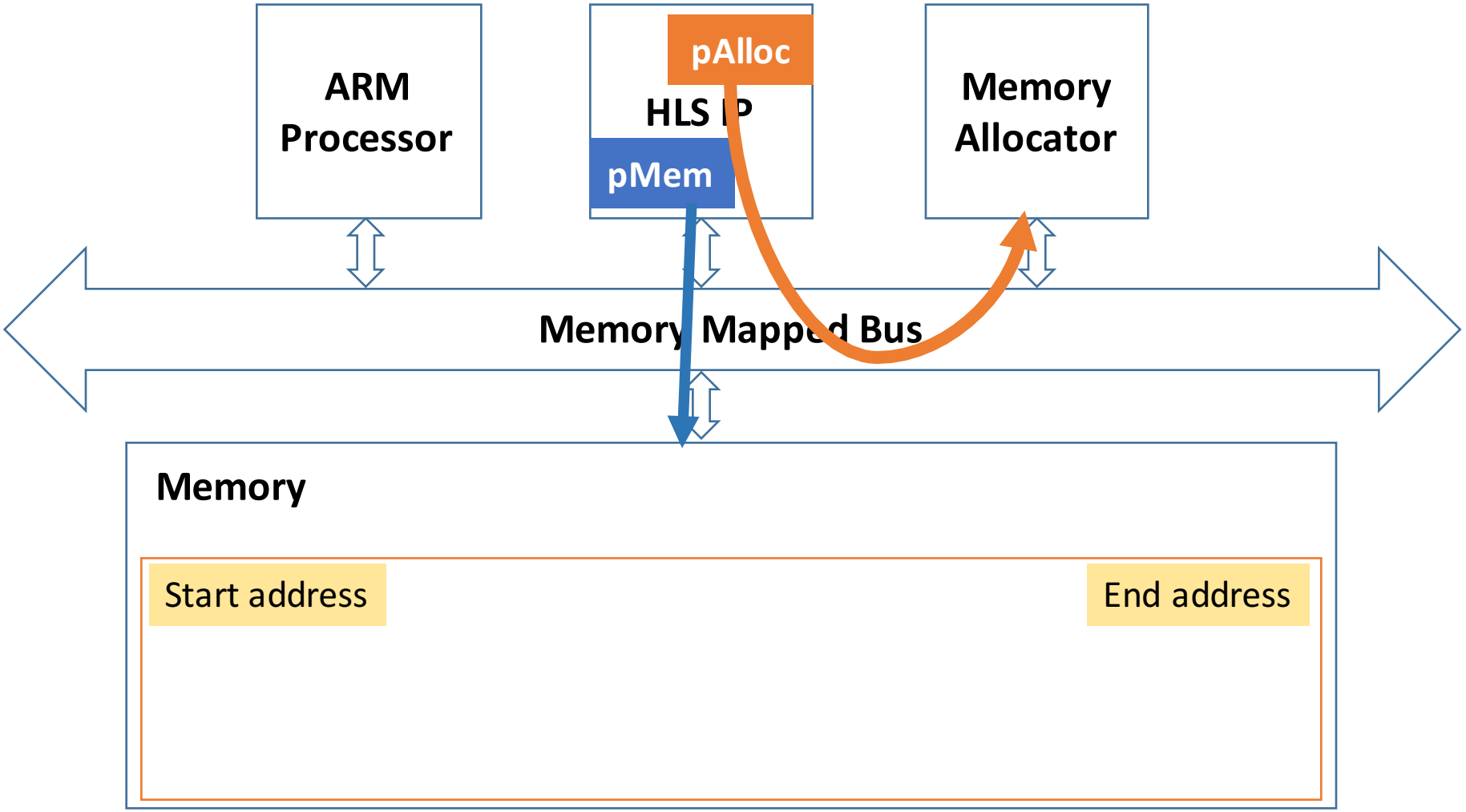
Pluggable malloc



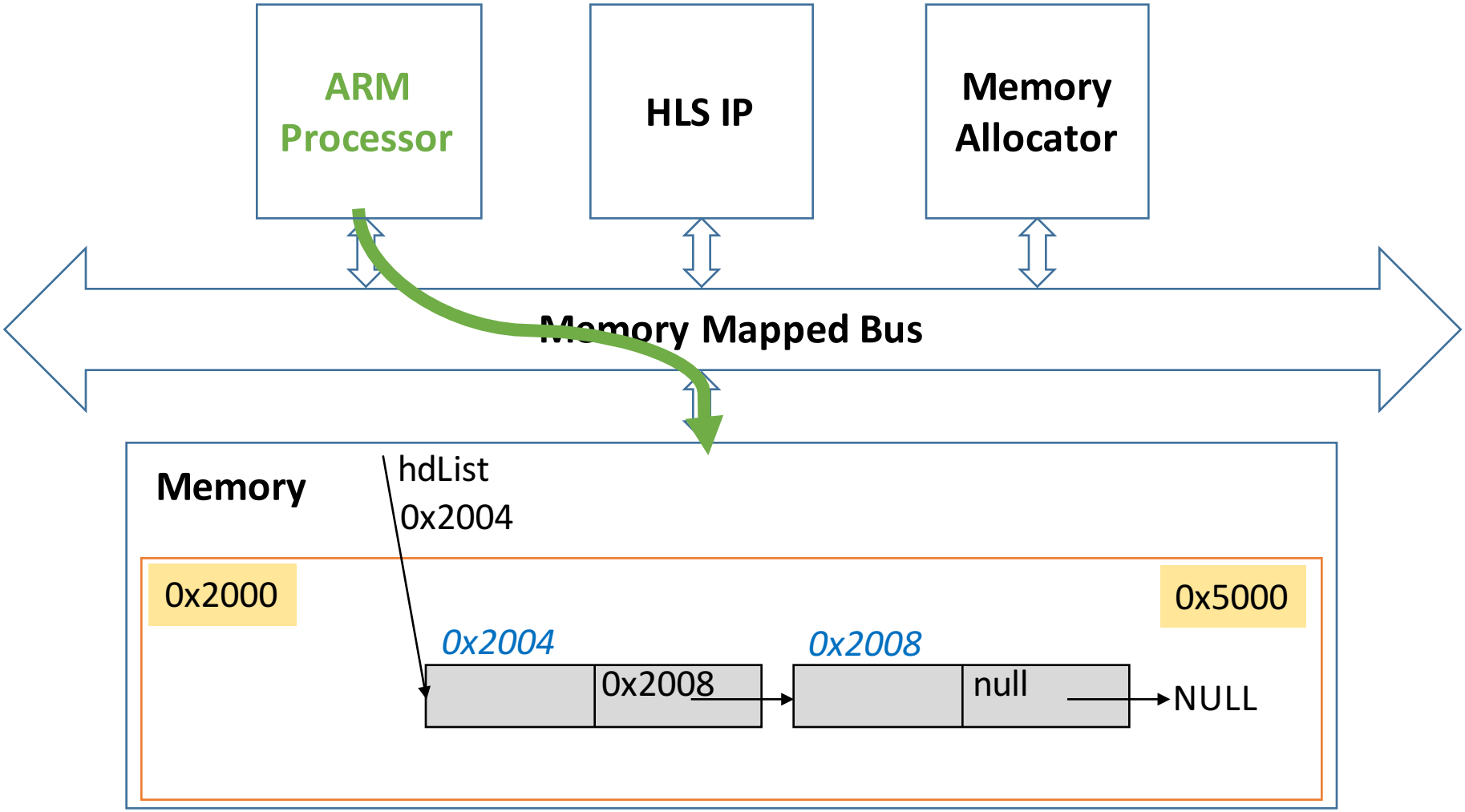
HLS pointer int *pAlloc, which points to the memory map and hence the memory allocator.

```
ptr_t hw_malloc(alloc_t pAlloc, int nBytes){
//Communication protocol with external memory allocator
}
```

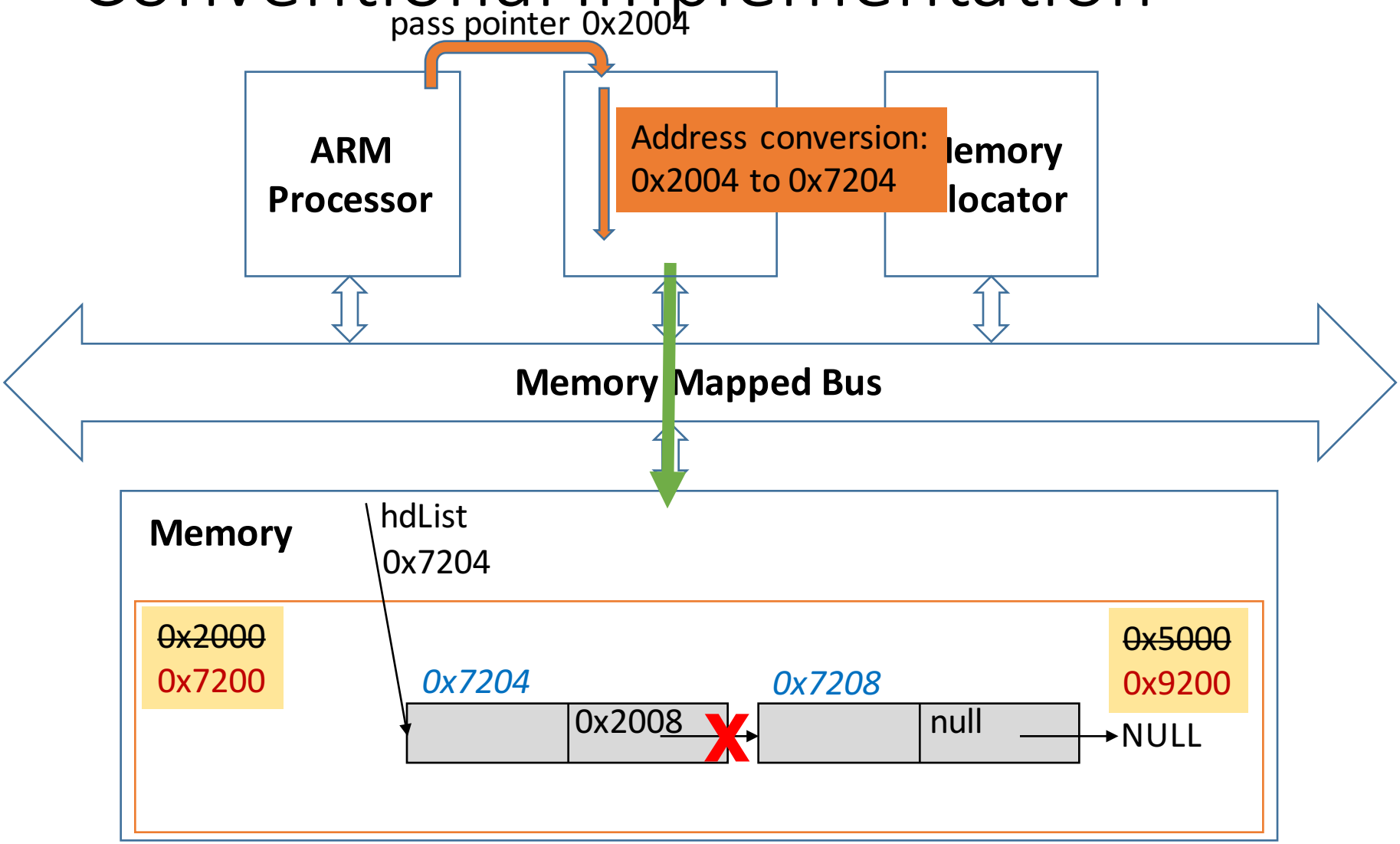
Pass components as arguments



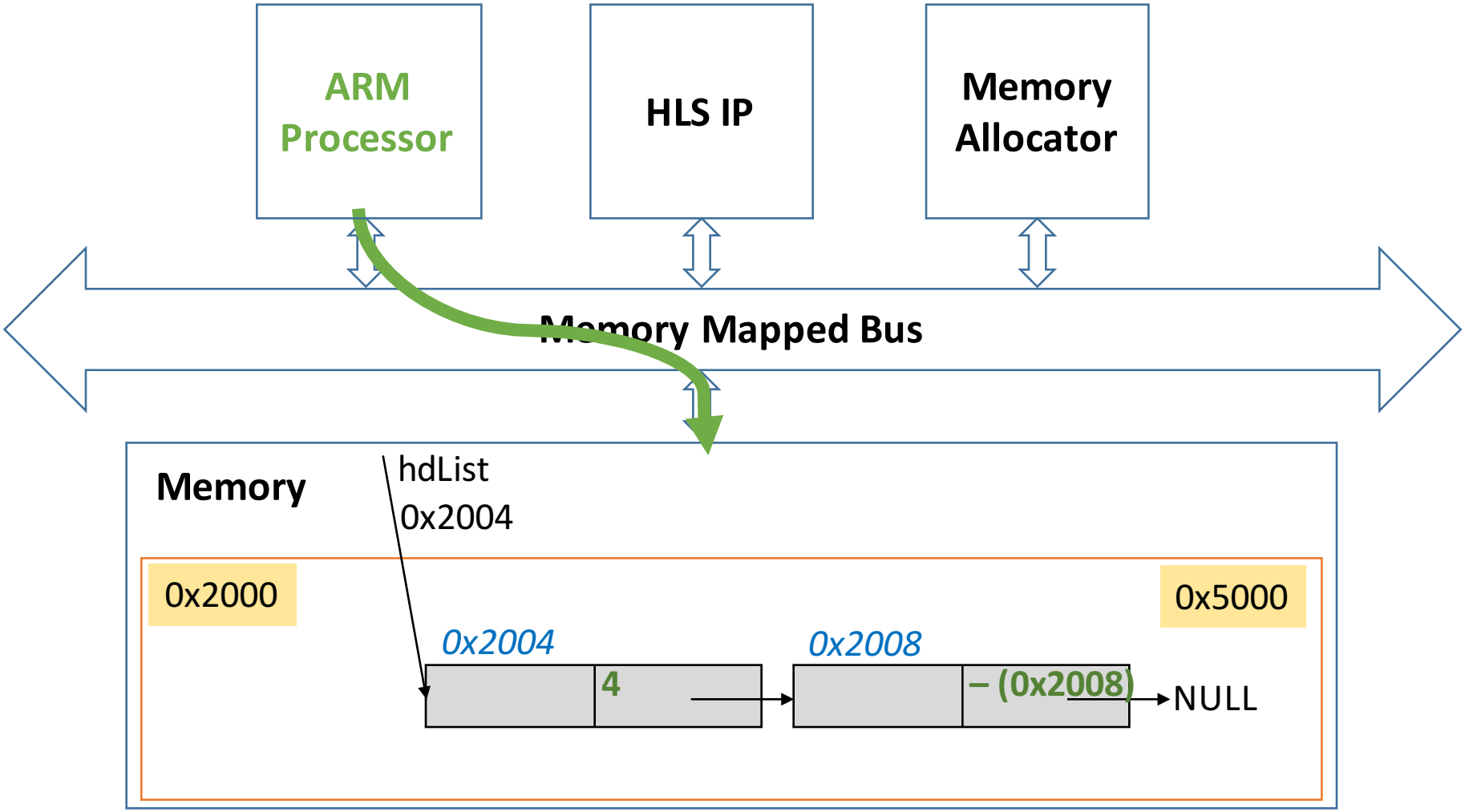
Conventional Implementation



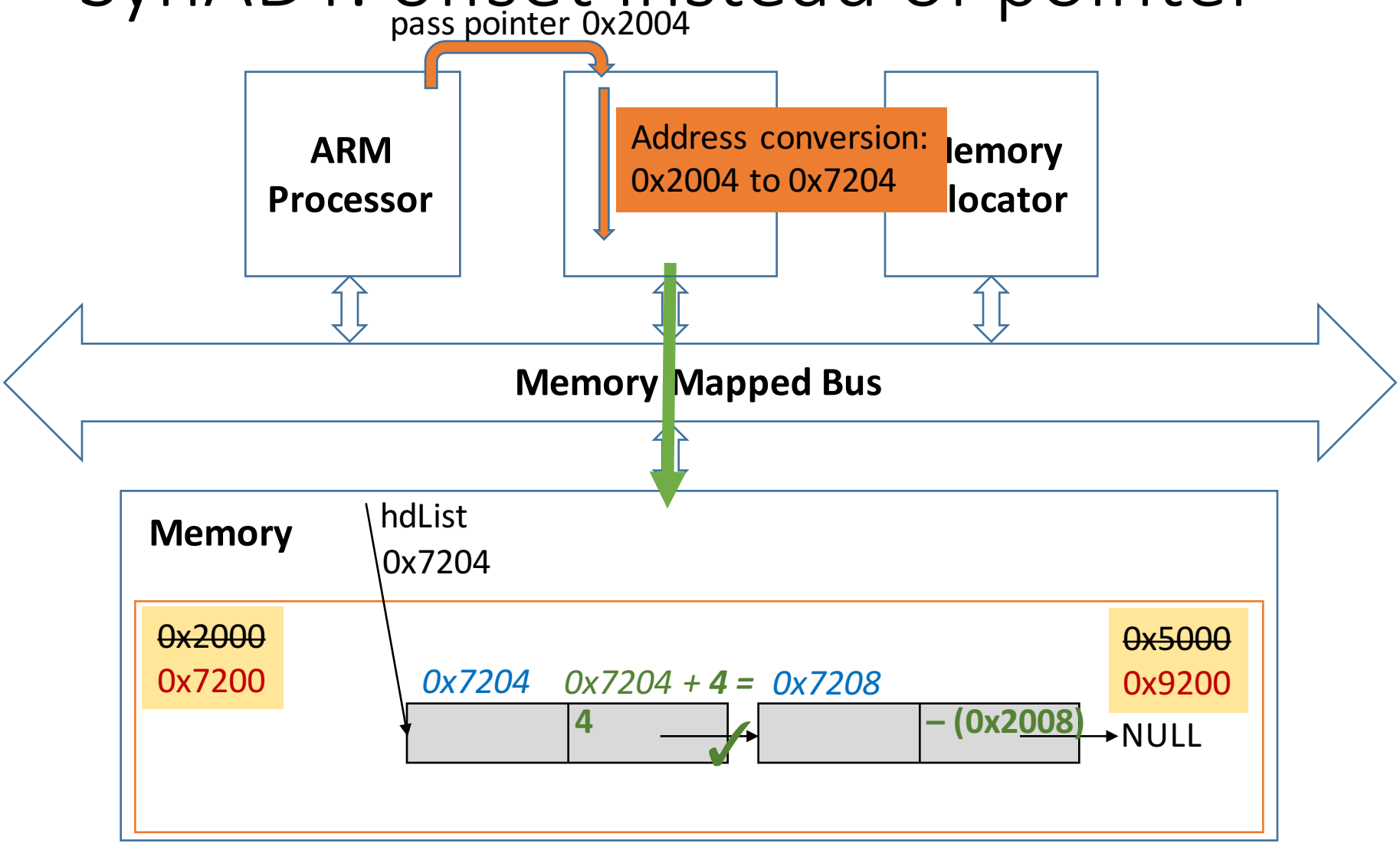
Conventional Implementation



SynADT: offset instead of pointer



SynADT: offset instead of pointer



API

```
/* Operational Functions of Linked List */

/* Insert a node */
list_t list_insert(mem_t pMem, alloc_t pAlloc, data_t data, list_t pList)

/* Reverse the list */
list_t list_reverse(mem_t pMem, list_t pList);

/* Delete the list */
list_t list_delete(mem_t pMem, alloc_t pAlloc, list_t pList);
```

Is it possible to do ADTs in HLS? ✓
Is it efficient to do ADTs in HLS?

BenchADT

Problem:
Multiple data structures with multiple platforms.
We want to analyze performance penalties of each platform.

Portability: Can these results be compared?

Measurement: end-to-end latency per operation.
Analysis: break down into phases (insert/lookup/delete).

Repeatability: Can these tests be repeated?

Use deterministic RNG to produce random inputs.
Clear scaling of structure size.

Evaluation Setup

Device

Zynq-7000 SoC XC7Z02*

Processing Platforms

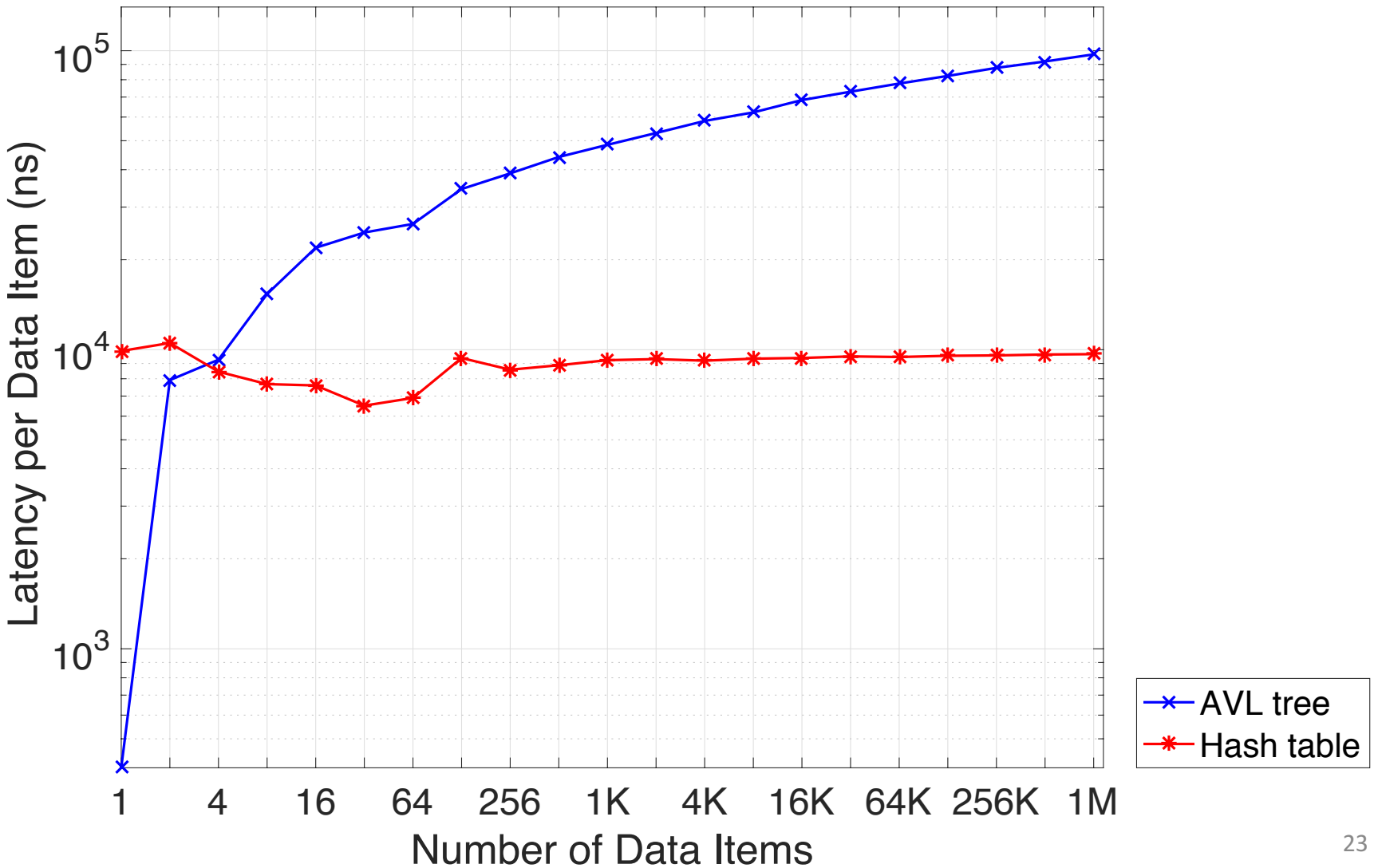
- **F** FPGA @150MHz
- **A** ARM processor @ 667 MHz with 512KB L2 Cache
- **M** MicroBlaze @150MHz with 64KB Cache

Identical data structure code on all platforms.

*AVL tree runs in XC7Z045 for circuit area reason.

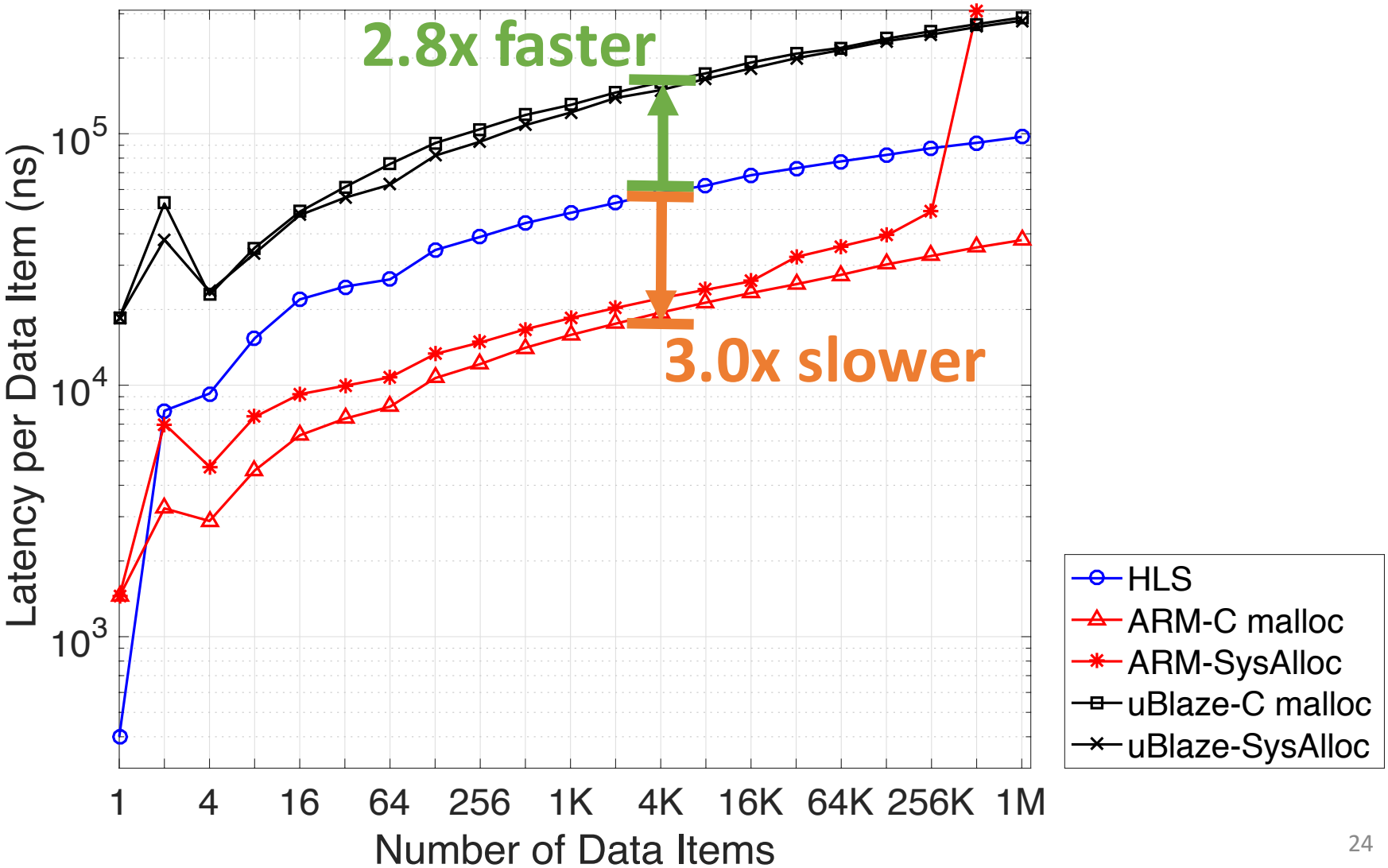
HLS : Dictionary ADT using two CDTs

Operations: Create, Lookup then Insert, Update, Delete Structure



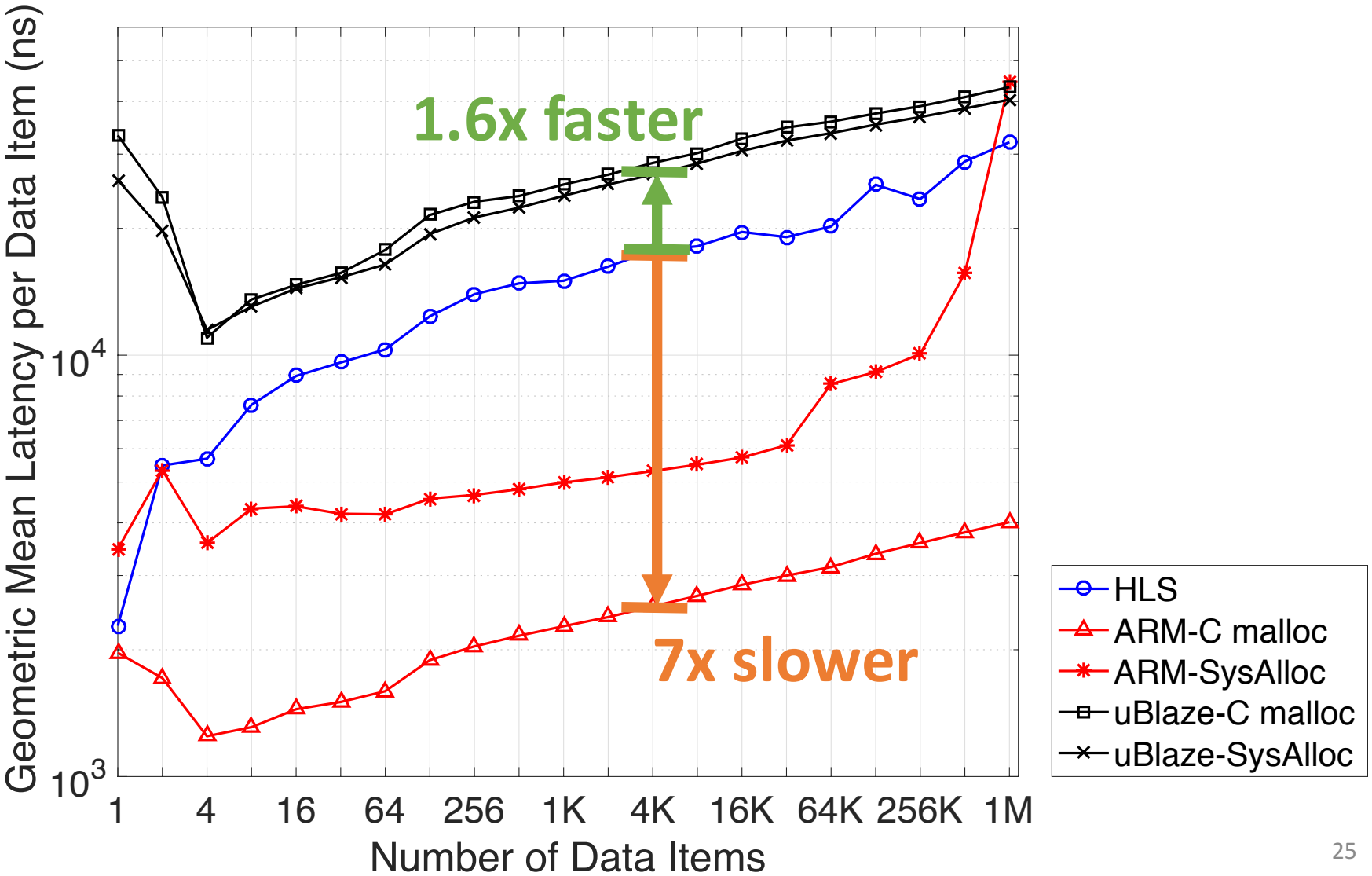
AVL tree in different platforms

Operations: Create, Lookup then Insert, Update, Delete Tree



Average performance across platforms

For all data structures



Average performance across platforms

For 10MB memory usage data structures.

Reference/Alternative		**Reference**
		HLS
Alternative	HLS	1.00
	ARM C malloc	7.97
	ARM SysAlloc	0.72
	uBlaze C malloc	0.74
	uBlaze SysAlloc	0.79

Reference platform geometric mean of latency/alternative platform geometric mean of latency.

Is it possible to do ADTs in HLS? ✓
Is it efficient to do ADTs in HLS? ✓ ✗

Open Questions from SynADT

How much need is there for dynamic data-structures in HLS?

Can ADTs also be shared with RTL?

Should we make hardware development more similar to software?

SynADT : Conclusion

Imperial College London

- SynADT provides dynamic data-structures for HLS
  - Separates implementation from interface using ADTs
  - Allows cross-device sharing of CDTs
- BenchADT allows evaluation of ADTs and CDTs
  - Supports cross-platform evaluation of implementations
- Open source API for dynamic data structures in High Level Synthesis (http://*github.com/Hilx/SynADT*)