HEXO: exokernel for heterogeneous MPSoCs

TSAR: Shared memory, cache-coherent, multi-core architecture

Alexandre Becoulet, Franck Wajsburt, Alain Greiner

October 1, 2008

Table of Contents

Hexo project overview

Hexo tries to bring transparent heterogeneity to application developer.

- Data resides in shared memory.
- All variables and data structures are shared.
- All base code is compiled for each processor.
- Tasks are synchronized using standard primitives.

Design choices

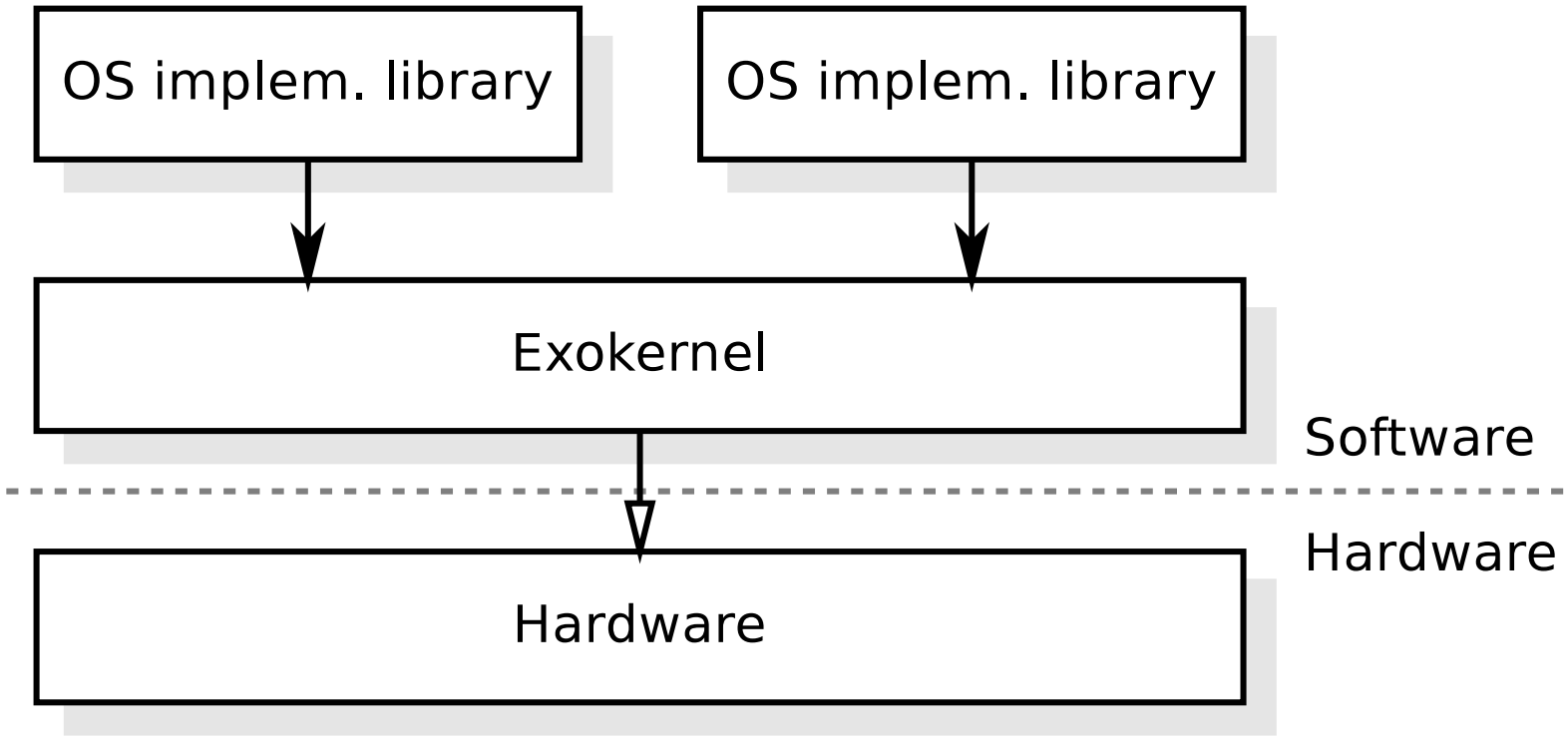
- Hexo design is based on exokernel architecture
- Designed for Heterogeneous multi processors platforms
- Source code is highly configurable
- Support different processor word widths (8 to 64 bits)
- Provide lightweight drivers and device classes
- Designed for easy implementation of OS interface libraries

Exokernel-based operating system

"*The idea behind exokernels is to force as few abstractions as possible on developers, enabling them to make as many decisions as possible about hardware abstractions. Exokernels are tiny, since functionality is limited to ensuring protection and multiplexing of resources.*" – Wikipedia

Libraries can be built on the top of the exokernel to implement different operating system interfaces.

Exokernel architecture

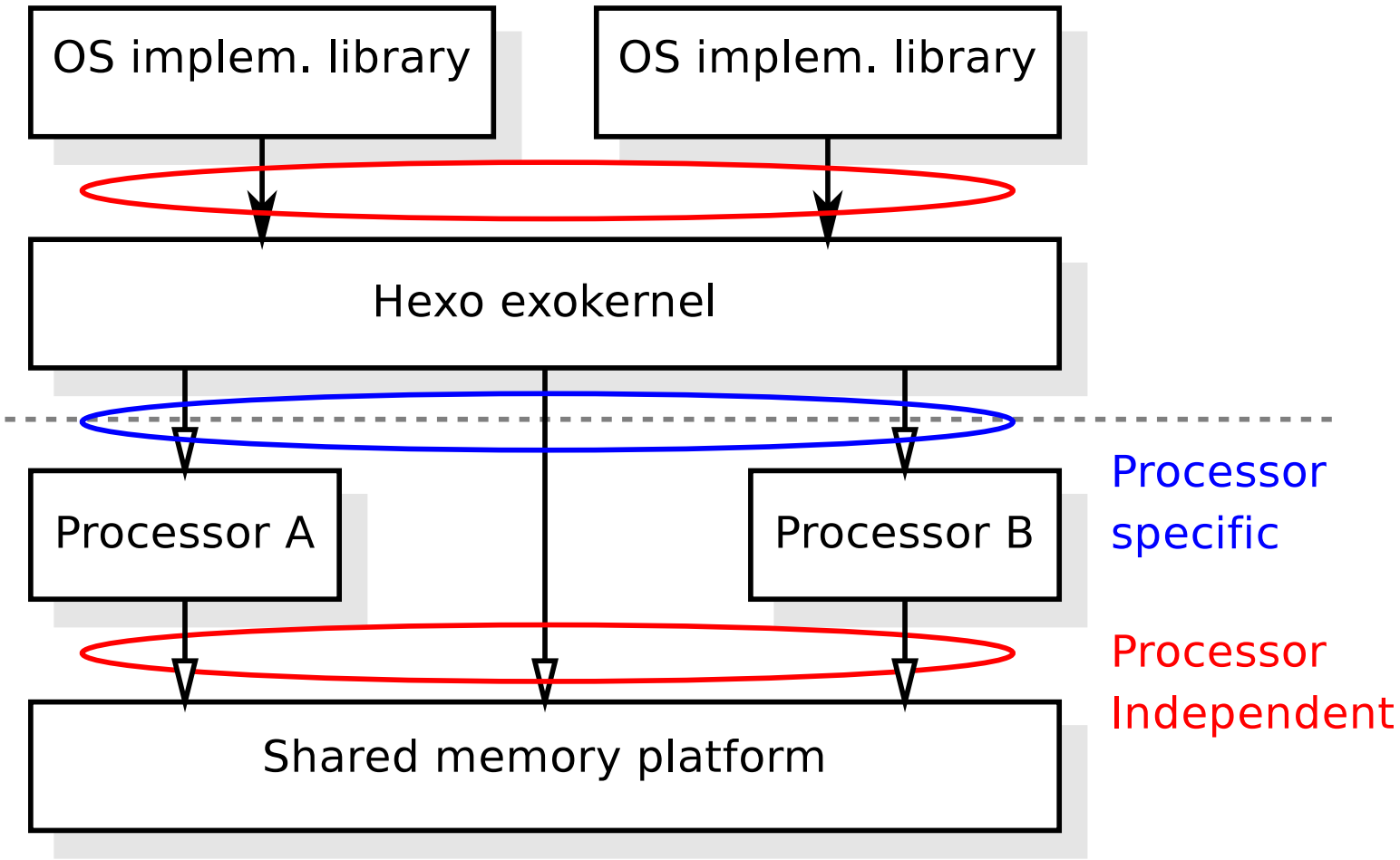


Hexo exokernel primitives

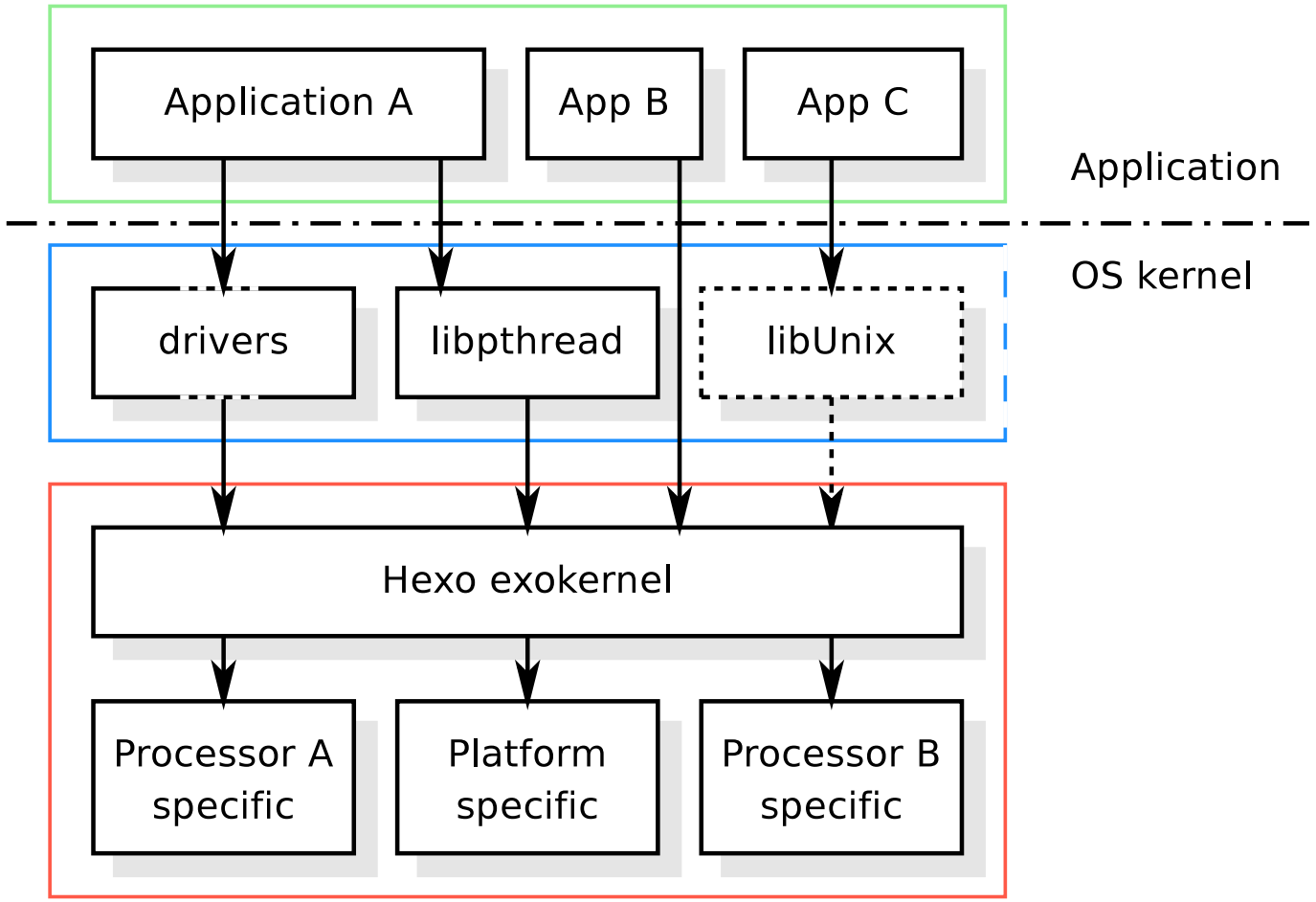
- Processor interrupts, events handling
- Basic context switching and generic scheduler
- Processor IO/memory space access
- Processor and context local storage memory
- Atomic memory access and spin locks
- Memory allocation, paged and protected memory management
- Device driver interface definition

Heterogeneity support in Hexo

Hexo abstraction layer ensures all API calls performs the same action toward the platform shared memory and peripherals. This behavior is guaranteed to be independent of the processor.


OS implem. library
OS implem. library
Hexo exokernel
Processor A
Processor B
Shared memory platform
Processor specific
Processor Independent

Hexo in MutekH project



MutekH development status

The following MutekH modules are currently available and working:

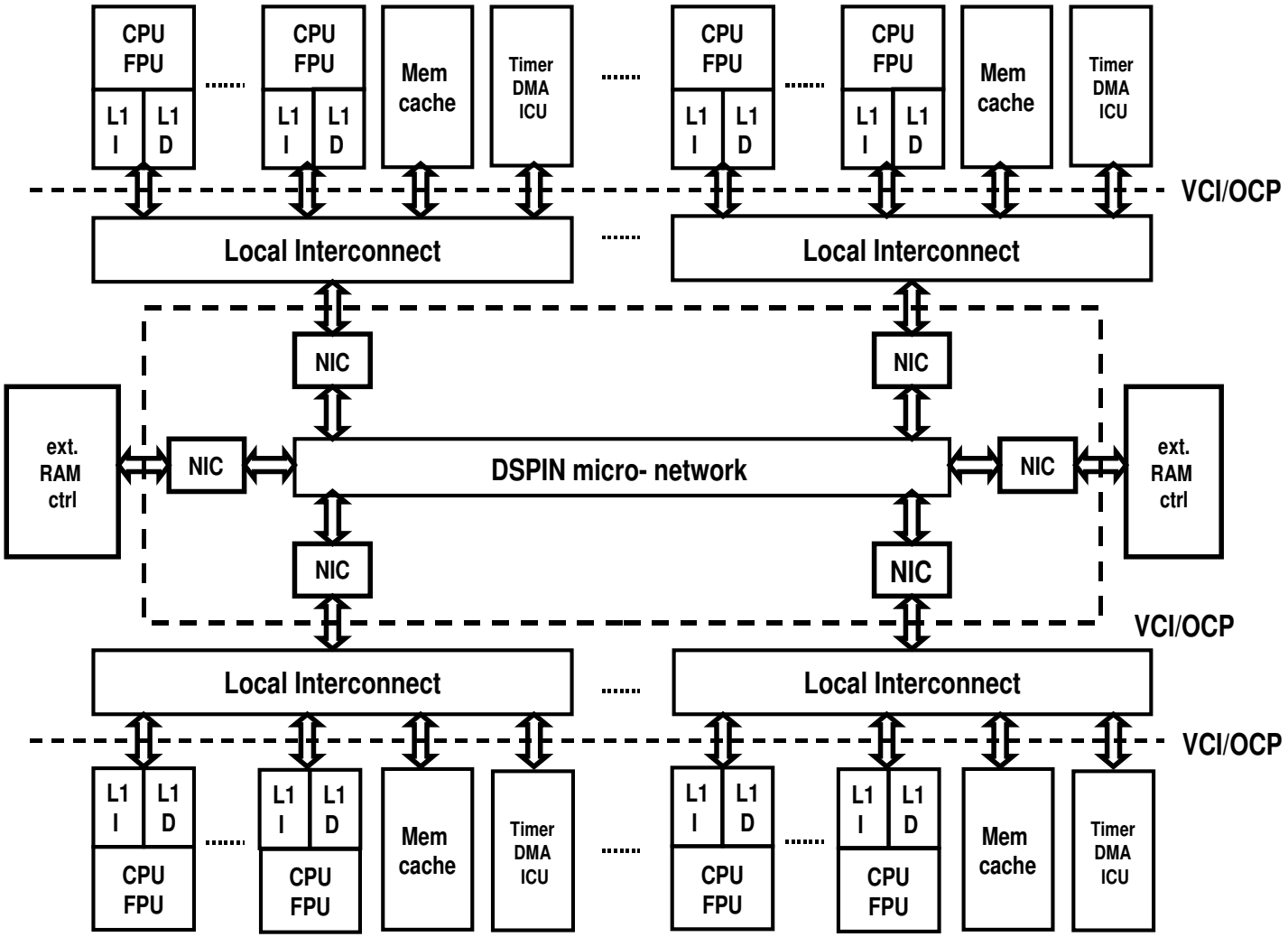
- Standard C library
- TCP/IP suite networking library
- Posix Thread library

The following MutekH modules are under development:

- Virtual File System library
- Unix operating system library

TSAR project overview

- TSAR is a scalable multi-core processor architecture jointly developed by BULL, Thales and Lip6 in the framework of a MEDEA+ European project.
- TSAR will contain a large number of simple 32 bits RISC processors. Its performance should be independant on the selected processor core : SPARC V8, MIPS32, PPC 405, etc.
- The TSAR architecture is clusterized, with a two-level interconnect. The global (inter-cluster) interconnect will use the DSPIN Network on Chip developped at Lip6.
- The Tsar architecture will support a (NUMA) shared address space. The memory is logically shared, but physically distributed, with cache coherence enforced by hardware, using a directory-based protocol.


CPU
FPU
L1
I
L1
D
Mem
cache
Timer
DMA
ICU
VCI/OCP
Local Interconnect
NIC
ext.
RAM
ctrl
DSPIN micro- network

Futur developments

Thales has recently chosen Hexo as Hardware Abstraction Layer for Linux kernel port over TSAR architecture.