AMD
Smarter Choice

Heterogeneous Processing Systems

Heterogeneous Multiset of Homogeneous Arrays (Multi-multi-core)

Processing Heterogeneity

AMD
Smarter Choice

CPU (x86, SPARC, PowerPC)

GPU (AMD/ATI, NVIDIA)

DSP (TI, ADI)

Vector processors (Cray)

Systolic arrays (regaining interest)

String processors (XML engines)

DMA engines

Simple scalar processors (ARM, ARC)

Pattern matchers (CAM/CAS/anti-virus/mal-ware)

Fixed function devices (LAN acceleration, streaming video display)

FPGA supported functionalities

Memory Interface Heterogeneity

AMD
Smarter Choice

Multiple types of memory interface:

- SRAM.
- DRAM.
- Content addressable memory.
- Transactional.
- Stable atomic.
- Active (in-memory operations supported).

Multiple concurrent coherency models:

- Fully coherent.
- Semi-coherent.
- Non-coherent.

Multiple coherency domains.

FPGA supported/created functionalities.

Homogeneity

AMD
Smarter Choice

Zero or more instances of each processor and/or memory type may be present in a given system.

A given solution (application) may desire:

- Many different types of processing and/or memory elements.
- Many instances of the same type of processing and/or memory element.
- Multiple instances of multiple types of processing and/or memory elements.

Different Type of Reconfigurable Computing

AMD
Smarter Choice

HPC has history of building systems to run specific types of applications.

Multi-multi-core creates opportunity to tune virtual server configurations to application requirements.

- Applications written to maximize performance via adoption of assumption of multi-multi-core
- Servers constructed on application specific basis.
- Fast virtual servers get dedicated/shared hardware resources.
- Slow virtual servers emulate multi-multi-core on reduced resource systems.

Dynamic core/hardware assignment.

- System image has stubs that are linked to real hardware when needed.
- System image has access to load balanced work queues.

Sun (SPARC guys) and Symbolics (Lisp guys)

AMD Smarter Choice

How far can the pendulum swing away from:

- Standard hardware architectures?
- Standard software architectures?

Dedicated hardware is faster than general purpose hardware?

Historic deltas between full-custom, AISCs, and FPGAs is diminishing?

- Latency?
- Bandwidth?
- Density?
- Power consumption?

Multi-core will become multi-multi-core?

- Software implications
  - System initialization
  - Operating System
  - Tool support
  - Language integration
  - Programming model
- Overlapping core functionality
  - ADI and TI in same system?
  - X86 and PowerPC in the same system?

Will cost of hardware/software infrastructure force coalescence around a richer, but still very limited number of core types and implementations?

Real Challenge

AMD
Smarter Choice

Specification of generalized communication/messaging model between subsystems.

Specification of para-virtual system level shared memory model for hardware components:
- Physical.
- Guest physical.
- Guest virtual.

Alternatives to current interrupt "sledge-hammer"
- Improved hardware interrupt models for sequestered cores?
- New OS interrupt models?
- Multi-threaded polling architectures?

Can computer science reverse its direction as a pure software science?
- Will computer engineers be the future of "hard-core" system level programming?

What is the desired future software level interface:
- MPI?
- Sockets?
- CORBA?
- ?

Processing Models

AMD
Smarter Choice

Directed processing

- One processing subsystem is tightly controlled by another processing subsystem.
- Example: graphics.

Proxied processing

- Processing subsystem is tied to remote program behavior, but highly autonomous in operation.
- Example: full TCP/IP stack offload.

Peer processing

- Processors interact, but each subsystem operates independently of the other.
- Example: front-end web, back-end data-base processing

Inter-Processor Communication

AMD Smarter Choice

Generally, a processor's I/O is limited to read and write operations.
- Actions must be address and/or data oriented.

Concepts apply equally to homo and heterogeneous communication.

None of these concepts are mutually-exclusive

Control
- Stream of execution
- Pre-emptive (interrupt)
- Co-operative

Metadata
- Register
- Message queue
- Cache-line messaging

Data
- System memory
- Special purpose memory
- Register
- Cache

Start-up Requirements

AMD
Smarter Choice

Initialization

- Booting all the pieces
- Usurpation of authority (emperor crowns itself)
  - Hypervisor/VM/VMM initialized.
  - Order imposed.

Discovery

- Fixed logic
- Reconfigurable logic
  - Fixed set of dynamically assignable profiles.
  - Functionally dynamic, assignable hardware resource.
    - System does not understand functionality beyond ability to assign resource to virtual server.

Allocation

- Creation of virtual servers within physical server.
- Hard partitioning of non-shareable resources.
- Protection of shareable resources.

Interconnect Model

Intra-Subsystem / Inter-Subsystem / Subsystem to Memory

AMD
Smarter Choice

Homogeneous interconnect

Heterogeneous interconnect

Create distinction between homogeneous and heterogeneous?
- GPU/GPU conundrum.

Shared memory

Partitioned memory

Shared and partitioned memory

Coherency
- Cache effects

Virtualization
- Paging
- Hypervisor/VM

I/O Model

AMD Smarter Choice

What is I/O in multi-multi-core environment?

Shared I/O.

Intelligent / off-load models.

- I/O becomes fixed function sub-system.

IOV (I/O virtualization).

- Partitioned multi-function devices.

Dedicated I/O.

IOV (I/O virtualization).

- Partitioned multi-function devices.

Ethernet hegemony.

I/O == Ethernet.

How far into the box does Ethernet extend?

Ethernet as new graphics interface.

iSCSI as EIDE/SATA/SAS/SCSI/FibreChannel replacement.

Impact of intelligent chameleon like logical interfaces to physical Ethernet based tunneling interfaces.

Programming Model

AMD
Smarter Choice

Compiler based multi-multi-core.

- Homogeneous.
- Heterogeneous.
- Explicit.
- Implicit.

Library based multi-multi-core.

Previous problems relatively easy compared to this one.

Can multi-multi-core programming model ever become practically useful?

- Restricted set of functionalities folded into standard system software components (libraries, compilers, assemblers, linkers, run-time systems).
- Pray for painless death if your problem is outside addressed multi-multi-core integration solutions.

AMD is interested in addressing the issues associated with multi-multi-core computing.

AMD has no interest in operating in a vacuum as it addresses these problems.

AMD is soliciting your input.

Should not be confused with an invitation to fund your research.

– Dr. David Mayhew

– d.mayhew@charter.net