MANCHESTER 1824

The University of Manchester

Virtualization and chip multiprocessor memory management: the JAMAICA architecture

Dr. Ian Rogers, Mohammad Ansari,

Matthew Horsnell, Prof. Ian Watson

The Advanced Processor Technologies group

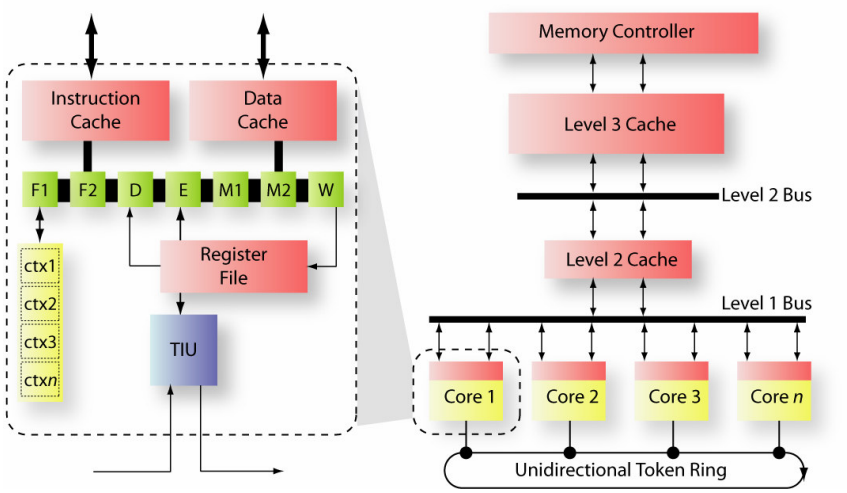
http://www.cs.manchester.ac.uk/apt

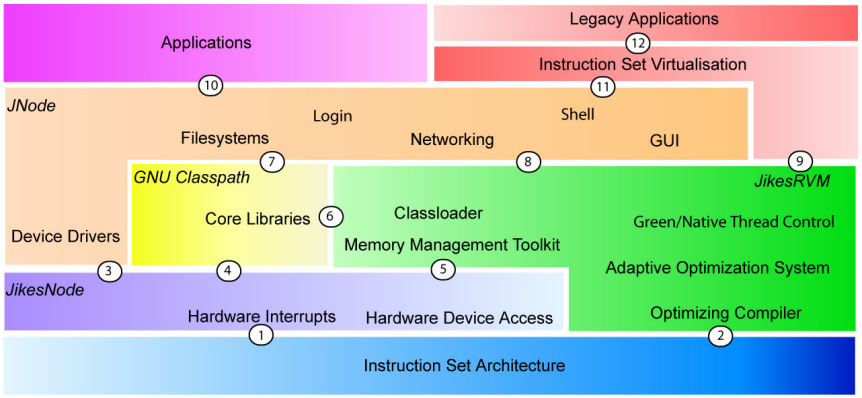
MANCHESTER 1824

The University of Manchester

Presentation outline:

- The JAMAICA Chip Multiprocessor architecture:
  - overview
  - a core and more cores
  - work distribution
- Memory management in the JAMAICA architecture (1)
- Software architecture:
  - virtualization
  - the Jikes RVM
  - binary translation
  - Java operating systems
- Memory management in the JAMAICA architecture (2)
- Summary





MANCHESTER 1824

The University of Manchester

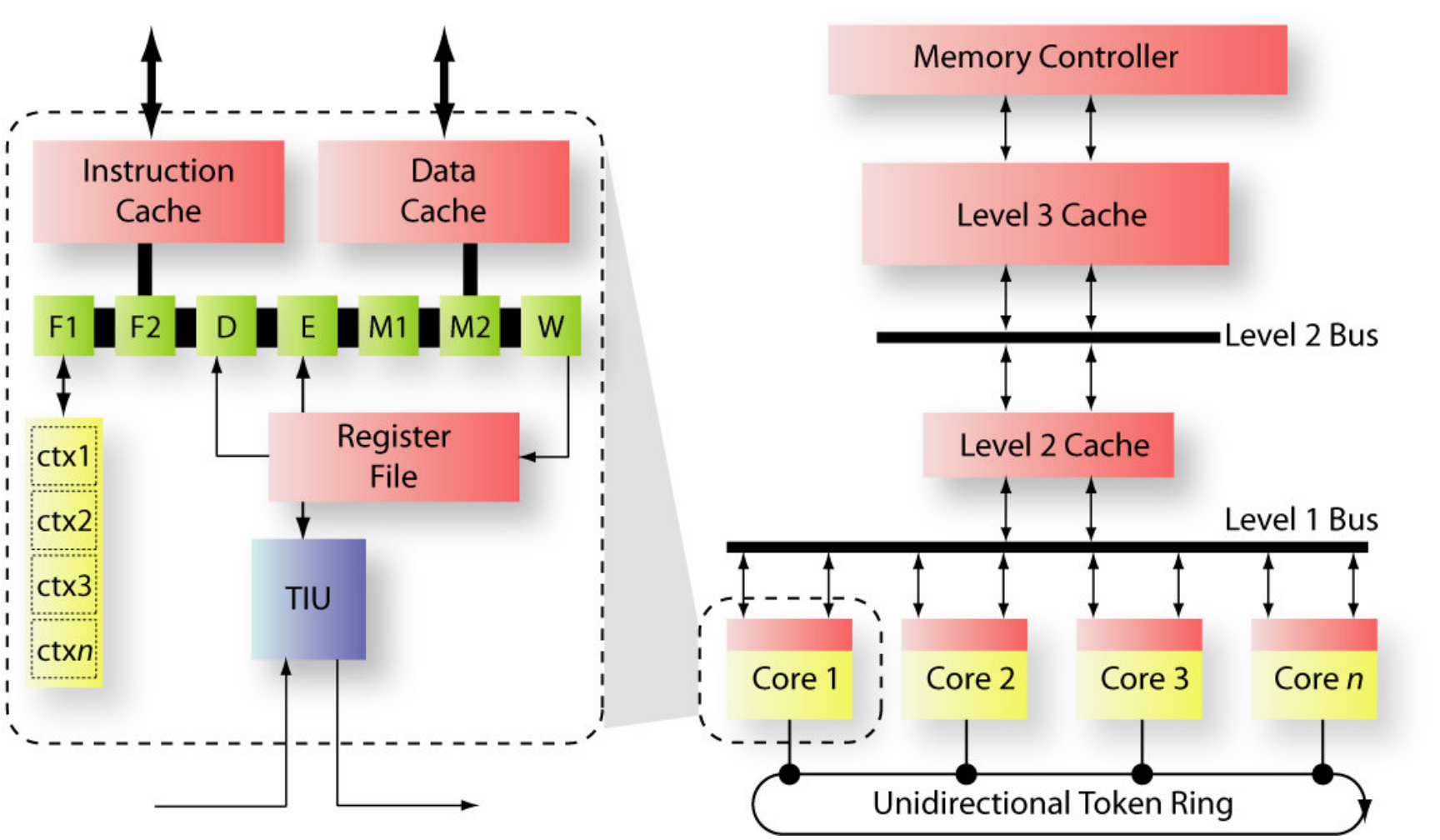
Overview of the JAMAICA architecture

- Java Machine and Integrated Circuit Architecture (JAMAICA)
- Design goals:
  - how to use a billion transistor budget?
  - targeted for Java programs
  - high-level initial simulation, for flexibility and speed
  - cycle accurate simulation to explore architectural decisions
  - extends the work on the VAULT architecture (uniprocessor multiple contexts)
- Features:
  - 2 tier architecture of CPU nodes and groups of nodes
  - split transaction cache coherent bus protocol
  - thread scheduling and work distribution support unit
  - heap allocated register windows
  - context switch on cache miss
  - simulators in both C and Java

MANCHESTER 1824

The University of Manchester

Overview of the JAMAICA architecture



MANCHESTER 1824

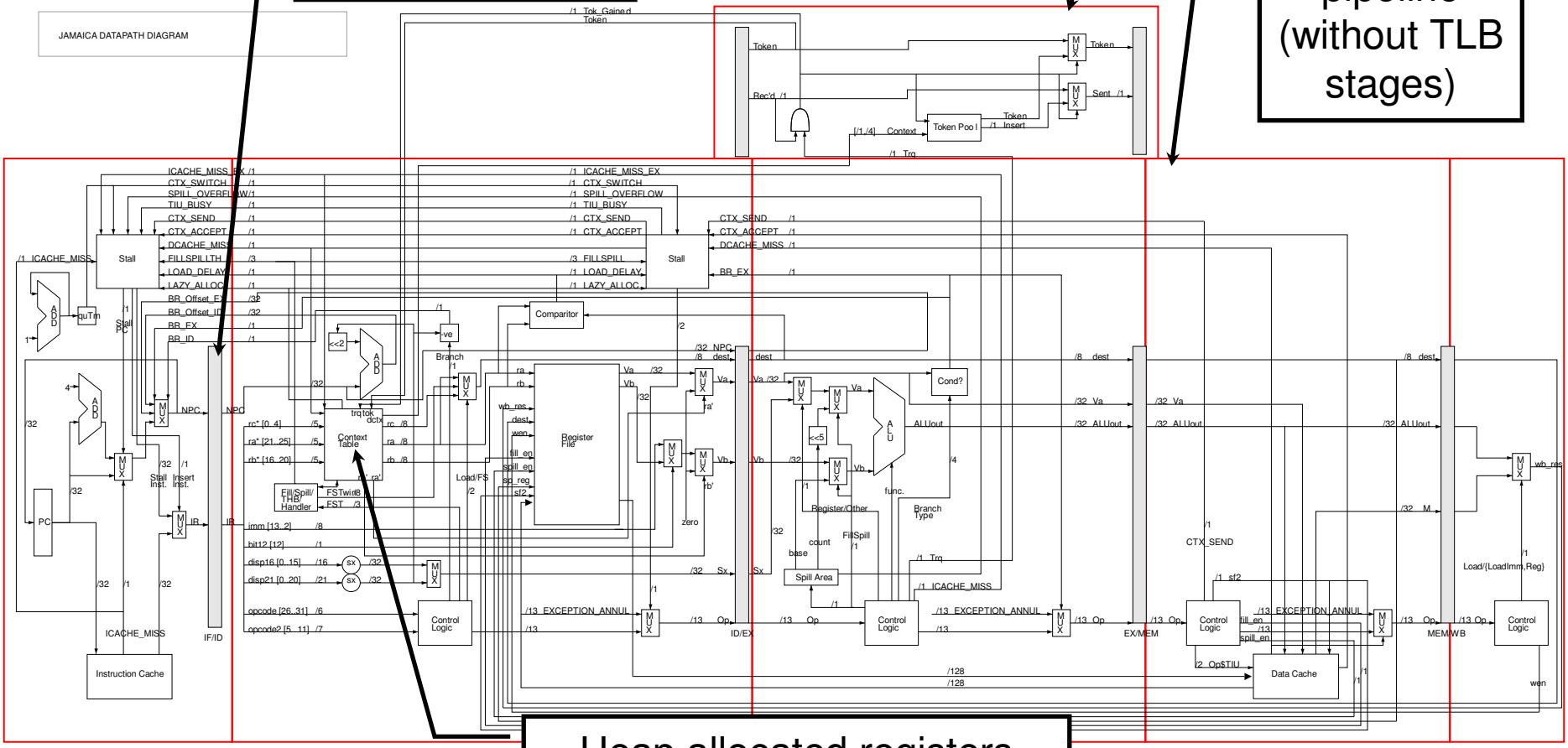
The University of Manchester

Some more detail

Token ring interface

5 stage MIPS based pipeline (without TLB stages)

Alpha based instruction set



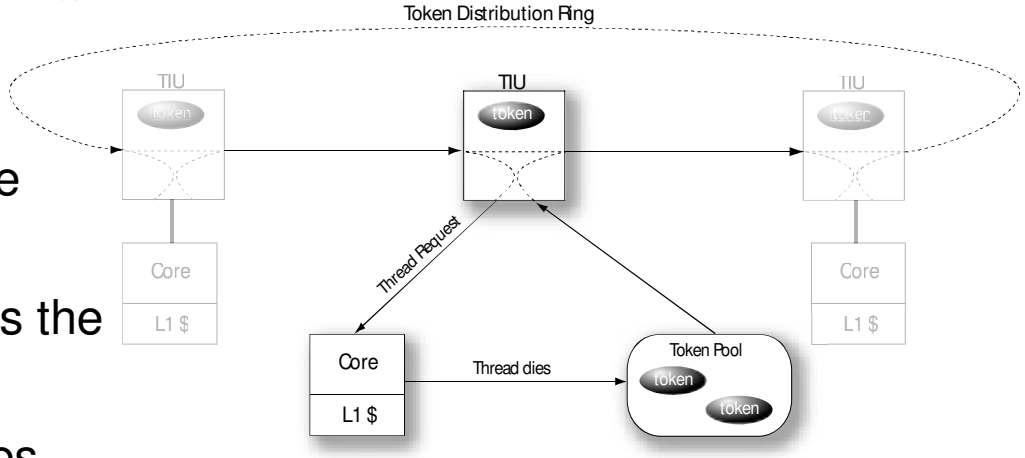
Heap allocated registers and context management

MANCHESTER 1824

The University of Manchester

Work distribution

- Idle threads distribute tokens on a token ring bus
- Executing context on a core requests to ship work to an idle context or core and context
- Taking a token from ring grants the use of a particular context
- Shipping of work between cores occurs over data bus
- Gives lightweight thread creation
- When token is redistributed, work has been completed
- Thread unit monitors for completion of forked work



MANCHESTER 1824

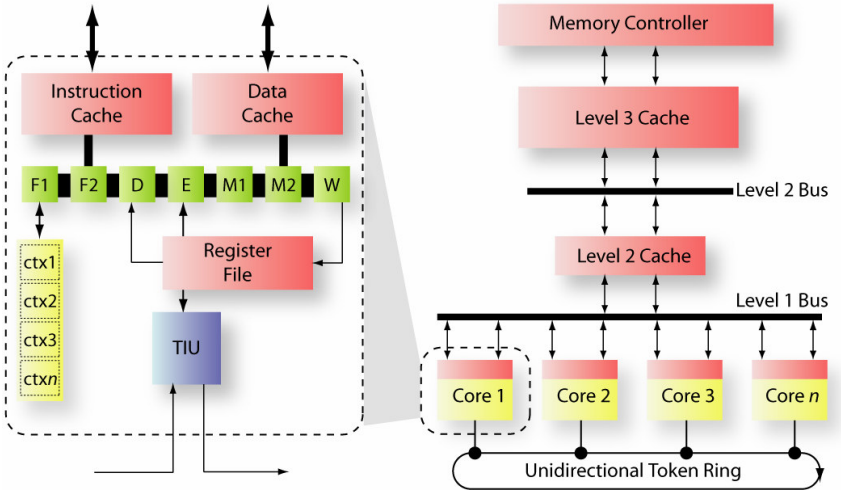
The University of Manchester

Status

- "Universities can't design processors as they are too big, too complex and it requires too many resources"
  - This is in part true, so we focus on "interesting areas"
  - Use of text book or other research results to fill in the blanks
  - Gradually filling in the blanks but in a way that allows us to reconsider our decisions in the simulator
  - Slow development of hardware means compiler and hardware ideas can grow up together

Memory management in the JAMAICA architecture (1)

- Work on buses is going on in parallel to design of the memory system
- Bus design is considering point-to-point asynchronous buses
- Directory based coherency makes sense in this context
- Temporary measure is to have TLB and MMU in each core, coherency achieved by not having virtual caches and a snoopy bus protocol
- However, software is driving some interesting design decisions…



Software support for the JAMAICA architecture

- Static tools
  - C compiler – based on Princeton's LCC
  - jtrans – Java class file to assembler
  - javar – modified to generate jtrans parallel constructs
  - sim-idbg – interactive debugger

MANCHESTER 1824

The University of Manchester

Virtualization

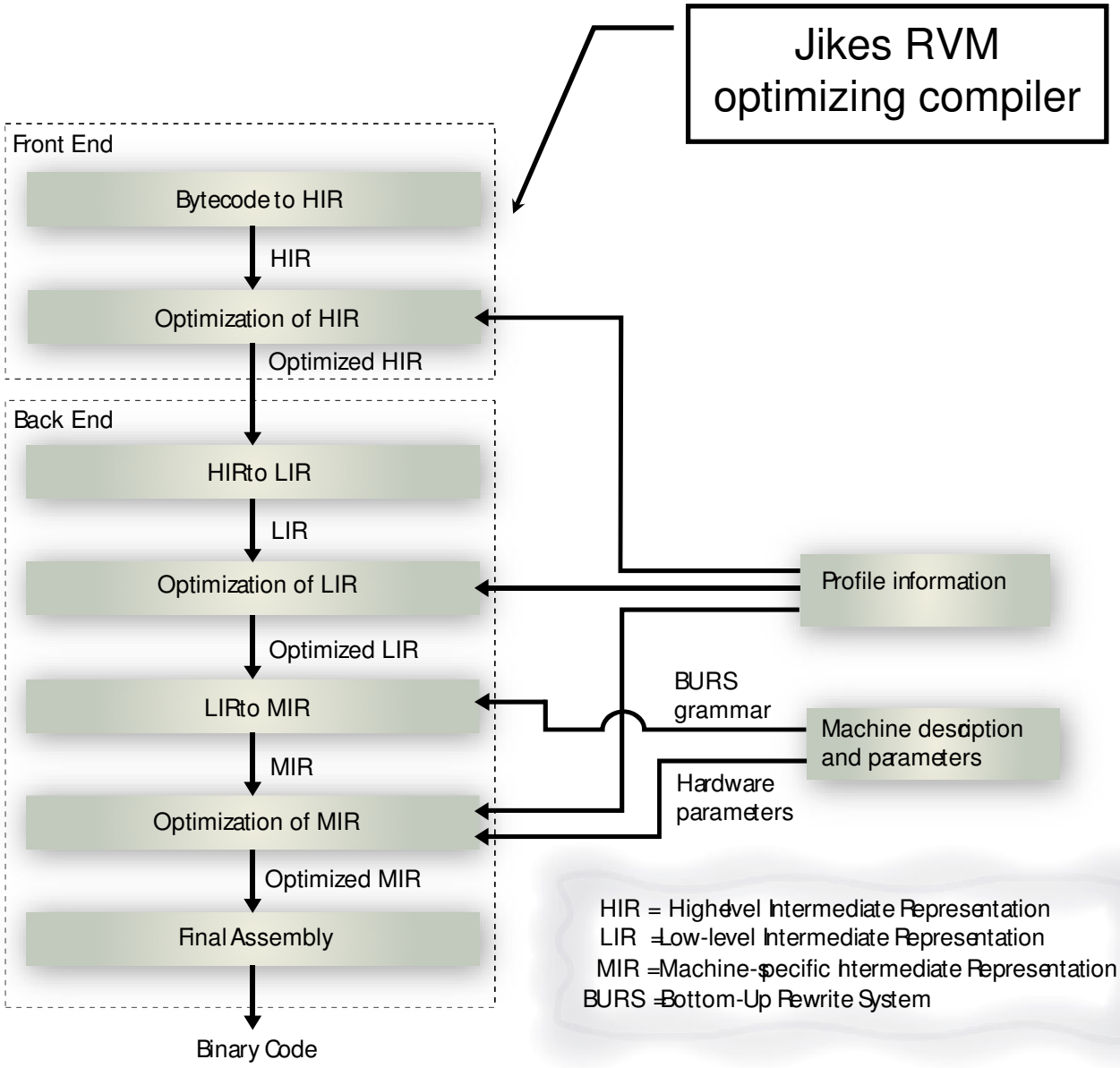
- Platform independence
  - Operating system virtualization
    - Run multiple operating systems simultaneously on virtualized hardware
  - Application virtualization
    - Standard application formats such as ELF can run on a multitude of operating systems as binary format and system call interface are standardized.
    - Wine allows windows applications to run on FreeBSD, Linux and Solaris
  - Instruction set virtualization
    - Dynamic binary translators …
- Hardware flexibility
  - Transmeta - 4-way VLIW TM3000 and TM5000 processors, 8-way VLIW TM8000 processor all run IA32 code
- New compiler optimizations …

MANCHESTER 1824

The University of Manchester

The Jikes RVM

- JVM written in Java
- Support for IA32, PowerPC and JAMAICA
- Baseline (quick) and optimizing compilers
- Adaptive optimization and feedback system
- Extended array SSA form sub-stages in HIR and LIR optimization



MANCHESTER 1824

The University of Manchester

Parallel loop optimisations

- Jikes RVM is good at removing dependencies:
  - extended array SSA form models scalar and heap dependencies, SSA means all dependencies are true dependencies
  - pi nodes used to rename a variable following a comparison, to remove redundant compares
- Loop analysis adds annotated Loop Structure Trees (LSTs) to Jikes RVM IR
- Annotated LSTs describe loop structure to optimisations that:
  - Explicitly test exceptions based on a loop iterator before a loop body
  - Create loop bodies that don't throw exceptions
  - Create thread forking and joining code for do-all amenable loops
- Performance increase of 79% on a dual core setup for a simple kernel
- SpecJVM improved by 3.7% best case, 1.9% on average
- Analysis of break-out paths leads to speculative parallelisation

MANCHESTER 1824

The University of Manchester

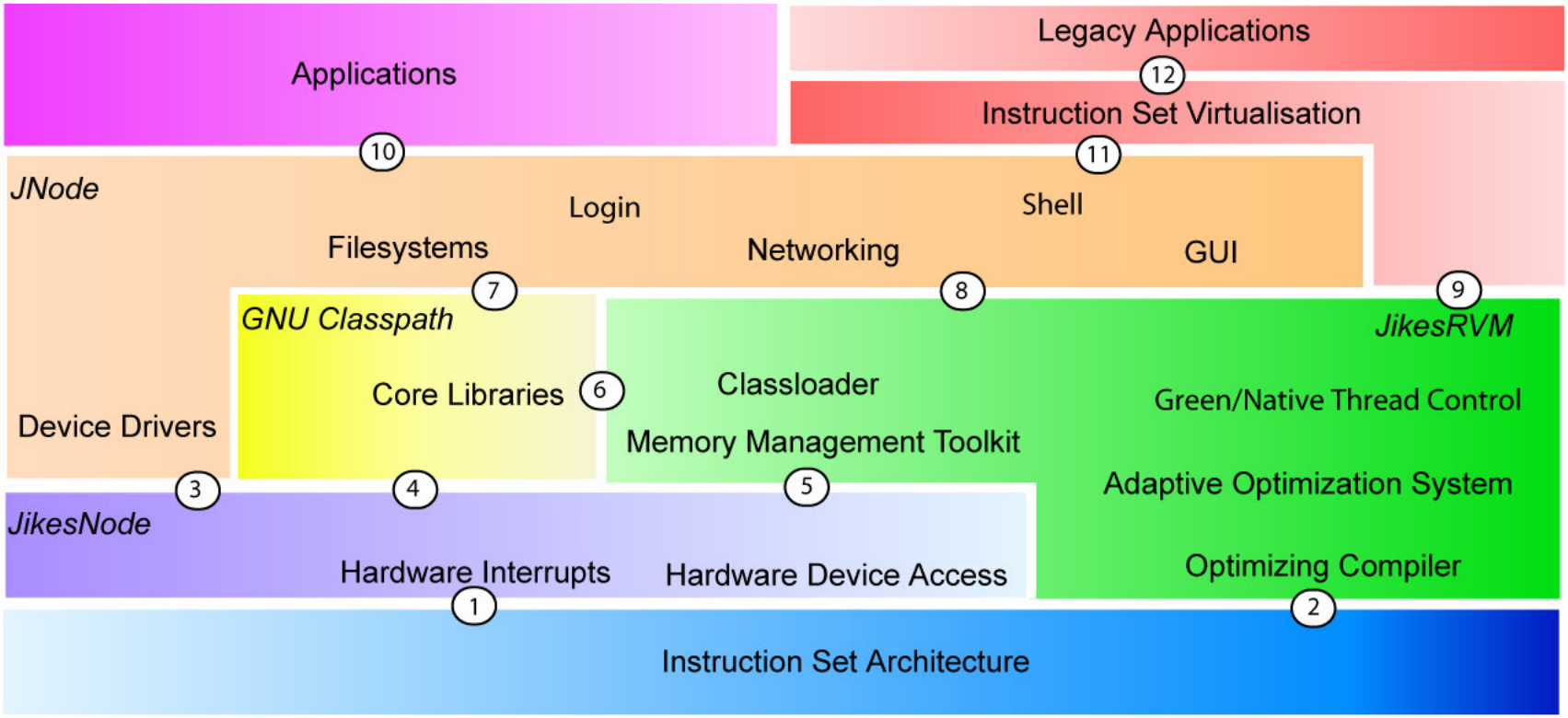
Dynamic Binary Translation

- A technique with mixed fortunes
  - FX!32 – IA32 → Alpha
  - Dynamo – PA-RISC/IA32 → PA-RISC/IA32
  - Rosetta – PowerPC → IA32
- PearColator - PowerPC Dynamic Binary Translator (DBT) written (in Java) to use the Jikes RVM optimising compiler
  - Trace based
    - trace length depends on optimisation level
    - traces start and stop at procedure calls and returns
  - HIR code can be converted to Java bytecode
  - Lazy flag optimisations
  - Adaptive/parallel recompilation
  - Virtual memory system accurately emulated
    - Translator not visible in translated code's memory space
    - *Slow page then value look up*

MANCHESTER 1824

The University of Manchester

Java operating systems



- Java OS –created a version of the Jikes RVM which will boot the JNode class library and device drivers
- Related works include MVM and JX OS

MANCHESTER 1824

The University of Manchester

Speculative execution

- Aim: increase number of parallel threads
- Range of speculative and non-speculative execution states
  - tree rooted at non-speculative state with branches for every spawned speculative context
  - speculative contexts may spawn more speculative contexts
- If speculation goes wrong squash speculative state
  - throw away values in cache or a buffer
- Detect speculation problems:
  - in software: when a value isn't that expected explicitly squash
  - in hardware: when an address is loaded by a speculative context, ensure that stores to the same address from a less speculative context cause a squash
- Problems with creating speculative threads and avoiding excessive squashing
- Mechanism may aid virtual machines, e.g. handling of unaligned memory accesses

MANCHESTER 1824

The University of Manchester

Memory management in the JAMAICA architecture (2): support for virtualization

- All software is part of the same software system
  - optimizations including parallelization between different running parts of the system
- No concept of processes at architectural level
  - but concept of light-weight threads
- Data and instruction spaces can be separated
  - detection of self-modifying code an issue
- Instruction and data spaces identified by an address space identifier (ASID) and is akin to conventional use of ASID to stop flushing the TLB on a context miss (although I$ and D$ have potentially different ASIDs)
- Separate address spaces enables binary translator to pass through load and store operations and remove PearColator performance bottle neck

MANCHESTER 1824

The University of Manchester

Summary

- Silicon density has already brought about CMP systems
- The JAMAICA project is refining a CMP system that relies on software virtual machine support
- Simpler hardware, due to software support, can increase flexibility and the amount of available parallel resources
- Knowledge of the hardware only known by the virtual machine
  - portable infrastructure
  - virtualised CPU system
- Hardware and software assist to extract more, possibly speculative, parallelism and to improve scheduling
- Operating and virtualization system amenable to runtime optimization
- Operating system and virtualization system prototyped on IA32 architecture – some of which is already open source

MANCHESTER 1824

The University of Manchester

Thanks!

- … and any questions?