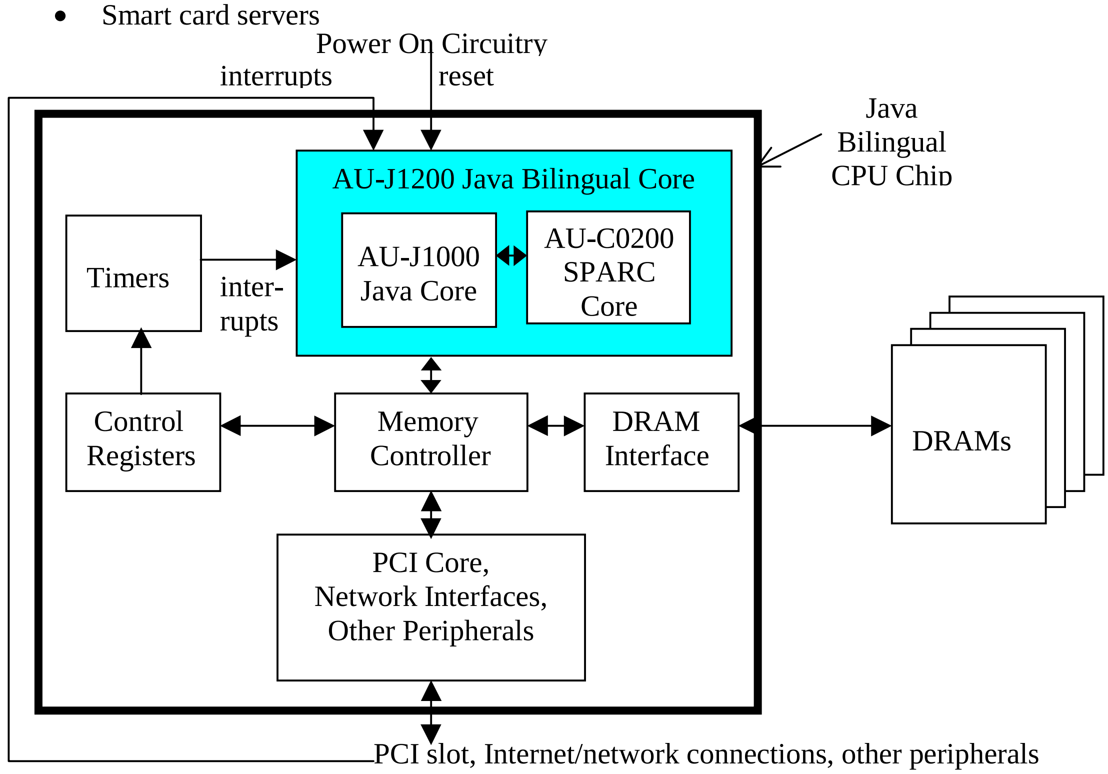
AU-J1200: High Performance Java Bilingual Processor Core

The AU-J1200 Java Bilingual Processor Core is a Java plus 32 bit SPARC processor core targeted at efficient high performance Java execution. It directly executes both Java code and 32 bit SPARC code, thus providing software compatibility for existing applications, as well as fast, efficient Java execution. New applications can be written in Java or traditional languages, and compiled into either Java bytecodes or SPARC binaries, as is best suited for the situation. Java performance is significantly faster than other software *and* hardware Java solutions due to several proprietary mechanisms that boost Java performance. In addition to fast Java execution, uncompromised legacy performance is achieved with Aurora VLSI's proprietary architecture. Performance on legacy SPARC code is comparable or better than that of other single scalar legacy processor cores at the same frequency and price points. See the white papers at www.auroravlsi.com for a discussion of the software advantages of this bilingual processor. The AU-J1200 Java Bilingual Processor Core is available as a synthesizable Verilog model from Aurora VLSI, Inc. Contact CustomerService@auroravlsi.com.

The AU-J1200 Java core is intended for a wide variety of high end Java applications. It can be integrated into numerous types of chips that run Java and legacy code in:

- Internet appliances
- Home gateways and servers
- Set top boxes, DTVs, personal TVs
- Game stations
- Smart card servers



SPARC is a trademark of SPARC International, Inc.

The Java Bilingual Core is a modular design consisting of the AU-J1000 Java Processor Core, the AU-C0200 32 bit SPARC Processor Core, the C02XX Memory Management Unit (MMU), interface logic connecting the Java Processor Core and SPARC Processor Core, and power management logic. The instruction and data caches (Cache Unit) are part of the Java Processor Core, and are accessible by both the Java Processor Core and SPARC Processor Core. From the perspective of the SPARC Processor Core and operating system that runs on it, the Java Processor Core is a coprocessor.

The Java Bilingual Core has two execution modes- Java mode and legacy mode. At any given time, it is in one of these two modes as determined by a software controlled mode bit. In Java mode, Java bytecodes are executed, all instruction fetches and data requests are from the Java Processor Core, and the Java Processor Core fields interrupts and other exceptions. In legacy mode, SPARC instructions are executed, memory requests are from the SPARC Processor Core, and the SPARC Processor Core detects exceptions and interrupts. In legacy mode, the Java hardware is in a low power state, and in Java mode the SPARC processor hardware is powered down.

The Java Bilingual Processor Core interfaces to external logic through two high bandwidth memory ports and an interrupt interface. Both the instruction fetches and data accesses each have 64 bit interfaces to external memory logic. At 250 MHz, this provides a bandwidth of 2 Gbytes/sec for both instruction and data transfers to/from the core, for a total bandwidth of 4 Gbytes/sec into the Java Bilingual Processor Core. These high bandwidth memory ports help ensure high performance. The interrupt interface includes sixteen independent interrupt request lines that are always monitored.

A bus interface and memory interface for the AU-J1200 Processor Core is provided by the AU-S1000 Processor Memory and Bus Interface Core or AU-SB1000 Processor Memory and AMBA AHB Bus Interface Core from Aurora VLSI. These Memory and Bus Interface Cores connect seamlessly to the AU-J1200 Processor Core.

AU-J1200 Java Bilingual Processor Core features are summarized:

- Legacy SPARC code + Java processor core
- Acitve mode (Java or legacy execution) selected by software programmable mode bit
- Seamless connection to Processor Memory and Bus Interface Cores from Aurora VLSI that provide a bus interface and memory interface
- Interrupts taken by the active processor

<u>Java Processor Core</u>

- Hardware accelerated Java execution- 20,000/30,000 Embedded Caffeine Marks at 200/350 MHz (predicted)
- Peak execution rate of 4 instructions/cycle (7 bytecodes/cycle)
- Proprietary hardware accelerates common Java specific functions
- Execution of all “The Java Virtual Machine Specification” bytecodes (fourteen bytecodes require software assists)
- Proprietary bytecode execution for the JVM run time system (system functions)
- 5 stage RISC pipeline

SPARC Processor Core

- Smallest possible 32 bit processor core- 20K – 25K gates
- Low power- .15 - .3mW/MHz
- 32 bit SPARC instruction set
- Good performance- 160/280 Dhrystone 2.1 at 200/350MHz (predicted)
- Peak execution rate of 1 instruction/cycle (single scalar design)
- 5 stage RISC pipeline
- Coprocessor port used by the operating system, to manage the Java Core

Cache Unit

- Separate instruction and data cache
- Sizes are configurable from 256 bytes to 8Kbytes each
- Direct mapped
- 16 byte line sizes
- Writeback data cache
- Physically addressed, virtually indexed
- Byte parity
- Direct access to data and tags of each cache for cache management
- Separate instruction and data high bandwidth memory interfaces- 8 bytes/cycle peak

Memory Management Unit (MMU)

- SPARC Reference MMU
- Simultaneous instruction and data virtual address translation for high performance
- 4K byte page size
- TLB
    - 32 or 64 entries (configurable)
    - Dual ported
    - Hardware replacement upon TLB miss

The core is delivered as a synthesizeable RTL Verilog model. Deliverables include:

- RTL Verilog source code model of the core
- Verilog testbench and test cases
- Synthesis scripts examples
- Complete detailed documentation and training class notes

Java Processor Core

The Java Processor Core is the AU-J1000 Java Processor Core. It is a single scalar processor core targeted at efficient, high performance Java execution. Several proprietary techniques that address Java specific architectural features, result in high speed Java execution. The Java Processor Core runs all Java bytecodes defined in “The Java Virtual Machine Specification”. Additionally, proprietary bytecodes are implemented for system functions. Its predicted Embedded Caffeine Mark benchmark (CM3.0) number is 20,000 at 200 MHz and 30,000 at 350 MHz.

The Java Processor Core achieves its high performance with a single scalar architecture that utilizes a five stage pipeline. Integer and floating point operation units are included. This pipeline and these operation units provide resources that allow up to seven bytecodes to execute each cycle.

SPARC is a trademark of SPARC International, Inc.

SPARC Processor Core

The SPARC Processor Core executes 32 bit SPARC code. It is an efficient single scalar 32 bit processor based on a five stage RISC pipeline. Its peak execution rate is one instruction/cycle. Predicted Dhrystone 2.1 performance is 160/280 at 200/350MHz.

Cache Unit

The Cache Unit provides caches for the Java Bilingual Processor Core. It consists of an instruction cache and a data cache. The size of each cache can be separately configured from 256 bytes to 8Kbytes. These caches are direct mapped with line sizes of 16 bytes. The data cache is a writeback cache. The caches are physically addressed and virtually indexed. Cache data and tags are protected by byte parity.

Memory Management Unit (MMU)

The Memory Management Unit (MMU) provides virtual address translation for the Basic Processor Core. 32 bit virtual addresses are translated to 32 bit physical addresses by the MMU. Address translation is done in parallel with instruction and data cache lookup resulting in single cycle cache accesses when a hit occurs in both the cache and MMU. The page size is 4K bytes.

The MMU contains a two port TLB and can therefore, simultaneously translate instruction and data virtual addresses. The TLB size can be configured to 32 or 64 entries. Upon a TLB miss, TLB entries are replaced by hardware.

The MMU has separate instruction and data interfaces to both the Basic Processor Core and Cache Unit. These separate interfaces allow simultaneous translation of instruction and data addresses, resulting in highest performance. The MMU accepts translation requests with their virtual addresses, from the Cache Unit. It then returns the physical address to the Cache Unit, in the next cycle.