64-bit ISA, multi-thread and multi-core extensions of the AJIT processor for high-performance embedded applications

Madhav Desai
Department of Electrical Engineering
IIT Bombay

March 20, 2023

AJIT core family

- From single-core-single-thread to quad-core-eight-thread.
- Variants: 1x1x32_lite, 1x1x32, 1x1x64, 1x2x32, 1x2x64, 2x1x32, 2x2x32, 2x1x64, 2x2x64, 4x1x32, 4x2x32, 4x1x64, 4x2x64.
- AJIT tool-chain
  - GNU bin-utils, GCC, G++.
  - UCLIBC.
  - Hardware access routines.
  - Device access routines: serial, timer, interrupt-controller, SPI, GPIO, I2C.
  - Print, timer routines.
- CORTOS2
  - Cooperative RTOS with support for locks, queues, dynamic memory allocation.
  - Hardware abstraction layer.

Single AJIT execution pipeline (thread)

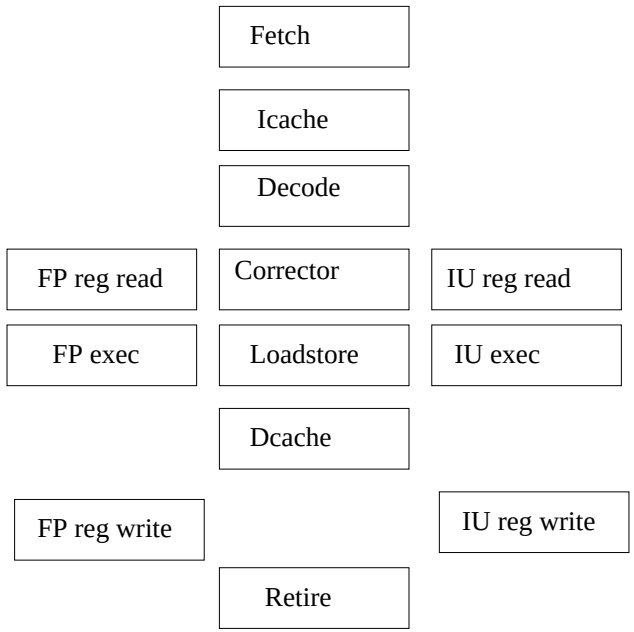

Figure: AJIT pipeline (thread)

Single core single thread AJIT processor core

- Specification and Goals.
  - IEEE 1754 standard ISA (Sparc V8).
  - IEEE floating point unit.
  - VIVT ICACHE (32kB) DCACHE (32kB) with 2-cycle latency on hit, MMU with 256 entry TLB and hardware page table walk.
  - Single issue, in-order, seven stage one instruction-per-cycle instruction pipeline, with branch-predictor, 500K ASIC gates, 75K LUTs on FPGA.
    - FP unit 20K LUTs, MMU 14K LUTs, Caches 8K LUTs.
  - C compiler, debugger, Linux (single thread only).
- Status
  - Entire processor successfully running on FPGA.
  - Compiler, Debugger functional, validated in hardware.
  - Linux port completed, validated on FPGA.
- 32-bit AJIT core successfully used in IITB NAVIC SOC ASIC which has taped out and is functional.

Single core single thread AJIT processor core

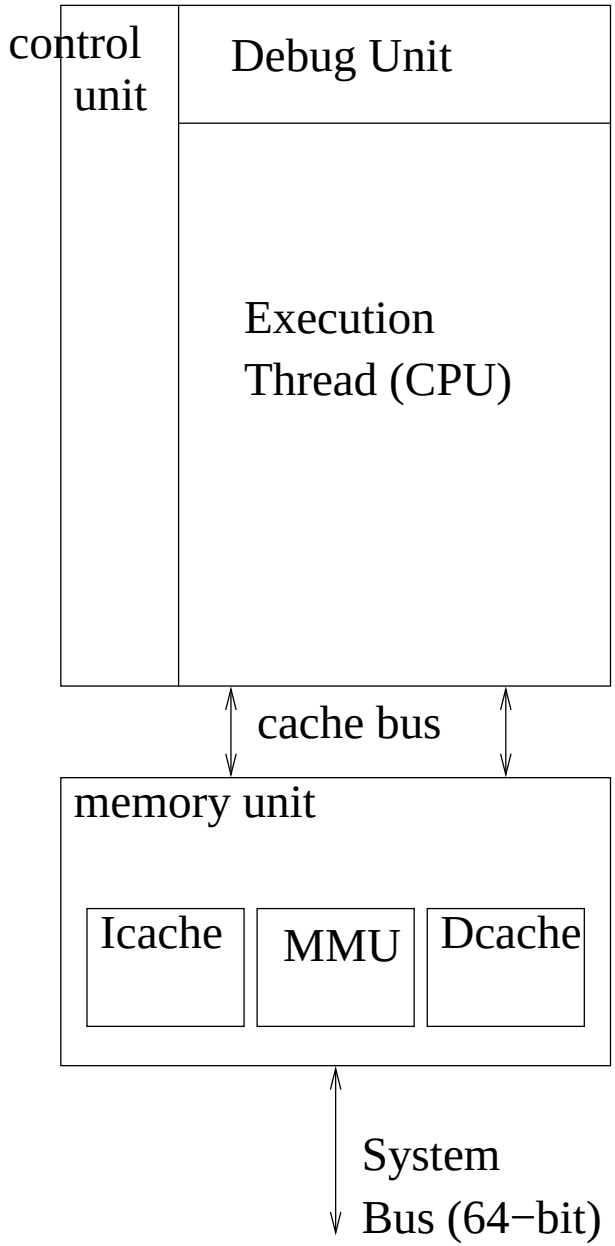

Figure: Single threaded core

Single core single thread AJIT processor lite core

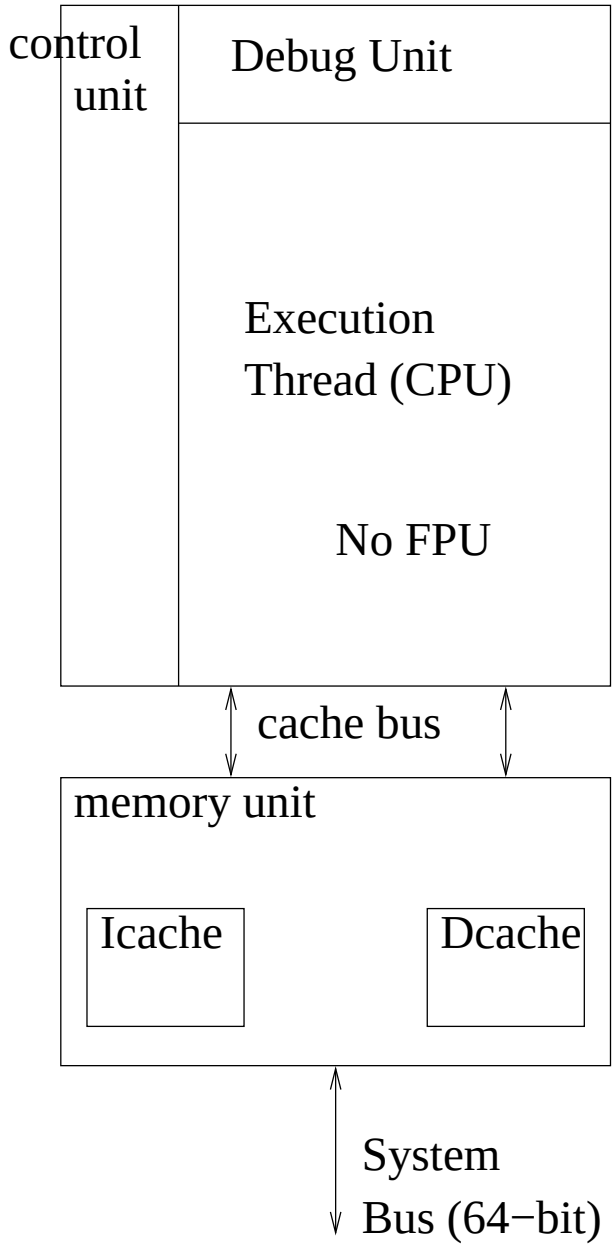

Figure: Single-threaded lite core

64-bit ISA extensions to the CPU thread

- Backward compatible: capable of executing 32-bit as well as 64-bit instructions.
- Instructions will continue to be 32-bits wide.
- Register file: can be viewed as 32x32 or 16x64.
- 64-bit integer unit: arithmetic, logical instructions for 64-bit data.
- SIMD instructions for byte/half-word/word and half-precision/single-precision/double-precision manipulation.
  - Reduce instructions for dot-product. Can perform 8-point 8-bit dot product in two instructions.
- 64-bit FPU (existing, will permit 2xSingle-precision or 4xHalf-precision ops/instruction).
- Virtual address width will be maintained at 32 bits, with physical address as 36 bits.

Impact analysis of ISA enhancements

- The 64-bit CPU needs 8% more hardware than the 32-bit CPU.
- BLAS library: 20% to 75% improvement in level 0 BLAS routines (dot product, matrix product, matrix vector product) on short integer and single precision floating types.
- String library: 40% to 60% improvement on aligned string operations.
- FFT: 29% improvement in single-precision performance.
- AI/ML convolution: 15% to 22% improvement for 4x4 kernel.

Single core two thread AJIT processor core

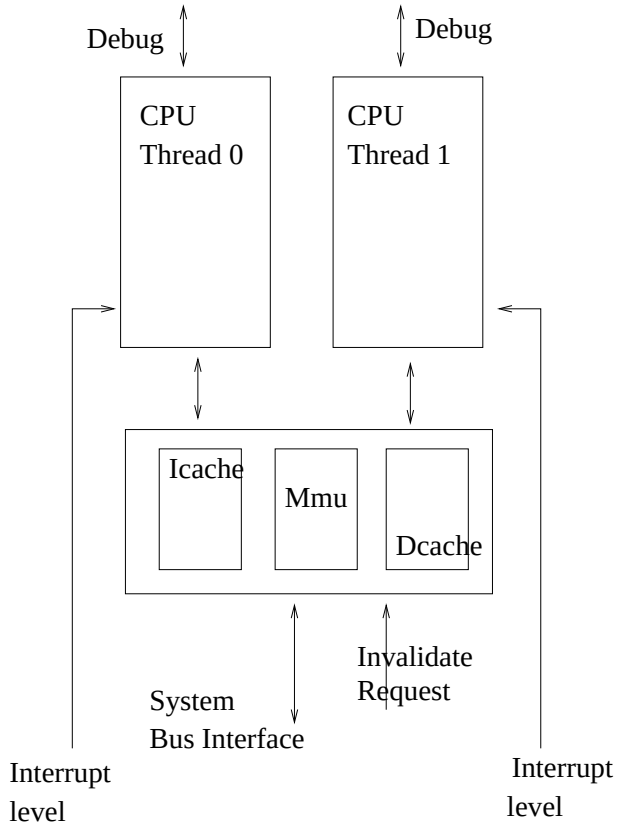

Figure: Dual threaded core

Impact analysis of two threaded core relative to single threaded core

- Two threaded implementation area estimate is 1.33X that of the single-threaded implementation.
- Cache hit latency for the two-threaded implementation is 3 cycles instead of 2 cycles for the single-threaded implementation.
- However, peak memory bandwidth for two-threaded core is equal to the single-threaded core.
- Performance improvement for two-threaded core is 1.55X for core-mark, 1.6X for matrix multiplication.
- Destructive interference between two threads sharing the caches is an issue: for example locking of caches by one thread (atomic instructions) can affect the other thread.
- We are currently investigating a lock-less parallel programming model (the SIDEKICK model) to exploit constructive interference between the two threads in the core.

Two threaded 1x2x32 AJIT processor relative to single threaded 1x1x32

	1x1	1x2	Speedup
Coremark	1.81	2.8	1.55X
Dhrystone	2.37	3.46	1.46X
Whetstone	0.48	-	-

Iterations per MHz are reported above. The GCC compiler version sparc-linux-gcc 4.7.4 was used. The observed IPC is approximately 0.8 for 1x1 implementations.

Multi-core implementation

- Invalidate mechanism inside core caches.
- Coherent memory controller.
  - Snoop invalidate using write-through property of core caches, with novel snoop filter mechanism.
  - Provably deadlock-free scheme.
- Programmable interrupt controller.
  - Assign 15 primary interrupt sources to any of the available hardware threads.
  - Complete flexibility to real-time embedded system designer.
- L2 cache (8-way set associative write-back cache, up to 1MB).

4x2x64 multi-core

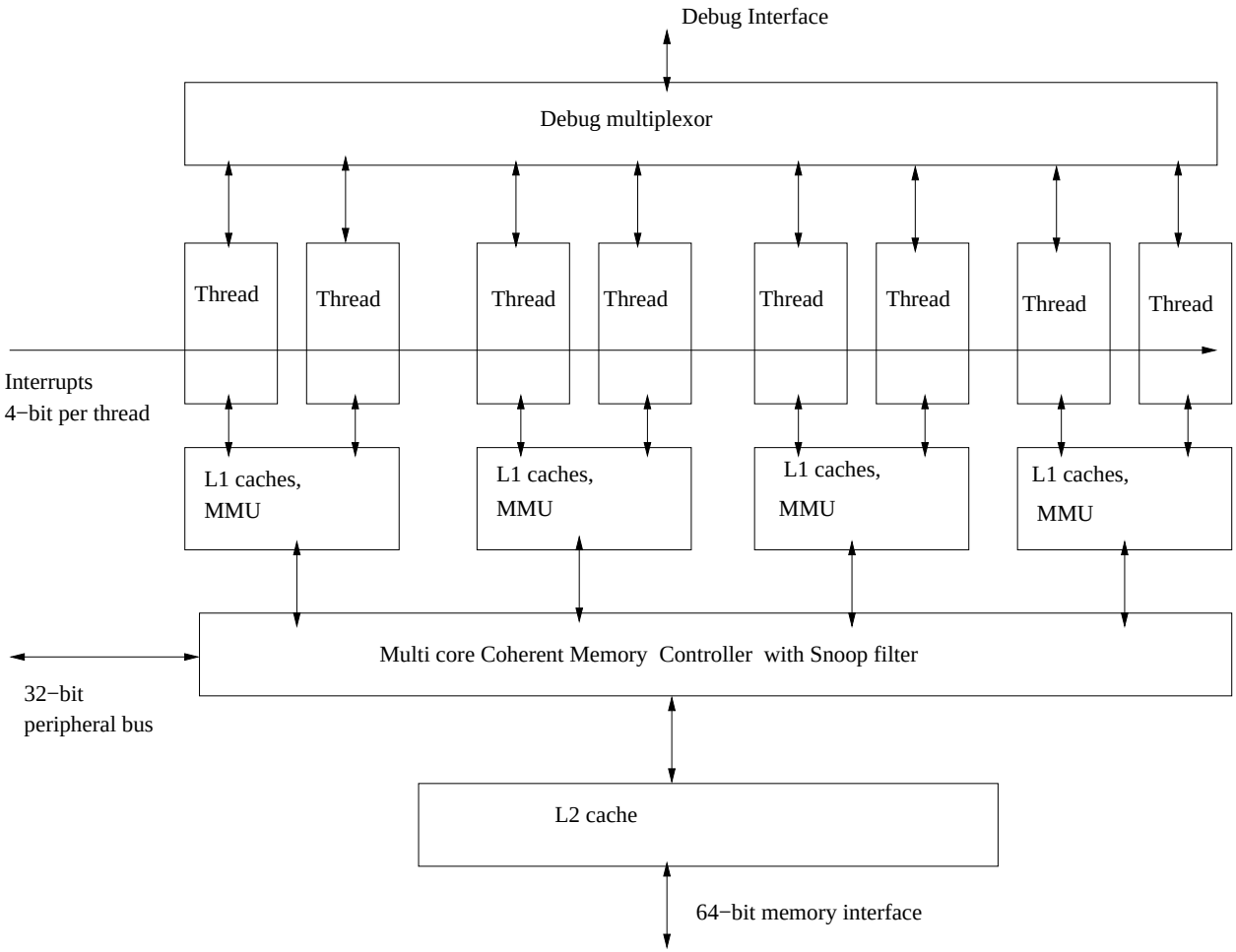

Figure: Final multi-core

Other building blocks developed as part of this project

- Native AJIT bus protocols: AJIT Core Bus (ACB) 64-bit, AJIT Fifo Bus (AFB) 32-bit
  - Bridges to AXI, AHB, APB busses.
- Bus multiplexors, Bus demultiplexors.
- Peripherals: UART, I2C, SPI, SPI-Flash, SRAM-controller, Dual-port SRAM controller.

Custom configurations can be assembled quickly.

Evolution path

- Heterogenous multi-core.
- Multi-channel, wider main memory interface.
- Clusters of multi-cores
  - Locally coherent, globally incoherent.
- Applications!

AJIT tool-chain

- ssh://git@github.com/adhuliya/ajit-toolchain
- GNU bin-utils, GCC, G++, UCLIBC.
- Hardware access routines.
- Device access routines: serial, timer, interrupt controller, SPI, I2C, GPIO.
- ISR, generic trap handling.

AJIT processor lite FPGA prototyping platform

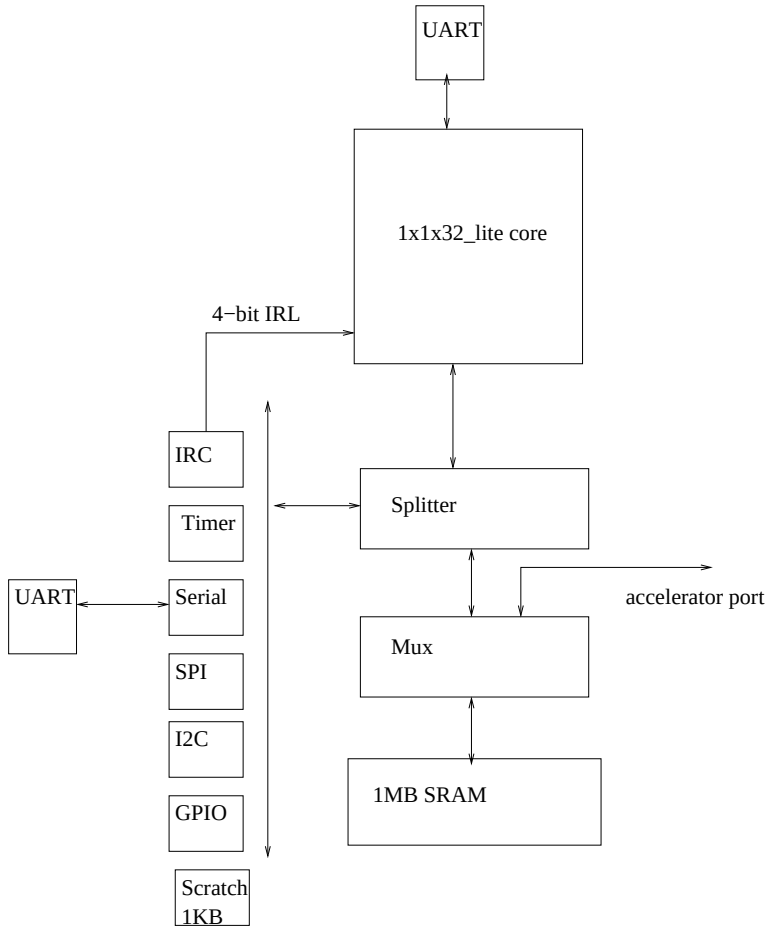

Figure: AJIT processor lite FPGA prototyping platform (KC705)

AJIT processor lite FPGA prototyping platform

- 32KB I, D caches.
- 49K LUTs (for everything).
- Accelerator memory interface (ACB bus protocol).
- 1 self calibrating UART for debug interface.
- 1 programmable UART for serial interface.
- SPI, I2C, GPIO, Scratch-pad, Timer, Interrupt Controller.

Demo

- Describe system to CORTOS2.
- Write program.
- Compile and build.
- Check on processor platform simulator.
- Download to hardware and run.
- Connect to debugger.