ARM1026EJ-S™
Synthesizable ARM10E™ Family Processor Core

Eric Schorn
CPU Product Manager
ARM Austin Design Center

Five Families of ARM Processor IP

ARM preserves SW & HW investment through code and process portability

ARM10E™
- Multiple S/TCM - Float in HW
- Dual 64bit I/O - 6 Stage Pipe
- 2 product offerings
- New: ARM1026EJ-S

ARM9E™
- Config S/TCM - Soft-IP Design
- DSP Instr. - Java™ in HW
- 3 product offerings

ARM9™
- Dual Caches - Hard-IP Design
- Performance - 5 Stage Pipe
- 3 product offerings

ARM7™
- Low Power - Area Efficient
- Code Density - 3 Stage Pipe
- 4 product offerings

SecurCore™
- Secure Apps - Performance
- Power, Area - Code Density
- 4 product offerings

Future
- See "New ARM Microarchitecture Implementing the ARM6 Architecture"
EPF presentation

This presentation
ARM1026EJ-S Objectives

- Significant new functionality and flexibility enhancements to the ARM10E family of processor cores
- Ideal extension of ARM9E based technology
- Microarchitecture performance optimizations based on detailed benchmark analysis
- EEMBC®, Dhrystone, OS Boot, key apps
- Maximize area efficiency through process-specific compiled SRAM memory arrays
  - Configurable cache and TCM sizes on a case-by-case basis
- Soft-IP design delivery, with hard-IP available
  - Rapid process migration and design flow compatibility

ARM1026EJ-S Overview

- Jazelle™ technology-enhanced ARM10E family processor for platform and embedded applications
- Convergence of proven ARM1020E™ and ARM926EJ-S™ technology
- ARMv5TEJ architecture: ARM, Thumb®, DSP, and Java™ instructions
- MMU and MPU support
- Configurable caches and TCMs
- Extensive 64-bit internal bussing
- Dual 64-bit AHB I/O interfaces
- IEEE 754 FP support with VFP10™
- Real-time trace with ETM10RV™

* According to strict interpretation of Dhrystone v2.1 rules

- 266-325 MHz worst-case on 0.13 μ
- 325-400+ Dhrystone 2.1 MIPS*
New Features in ARM1026EJ-S

- **Jazelle technology**
  - Industry leading Java bytecode execution in HW
  - Compatible with ARM7EJ-S™, ARM926EJ-S, and future cores

- **MMU and MPU support via shared logic**
  - Enables both platform OS and RTOS solutions on the same part
  - Windows® CE, Symbian OS™, VxWorks®, Linux, …

- **Direct-attach vector interrupt controller (VIC) port**
  - Supports more interrupt sources and vectors
  - Significant reduction in interrupt handling latency
  - Solution is compatible with future cores

ARM1026EJ-S in Context

- **Industry-leading feature set**

<table>
<thead>
<tr>
<th>Feature</th>
<th>ARM1026EJ-S</th>
<th>ARM922T™</th>
<th>ARM926EJ-S</th>
<th>Intel® XScale™</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM Architecture</td>
<td>v5TEJ</td>
<td>v5TE</td>
<td>v5TEJ</td>
<td>v5TE</td>
</tr>
<tr>
<td>Core pipeline depth</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>7, 8</td>
</tr>
<tr>
<td>Branch prediction</td>
<td>Static</td>
<td>No</td>
<td>No</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Memory management</td>
<td>MMU &amp; MPU</td>
<td>MMU</td>
<td>MMU</td>
<td>MMU</td>
</tr>
<tr>
<td>Real-time trace</td>
<td>Yes (ETM10RV)</td>
<td>Yes (ETM9)</td>
<td>Yes (ETM9)</td>
<td></td>
</tr>
<tr>
<td>Java bytecodes in HW</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>IEEE Floating-point in HW</td>
<td>Yes (VFP10)</td>
<td>No</td>
<td>Yes (VFP9)</td>
<td></td>
</tr>
<tr>
<td>CPU Cache/TCM</td>
<td>Config Caches/TCMs</td>
<td>16k/16k Caches</td>
<td>Config Caches/TCMs</td>
<td>32k/32k Caches</td>
</tr>
<tr>
<td>Vector interrupt support</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Design delivery</td>
<td>Portable Soft-IP</td>
<td>Portable Hard-IP</td>
<td>Portable Soft-IP</td>
<td>ASSP</td>
</tr>
<tr>
<td>CPU core I/O width(s)</td>
<td>2 x 64 AHB</td>
<td>1 x 32 AHB</td>
<td>2 x 32 AHB</td>
<td>1 x 32</td>
</tr>
</tbody>
</table>

- **Industry-leading support infrastructure**
Squeezing MIPS from a Dhrystone

- Dhrystone has a long history with well defined rules
  - See http://www.arm.com for links to discussion, references, and repeatable code examples

- Yet, confusion appears prevalent even today
  - ARM1020E is 1.23 MIPS/MHz according to the rules of v2.1
  - Auto-inlining improves the score to 1.45 MIPS/MHz
  - File merging improves the score further to 1.71 MIPS/MHz
  - Version 1.1 on an unreleased compiler improves the score further to 1.92 MIPS/MHz

- Better benchmarks are needed for accurate comparisons
  - Ideally would also be more relevant to the real world
  - Reliable, consistent, repeatable …
ARM1020E Certified EEMBC Benchmarks

- The ARM1020E processor was the starting point for ARM1026EJ-S definition and RTL development

- Completed EEMBC certification of ARM1020E with Verilog RTL
  *with the exception of ConsumerMark on cycle model due to simulation time

- Base measurements are ECL certified and publicly available from EEMBC

- Developed a highly-accurate cycle model of ARM1020E in C
  - Verified against RTL using Dhrystone and EEMBC
  - Enables what-if analysis for ARM1026EJ-S

<table>
<thead>
<tr>
<th>Vendor</th>
<th>ARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part #</td>
<td>ARM1020E</td>
</tr>
<tr>
<td>ISA</td>
<td>ARM v5TE</td>
</tr>
<tr>
<td>L2 Cache Size</td>
<td>1M byte</td>
</tr>
<tr>
<td>Instruction Issue</td>
<td>144 byte</td>
</tr>
<tr>
<td>Tool Chain</td>
<td>ARM ADS 1.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>EEMBC</td>
<td></td>
</tr>
<tr>
<td>NetMark</td>
<td>1.62</td>
</tr>
<tr>
<td>AutoMark</td>
<td>120.5</td>
</tr>
<tr>
<td>ConsumerMark</td>
<td>3.9</td>
</tr>
<tr>
<td>OAMark</td>
<td>206.2</td>
</tr>
<tr>
<td>TeleMark</td>
<td>5.42</td>
</tr>
<tr>
<td>NetMark</td>
<td>4.14</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ARM1026EJ-S Branch Prediction

- ARM1020E branch predictor inspects only single entry of prefetch queue
  - Predictor can miss a branch since code alignment, CPU stalls, and branch sequences can affect queue fills

- ARM1026EJ-S branch predictor operates on two queue entries
  - No branches missed by predictor
  - 11% improvement on NetMark
  - 3% improvement on Dhrystone 2.1

<table>
<thead>
<tr>
<th>Entry A</th>
<th>Entry B</th>
<th>Entry C</th>
<th>Entry D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Observations
  - Dramatic (~40%) benefit on NetMark relative to ARM926EJ-S
  - Branch prediction on entire Telecom suite is over 82% correct
  - Static predictor is now over 70% accurate on average
ARM1026EJ-S Memory Subsystem

- Dhrystone loops don’t cause linefills…but real code does
- Optimized BIU to further increase BW of 2x64bit AHB I/O
  - Most improvement seen on larger NetMark packet flows
  - 38% improvement on the first iteration of Dhrystone
  - Tremendous improvement in OS boot cycle count over ARM9
- Optimized write buffer for sequential stores
  - Collapse sequential stores into AHB burst transaction
  - Biggest change in any single component – 68% in Consumer:CMY
- Observations
  - All Net/Auto/Tele code fits in the ICache – see code size stats
  - Data cache hit rate for Net/Auto/Tele upwards of 95%

ARM1026EJ-S Return Stack

- R14 is our link register to hold return addresses
- New feature to predict MOV PC, R14 and BX R14 branch addresses – common return from subroutine
  - Longer ARM10E pipeline allows this logic to fit nicely
- Observations
  - Improvement varied wildly across algorithms
  - Automotive PWM shows over 10% improvement
  - Networking Open Shortest Path First shows 0% improvement
  - Various stack depths were investigated … returns diminish quickly as complexity increases
ARM1026EJ-S MHz Optimization

- Migrating from Hard-IP to Soft-IP is non-trivial
  - Objective is to maintain both MHz and IPC
  - Previous IPC enhancements helped to open up headroom

- Extensive timing analysis was done to determine what features of the design were penalizing speed
  - Objective is to analyze total performance tradeoff, on a feature by feature basis, as measured by EEMBC, Dhrystone, and OS boots

- Several places to tradeoff against the uncommon case
  - Unpredicted and mispredicted branches … mostly hidden by improved branch prediction and return stack
  - Data to LS address forwarding … 2-5% impact
  - Byte/half-word load forwarding … 8% impact on TeleMark only

ARM1026EJ-S Design Summary

- ARM1026EJ-S builds upon ARM1020E, by adding:
  - Soft-IP option with configurable caches and TCMs
  - Jazelle, VIC port, and both MMU/MPU support
  - Improves both performance and area efficiency
  - 7% improvement on geometric mean of EEMBC suites
  - 5% improvement on Dhrystone v2.1

- ARM1026EJ-S is the ideal extension of ARM9E family technology
  - Additional pipeline stage to increase frequency
  - Branch prediction, branch folding, and return stack to improve CPI
  - Extensive 64-bit internal bussing and dual 64-bit AHB I/O for increased bandwidth
ARM1026EJ-S System Overview

ARM1026EJ-S System Solution

- VFP10 FP Coprocessor
- L2 SRAM Arrays
- ARM1026EJ-S CPU
- ETM10RV Trace Macrocell
- L2 Cache/TCM Controller
- AMBA® AHB System Busses

ARM1026EJ-S Integer CPU
- ARM, Thumb, DSP, and Java
- Configurable caches and TCMs
- Dual 64-bit AHB I/O
- 400+ Dhrystone 2.1 MIPS

VFP10 FP Coprocessor
- IEEE 754 SP/DP, vector support

ETM10RV Embedded Trace Macro
- Real-time, zero perf overhead
**ARM1026EJ-S System Solution**

- Optional L2 cache/TCM controller
  - Multiple AHB ports in multiple clock domains up to CPU speed
  - Configurable SRAM array sizes and latencies, set associativity, line lengths, write back/through modes, and line allocation policy
  - Flexible cache and TCM combinations are supported
  - Allows optimization for both performance and cost

- Vector Interrupt Controller
  - Applies prioritized interrupt and associated vector to CPU
  - Addresses significant latency overhead in systems with a large number of frequent interrupts

**Conclusion**

- ARM1026EJ-S is a large step along the ARM roadmap
  - Further extending flexibility, functionality, and compatibility
  - Performance tuning that reflects real-world applications

- Benchmarking remains a difficult and subjective art
  - More and more data enables intelligent judgments
  - Not all important features and effects make it to the bottom line
  - Final performance is the sum of a lot of small improvements

- ARM1026EJ-S demonstrates ARM’s continued focus on the entire solution
  - CPU performance AND flexibility, compatibility, functionality, area, power, integration, software, tools, support …
  - Minimizing total development and production cost