ARM Introduction

- IP licensing company
  - R&D outsourcing, supplying all major semiconductor companies
  - Processor “brain” in the chip
- Started in 1990
  - Based in Cambridge, UK
  - Listed on London and NASDAQ
  - $8 Bn market cap (4x in two years)
- Now 1900 people
  - Mainly R&D engineers
- $600m revenue, 40% operating profit
- Partnership business model
  - ~6 Bn shipments in 2010

ARM started in a barn

Now 30 offices in 15 countries
## How many ARM’s Do You Have?

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile phones</td>
<td>~100% market share</td>
</tr>
<tr>
<td>Smartphones</td>
<td>3x 100% market share</td>
</tr>
<tr>
<td>Mobile Computers</td>
<td>5x 100% market share</td>
</tr>
<tr>
<td>Digital TVs</td>
<td>30% market share</td>
</tr>
<tr>
<td>Disk Drives</td>
<td>~70% market share</td>
</tr>
<tr>
<td>PC Peripherals</td>
<td>30% market share</td>
</tr>
<tr>
<td>Cars</td>
<td>5x 40% market share</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>10% market share</td>
</tr>
</tbody>
</table>
Huge Opportunity For ARM Technology

20+ billion cores to date

100+ billion cores accumulated after next 10 yrs

1998  2010  2020
ARM’s Opportunity at all Price Points

Average Selling Price of a Semiconductor Chip

- $1-2
- $3-6
- $10-15
- >$25

Volume

Opportunity

ARM Usage Today
ARM Connected Community – 700+

Software, Training and Consortia Partners

Silicon Partners

Design Support Partners
ARM Austin

- Austin site opened in 1999
- Currently 250 engrs
- Growing 10%+ per year

- Top right of CPU roadmap
- Interconnect fabric
- Verification tools
- R&D
- Sales, AEs, Support…
Austin is the center of the CPU world

- **CPU Teams**
  1. ARM - High end of Mobile
  2. Qualcomm - DSPs
  3. Intel - Atom
  4. Freescale - PowerPC and more
  5. IBM - Servers
  6. Oracle - Servers
  7. Centaur - Low cost X86
  8. Broadcom - Networking processors
  9. AMD - Multiple CPUs
  10. Samsung - ARM CPUs
  11. Apple - shh… it’s Apple

- **SoC teams**
  1. Calxeda - ARM servers
  2. Nvidia
  3. TI
  4. Cirrus
  5. Plus another 5-10
WHAT IS CORTEX-A15?
Cortex-A15: Next Generation Leadership

Cortex-A class multi-processor
- 40bit physical addressing (1TB)
- Full hardware virtualization
- AMBA 4 system coherency
- ECC and parity protection for all SRAMs

Advanced power management
- Fine-grain pipeline shutdown
- Aggressive L2 power reduction capability
- Fast state save and restore

Significant performance advancement
- Improved single-thread and MP performance

Target Markets
- High-end wireless and smartphone platforms
- Tablet, large-screen mobile and beyond
- Consumer electronics and auto-infotainment
- Hand-held and console gaming
- Networking, server, enterprise applications

Targets 1.5 GHz in 32/28 nm LP process
Targets 2.5 GHz in 32/28 nm G/HP process
Cortex-A15 MPCore Block Diagram

- Global Interrupt Control, Trace and Debug handled across all cores

- 1-4 Processors per cluster
- Each processor has full Out-of-Order (OoO) pipeline.

- Integrated Level 2 cache
Cortex-A15 Pipeline Overview

15-Stage Integer Pipeline

- 4 extra cycles for multiply, load/store
- 2-10 extra cycles for complex media instructions

Fetch
- 5 stages

Decode
- 7 stages

Rename

Dispatch

15-stage Integer pipeline

Issue

NEON/FPU

Int

Branch

Multiply

Load/Store

WB

WB

WB

The Architecture for the Digital World®
# Configuration Challenge

<table>
<thead>
<tr>
<th>System feature</th>
<th>Cortex-A15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>1-4</td>
</tr>
<tr>
<td>L1 cache size</td>
<td>Fixed at 32 KB</td>
</tr>
<tr>
<td>L2 cache controller</td>
<td>Included</td>
</tr>
<tr>
<td>L2 cache size</td>
<td>512KB, 1MB, 2MB, 4 MB</td>
</tr>
<tr>
<td>L2 tag RAM register slice</td>
<td>0, 1</td>
</tr>
<tr>
<td>L2 data RAM register slice</td>
<td>0, 1, 2</td>
</tr>
<tr>
<td>L2 arbitration register slice</td>
<td>0, 1</td>
</tr>
<tr>
<td>Error protection</td>
<td>None, L2 cache only, L1 and L2 cache</td>
</tr>
<tr>
<td>Interrupt controller</td>
<td>Optional</td>
</tr>
<tr>
<td>Number of SPIs</td>
<td>0-224 in steps of 32</td>
</tr>
<tr>
<td>Power management</td>
<td>Optional clamp/power-gate control pins</td>
</tr>
<tr>
<td>Floating point / NEON</td>
<td>None, VFP Only, VFP and NEON</td>
</tr>
<tr>
<td>Trace</td>
<td>PTM (integrated, required)</td>
</tr>
</tbody>
</table>
Cortex-A15 System Scalability

- Processor-to-processor coherency and I/O coherency
- Memory and synchronization barriers
- Virtualization support with distributed virtual memory signaling
VERIFICATION METHODOLOGIES
ARM CPU Verification Strategies

- Design practices – “correct by construction”
- Test planning
- Multiple and varied verification methods emphasizing:
  - Unit level
  - Top level - RIS (random instruction sequences)
  - System level - stress testing
- Coverage
- Soaking / Bug Hunting
Bug Discovery Timeline - Theoretical

- Where are bugs discovered?
Where Are Bugs Found - Actual
Cortex-A15 Unit Level Testbenches
Unit Level Simulation

- Simulation is the corner-stone verification method
- Coverage driven, constrained random SystemVerilog
  - Assertions for interfaces, white box internals
  - Higher level checkers
  - Code and functional coverage drives stimulus completeness
- Well-defined and testplan-linked functional coverage
- Multi-unit testbenches are used where appropriate
- Simulator performance and compute cluster
- Debug visualization and automation
Top Level Testbench

Quad Cortex-A15 MPCore

Processor Coherency (SCU)
Up to 4MB L2 cache

64/128-bit ACP

AXI3 BFM

Clocks, Resets, Power, Interrupts, ECC

AXI3 Xactor

64/128-bit ACE

ACE Snoop Generator

ISS (Arch Model)

Page Maker

Memory Model

Test Programs

Directed
• AVS, DVS
• ISA, MP

Random (RIS)

“Chicken” bits

Trickbox

AXI4 Decoder
Top Level Simulation

- Uses a CPU Top-Level Testbench
  - Simple memory, simple trickbox, Arch reference model integration
- Tests are binary executable programs
- Exercise various Cortex-A15 configurations
- Directed tests
  - AVS is architecture compliance suite every ARM CPU must pass
  - DVS is a suite of directed tests for this ARM implementation
- Random tests (RIS = Random Instruction Sequences)
  - ISA
  - MP/coherency
- Irritators: interrupts, ECC, page tables, “chicken bits”
RIS (Random Instruction Sequences)

- Track record of hitting un-planned scenarios
- Multiple RIS engines have been developed over >12 years and applied to all CPUs
  - 3 mainstream ISA based engines
  - 3 MP targeted engines
  - Plus 5 additional engines to target load/stores, VFP, M and R class cores
- Engines being enhanced to scale in H/W platforms
  - To achieve much higher throughputs (>10^{15} cycles)
RIS Generator Testing Space
System Level Validation

- Objective to perform “in-system” validation of ARM IP
  - Extended validation of IP in system context
  - Find IP product bugs from real-world testing

- Platforms
  - Emulation
    - SystemBench = *configurable* platform for running SV tests
  - FPGA
    - High throughput to enable deep soaking of the design

- Test Content
  - Bare-metal
  - OS-based apps, stress tests
System Validation Platform Example

Coherency
- CCI-400
  - Full cache coherency
  - I/O coherency
  - Prioritization and utilization

Virtualization
- MMU-400
  - OS level virtualization
- GIC-400
  - Virtual interrupts
  - Multicore support

External Memory Subsystem
- DMC-400
  - DDR utilization
  - PHY integration

Rest of SoC Interconnect
- NIC-400
  - Routing efficiency
System-Level: Validation Strategies

**TEST CONFIGURATIONS**
- IP component build configs
  - Multi-core, Neon/VFP engine, cache sizes, interconnect configs, etc
- Systembench topologies
  - Multi-cluster, ACP, DMC, SMC, DMA, etc
- Runtime initialisation
  - Memory regions, performance modes, etc

**TEST PAYLOADS**
- OS and Application compatibility testing
  - Linux, Windows, Android, LTP, benchmarks
  - Hypervisor, TrustZone
- MACK (simplified OS for validation) based stress testing
  - MPRIS – pthead based tests for MP
  - ‘C’ Stress testing library (including coherency tests and targeted stress)
- Bare metal directed/random tests
- RIS
- Runtime traffic irritators (DMA, GPU, VIP)
System level: Emulation/FPGA Farm

- Configurable “System-Level” Testbench
- Emulation achieves ~1MHz
- Effective debug visualisation
- More suitable to longer tests (OS boots, benchmarks, longer RIS sequences)

- Limited fixed configurations
- FPGA achieves 10-40MHz
- Poor debug visualisation
- Targeting RIS testing and stress testing
FPGA Farm

- 21 FPGA platforms per rack
  - V2F-2XV6
    - LX760 & LX550T
    - 4GB DDR2 SODIMM
    - JTAG and Trace
  - V2M-P1 motherboard
    - NOR Flash bootloader
    - Basic peripherals
      - UART for SW debug
      - Ethernet for network boot
      - Video/audio
      - SD/CF for local storage

- Cluster Control
  - Redhat Linux box
  - UART concentrator for debug
  - RVI for software debug
  - FPGA and SW image download
**Dual Cluster Cortex-A15**

- **Solution per VE Platform**
- **Use three V2F-2XV6 boards**
  - 3x LX760
  - 3x LX550T
- **Processor Support**
  - Dual Cluster A15
  - A15 Neon & A7
- **Performance**
  - 10MHz system speed
  - 2-4GB memory space
Formal Property Verification

- ACE proof kit
  - Complete set of bus protocol properties
- Low level assertions
  - Prove assertions on LS unit interfaces
- High level properties
  - L2 ECC proof
  - L2 arbitration register slice
Verification Methodology Summary

**SPECIFICATION/PLANNING**
- Test planning
- SV requirements

**IMPLEMENTATION**
- Unit TB implementation
- Top TB implementation
- System TB integration & bringup
- Baremetal directed test & debug
- OS Matrix testing

**TESTING/DEBUG**
- UTB test & debug / bug extensions
- AVS/DVS test&debug / bug extensions
- RIS test&debug

**COVERAGE CLOSURE**
- Coverage Implementation->Closure
- System TB integration & bringup
- Baremetal directed test & debug

**SOAK/STRESS TESTING**
- UTB Errata extensions
- DVS Errata extensions
- RIS Errata extensions
- SV Errata extensions

**Testing Environments**
- Simulation (200Hz)
- Emulation (1MHz)
- FPGA (10MHz)
- Si (1GHz)
LESSONS LEARNED
Planning

- Take a step back now and then…
- Make sure to plan for the unplanned
Functional Coverage

- Don’t start too early
- Focus on the places where the bugs are
CHALLENGES
Configurability

<table>
<thead>
<tr>
<th>System feature</th>
<th>Cortex-A15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>1-4</td>
</tr>
<tr>
<td>Interrupt controller</td>
<td>Optional</td>
</tr>
<tr>
<td>Number of SPIs</td>
<td>0-224 in steps of 32</td>
</tr>
<tr>
<td>Power management</td>
<td>Optional clamp/power-gate control pins</td>
</tr>
<tr>
<td>Floating point / NEON</td>
<td>None, VFP Only, VFP and NEON</td>
</tr>
<tr>
<td>Error protection</td>
<td>None, L2 cache only, L1 and L2 cache</td>
</tr>
<tr>
<td>L2 cache size</td>
<td>512KB, 1MB, 2MB, 4 MB</td>
</tr>
<tr>
<td>L2 tag and data slices</td>
<td>00, 01, 02, 11, 12</td>
</tr>
<tr>
<td>L2 arb slice</td>
<td>Present or not</td>
</tr>
</tbody>
</table>

- $4 \times 9 \times 2 \times 3 \times 3 \times 4 \times 5 \times 2 = 25920$ total configurations 😞
- Exhaustive crossing of slices, ECC/no-ECC, number of CPUs at unit, top, and system level
- Focused directed testing of less intrusive configuration choices, then pairwise crossing in random testing
Virtualization: A Third Layer of Privilege

- Guest OS same privilege structure as before
  - Can run the same instructions
- New Hyp mode has higher privilege
- VMM controls wide range of OS accesses to hardware
Virtual Memory in Two Stages

Stage 1 translation owned by each Guest OS

Stage 2 translation owned by the VMM

Hardware has 2-stage memory translation

Tables from Guest OS translate VA to IPA

Second set of tables from VMM translate IPA to PA

Allows aborts to be routed to appropriate software layer

Real System Physical address map

Virtual address map of each App on each Guest OS

“Intermediate Physical” address map of each Guest OS

The Architecture for the Digital World®
Virtualization - Testing

- Constrained random testing of instruction/event traps at core level
- PageMaker – constrained random generation of LPAE/v7 pages
- Unit level: exhaustive testing of tbw logic in L2TLB/TBW
- PageMaker reused in memory system testbenches and top level testbench
- Independently developed Virtualization AVS
- “Real” hypervisor at system level, running real and rogue OSes/apps
Out of Order Execution
OoO in Cortex-A15
OoO - Testing

- Unit Level: detailed testbench models/checking
- Exhaustive fcov on retire/flush/rebuild scenarios
- Independent architectural checking vs. ISS model, AVS
Hardware Coherence
Hardware Coherence in A15

[Diagram of hardware coherence in A15 processor]
Hardware Coherence - Testing

- ACE functional coverage and protocol checkers
- Unit level: Detailed white-box modeling/checking
- Multi-unit: LS/L2
  - Focused hazard/starvation scenario testing
  - Global ordering data consistency checker
- Top level: RIS tests
  - False-sharing
  - Non-deterministic sharing
- System level: True-sharing, order-sensitive testing