How to Design with ARM Technology

Craig Slorach (ARM Cambridge)
刘澍 (ARM Shanghai)

Agenda

- The System Design Cycle
- Architecture Exploration
- System Development & Integration
- System Prototyping
- System Implementation
The System Design Cycle

1. Architecture Exploration
   - System Architecture
   - AMBA
   - System Development & Integration
   - Ready to Run Software

2. Physical Design
   - ReadView
   - System Prototyping
   - RTL (+ Design Confidence)
   - Ready to Run Software

3. System Implementation
   - Artisan
   - RTL
   - System Development & Integration
   - Physical Design

ARM “System Ready”

- ARM CPU Cores
  - Apps processors
  - Data Engines
  - Graphics Engines
  - Video Engines
- ARM CPU Cores
- ARM “System Ready”
- ARM System-on-Chip
  - Models
    - Instruction Set Simulator
    - Cycle callable
    - Micro-architecture Design Sign-off Design Kits
- Development Systems
  - Compiler, Debugger
  - Real Time Trace
  - Development Boards
  - RTOS awareness
- System Monitors
- Single/Multi Core Trace
eDSP debug and Trace
- Debug
- Single/Multi-core debug
- Bus visibility
- System Monitors
- Single/Multi Core Trace
eDSP debug and Trace
- On-chip buses
  - High performance bus
  - System bussing
  - Peripheral bussing
- AMBA design Kit
- SoC building blocks
  - Bus generators
  - Memory controllers
  - Peripheral controllers
  - DMA engines
  - Display controllers
  - Secure system controllers
  - Energy management
  - Platforms
  - Reference designs
- Software
  - Energy management
  - Security
  - 3D Graphics
  - Java support
  - New execution environments
- RTOs
  - Nucleus
  - ThreadX
  - 32-bit
- OS
  - WinCE
  - Symbian
  - Linux
- Design Environment
Supporting The Local Design Community

- Traditional support and training sourced from UK
  - English language, supported on UK/US office time

- **We can do better... support local design community directly...**

- Target: Provide localized training
  - Delivery of mainstream training material in Chinese by local FAE’s

- Target: Support local community directly
  - Local FAE’s doing a great job of this already
  - Out of hours support via ARM global support network
  - But, we can do better...
    - Building up China support group (Craig S in China next few months)
  - Goal: Locally based support in China
  - More detail on this coming soon!
Architecture Exploration

Key System Architecture Questions

- Application analysis
  - Which processor should I choose?
- Hardware/Software Mapping
  - What is in hardware, what is in software?
- System Partitioning
- Bus and Memory Architecture
  - How do I implement the control stream?
  - How do I implement the data stream?
Processor Choice: Product Features

Foundry Core Availability

<table>
<thead>
<tr>
<th>Architecture</th>
<th>ARM7TDMI</th>
<th>ARM922T</th>
<th>ARM946E</th>
<th>ARM1022E</th>
<th>ARM926EJ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>v4TE</td>
<td>v4T</td>
<td>v5TE</td>
<td>v5TE</td>
<td>v5TEJ</td>
</tr>
<tr>
<td>0.13*</td>
<td>125</td>
<td>258</td>
<td>200</td>
<td>364</td>
<td>243</td>
</tr>
<tr>
<td>Frequency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.13*</td>
<td>0.26</td>
<td>3.18</td>
<td>3.26</td>
<td>6.9</td>
<td>5.1</td>
</tr>
<tr>
<td>Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.13*</td>
<td>0.062</td>
<td>0.258</td>
<td>0.55</td>
<td>0.903</td>
<td>0.44</td>
</tr>
<tr>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus Interface</td>
<td>Native</td>
<td>ASB</td>
<td>AHB</td>
<td>Dual AHB</td>
<td>Dual AHB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache</td>
<td>8K</td>
<td>8K</td>
<td>32K</td>
<td>16K</td>
<td></td>
</tr>
<tr>
<td>MMU / MPU</td>
<td>MMU</td>
<td>MPU</td>
<td>MMU</td>
<td>MMU</td>
<td></td>
</tr>
<tr>
<td>TCM</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Figures quoted are final Silicon, Generic process unless otherwise stated.

Clock Frequency (MHz) is Worst Case Silicon, Area (mm\(^2\)), Power (mW/MHz) in Typical Silicon.

Please consult local sales office for exact performance for specific foundry processes.
System Level Design Challenges

- How do I understand performance of key IP blocks before designing?
- How to build an optimized system?
- System-on-Chip simply put IP into a chip

In 65% of the design fails, reason cited:
"limited visibility into the complete system"

Processor Performance

- Software Profile
- Debug Capability
- Trace Capability
- 3rd part IP Support
Bus Throughput

- AMBA Components ready
- Plenty of Primecell support
- Accurate bus profile
- Determine bus structure

Memory Usage

- Explore design trade-offs and impacts of different
  - Memories
  - Caches
  - Bus structures

Relative performance

- 128K L2 +104%
- 256K L2 +102%
- 512K L2 +74%
- No L2

MPEG4 Decode
ARM1136J-S
PrimeXsys Platform

74%
6mm²

128K L2 cache 59mm²
SoC on TSMC 0.13µm 6LM
ARM Offer Key Features

- Easy to build system
- Accuracy
- Simulation speed

Efficiently Verify Your SoC

Each parts can move smoothly to your SoC implementation
Maximising Concurrent Development

Example: Boot Linux on Versatile Platform
System Development & Integration

System Development Challenge

- **Issue:** I have my system design, how can I get my SoC quickly to market?

  - Library level approach to core IP integration with Design Kit
    - Goal: integration of CPU similar to Artisan SRAM
    - Full RTL → GDSII signoff flow support for Synopsys, Magma, Cadence

  - Key requirements for fast time-to-market
    - Proven CPU implementation & SoC IP (PrimeCell)
    - Example SoC systems used as basis for SoC design (ADK)
    - Out of box platform solution (PrimeXsys)
    - Automatic generation of SoC Fabric (AMBA Designer)
CPU Design Kit Overview

- Black box industry standard views supplied for simple integration
  - Large number of tape-outs and working Silicon prove the design kit
  - Comprehensive documentation also supplied

![Diagram of CPU Design Kit Overview]

---

Functional Verification (RTL Simulation)

- During SoC design flow, require a functional view of the ARM core for simulation
  - Simplified through abstract model as with other library components (e.g. SRAM)
- Design Simulation Model (DSM) supplied
  - Compiled model based on core functionality
  - Simulator / Platform Specific
- Additional Features
  - Programmers Model (R0~R15) visible
  - Instruction Tracing (EIS Trace)
- Other development models available

![Diagram of Functional Verification (RTL Simulation)]
Timing, Layout & Power

- Views supplied suitable for integration into common EDA flows
  - Synopsys, Magma, Cadence
- Industry standard .lib model supplied for all cores
  - Synthesis and STA
- EDA Vendor specific views supplied for physical implementation flow
  - Synopsys / Avant! (FRAM / CLF)
  - Cadence (LEF / TLF)
  - GDSII Abstract + Layer Mapping File
- Antennas
  - CLF / LEF antenna views supplied for repair at SoC level
- Noise and Power Grid Models currently under development
  - mW / MHz datapoint supplied for simple power analysis

Design for Test: ARM7TDMI / ARM922T

- Full Custom Cores: Functional Verification Methodology
  - Application of vectors via AHB infrastructure (fast and portable)
  - ARM7TDMI can apply via JTAG interface (low pincount)
- Example (ARM7TDMI)

Components and Design Examples Included In ADK
Design for Test: ARM946E, ARM926EJ

- ‘Synthesized Designs’: Scan Based Testing Methodology
  - Pre-defined ATPG Patterns Supplied for Standard Logic
  - BIST Patterns Supplied for Memory BIST
  - Proven Porting and Verification Flow

![Diagram of dual function wrapper](image)

- Dual Function Wrapper operates in both internal and external test modes
- Wrapper Netlist Supplied For ATPG
- 'Canned' Vectors supplied for testing core

Signoff Simulations

- Design Simulation Model (DSM) is a view built from the original RTL of the processor core
  - Consequently, does not include detail on the implementation features of the core
  - BIST / Scan Chain Connectivity
  - For ARM7TDMI/ARM922T not an issue

- How to replay test vectors to ensure correct DFT integration of core?
  - Use Sign-Off Model (SOM)

- SOM is a model built from the processor netlist
  - Integrated into simulation flow in a similar manner to DSM (but different model)

- SOM also supports direct SDF timing annotation (c.f. other library components such as memories)

![Simulation Environment](image)
**AMBA Design Kit Overview**

- **Contains:**
  - Basic AMBA infrastructure and simple system components
  - Example systems to use as a starting point for simple AMBA-based systems
  - **APB components**
    - Example APB slave
    - GPIO
    - Remap/pause controller
    - Watchdog
    - Counters
  - **Example AMBA systems**
    - FRBM-based AMBA system
    - ARM7-based single-layer AMBA 2 AHB
    - ARM9-based multi-layer AMBA 2 AHB

- **AHB infrastructure and components**
  - Multi-layer AHB bus matrix
  - Single-layer AHB components
  - Example AHB Bus Master
  - Example AHB slave (with Retry)
  - AHB to AHB Bridge
  - AHB to APB bridge
  - DWC
  - AHB-lite to AHB wrapper
  - Interrupt Controller
  - Default slave
  - Reset controller
  - TIC – Test Interface Controller for hard macrocell ARM (eg ARM7TDMI, ARM922T)

- **Behavioural models**
  - External and internal memory models (RAM and ROM)
  - FRBM – bus functional model
  - Tube – console text output model
  - TIC driver model

---

**ARM926EJ PrimeXsys Platform**

- Sys-Cntl
- Timer
- Watchdog
- VIC
- AMBA
- Console
- Misc
- Multi-Layer AHB
- GPD
- GPD
- RTC
- UART
- SD
- ESD
- DMAC
- MMC

**ARM926EJ-S™ PrimeXsys™ Platform**

**OS Ports**

**Development tools & boards**

**Platform IP**
AMBA Automated Design Flow

AMBA Designer Improves Time-To-Market

- Design Automation solution for AMBA
- True “Plug-and-Play” architecture
- Efficient IP re-use
- Time-To-Market improvement based on “Model first” philosophy
- Right 1st time design
- Automatically generates SystemC, RTL and Validation
- Plug-in to MaxSim technology
- Common look and feel
- Supports Hardware/Software co-development and co-simulation
- Enables design at transaction level for fast simulation speeds
- Seamless integration into 3rd Party EDA tools via SPIRIT
Supporting Low Power Design

- ARM cores already targeting low power implementations
  - CPU architecture includes low power features (clock gating and other low power implementation techniques)
  - Many cores available on low power processes

- Artisan Physical IP supports low power implementation

- Future innovations on low power design through Physical IP and partnership with wider community
System Prototyping Challenge

- Want to verify design in real hardware before tapeout
  - Need a hardware platform to run OS & RTOS

- Need to adopt a common hardware platform
  - Concentrate on system - don’t waste time on debugging a PCB!
  - Open interface enables flexible extension of system

- Low cost and low risk solution

Remove PCB design and manufacturing effort... and concentrate on debugging your system, not your PCB
PrimeXsys Example

RealView - Product Families

Developer Suite
- Compilation Tools
- Debug Tools
- Instruction Set Simulator

Debug Hardware
- System Control
  - RealView RVI
  - Multi-ICE
- Data Capture
  - RealView RVT
  - Multi-Trace

Hardware Platforms
- System Prototyping
- Integrator™ Family
- Versatile Family
- System Emulation
- Soft Macrocell Models
System Implementation Challenge

- Where do I get the Physical IP to build my design?
  - Implementation of SoC design requires broad range of Physical IP

- Hard Macro CPU for simply dropping into your design
  - No lengthy implementation cycle involving soft-IP
- Cell libraries/ I/O Cells and SRAMs
  - Broad offering supports different PPA goals
- High Speed PHYs for Standards IP support

**ARM can provide all Physical IP required for implementing a complex SoC**
### ARM’s Artisan Physical IP Platforms

**Classic**
- SAGE-X™ Standard Cell
- Single/Dual Port SRAM
- 1/2-Port Register File
- ROM Via or Diffusion
- GPIO Inline/Staggered
- Power Management Kit
- Analog/Mixed Signal PLLs, DLLs, Specialty I/Os

**Advantage**
- Advantage™ Standard Cell
- Single/Dual Port SRAM
- 1/2-Port Register File
- ROM Via or Diffusion
- GPIO Inline/Staggered
- Power Management Kit
- Analog/Mixed Signal PLLs, DLLs, Specialty I/Os

**Metro**
- Metro™ Standard Cell
- Single/Dual Port SRAM
- 1/2-Port Register File
- ROM Via or Diffusion
- GPIO Inline/Staggered
- Power Management Kit
- Analog/Mixed Signal PLLs, DLLs, Specialty I/Os

**Velocity**
- Power Management Kit

**Mainstream Platform**
- Advantage™
- Metro™
- Velocity™

---

### Summary
ARM Offers Complete System Solution

+ All of the Physical IP to Realize your Design