Developing and Optimizing Linux on ARM

CELF Plenary Meeting

San Jose, 2005

Philippe Robin

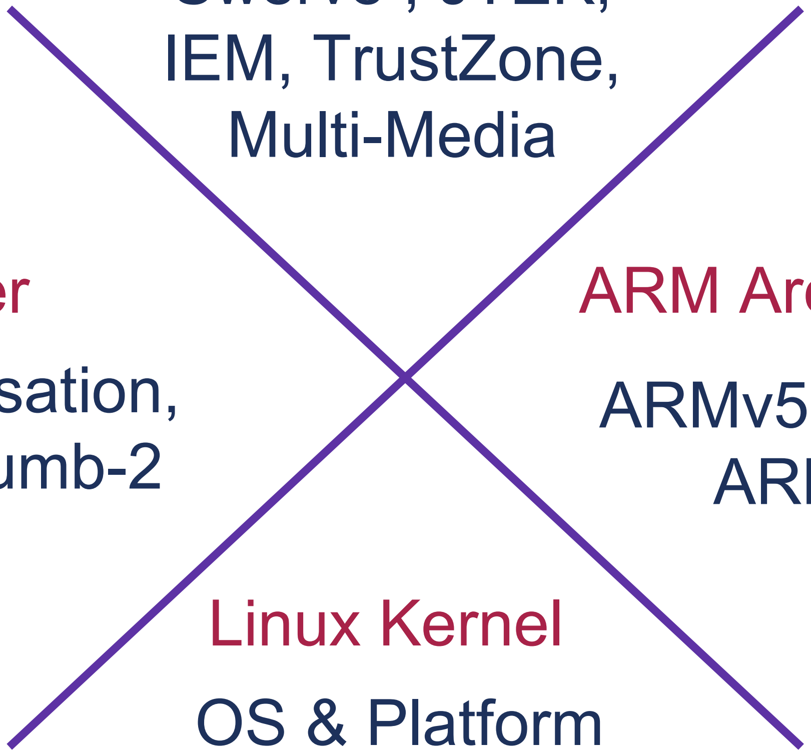
Philippe.Robin@arm.com

ARM Ltd.

Overview

- Introduction
- Areas of optimization
  - Hardware optimisations
  - Development tool chain
  - Kernel and applications
  - Power Consumption, Security, Multiprocessing
  - Test and validation environment
- Evolution of the ARM Architecture
  - Impact on Linux kernel
  - Use of architectural features
- Development tools
- Summary

Linux Platform Components



Libraries and Applications

Swerve , JTEK,
IEM, TrustZone,
Multi-Media

Compiler

Code Optimisation,
Thumb, Thumb-2

ARM Architecture

ARMv5, ARMv6,
ARMv7...

Linux Kernel

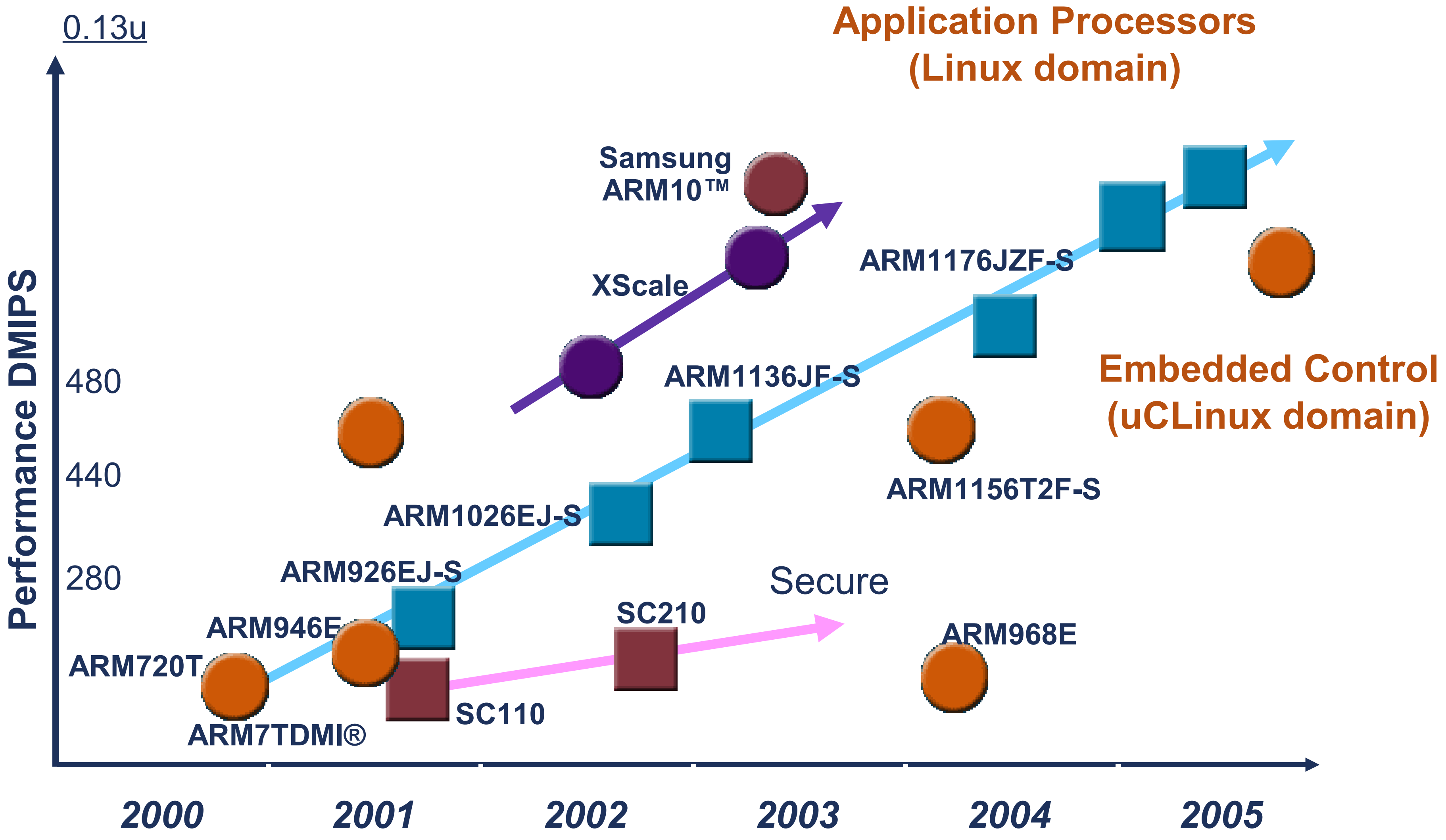
OS & Platform
support

ARM Architectures

	Feature Set			
Architecture	THUMB™	DSP	Jazelle™	Media
v4T	✓			
v5TE	✓	✓		
v5TEJ	✓	✓	✓	
v6	✓	✓	✓	✓

- Enhance performance through innovation
  - THUMB™: 35% code compression
  - DSP Extensions: Higher performance for fixed-point DSP
  - Jazelle™: up to 8x performance for java
  - Media Extensions up to 4x performance for audio & video
- Preserving Software Investment through compatibility

ARM CPU Roadmap

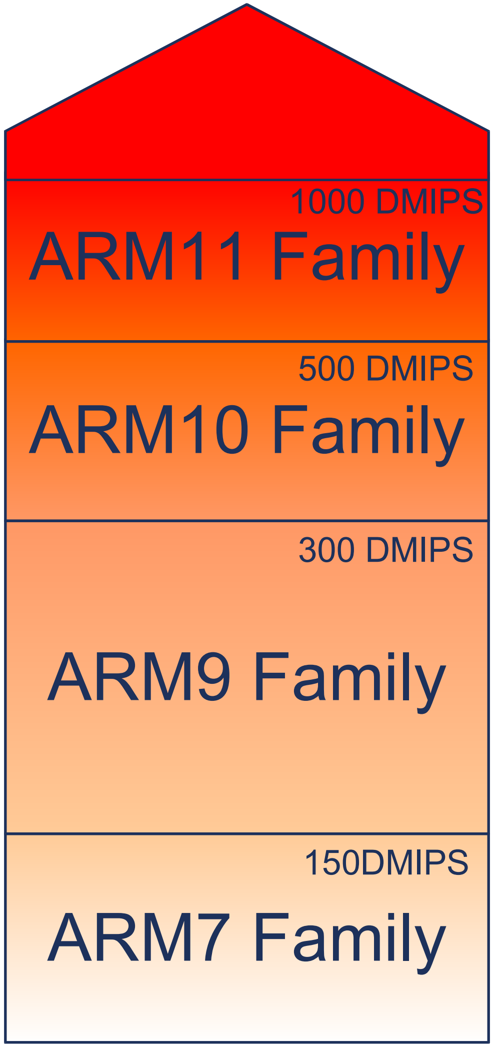


ARM®

THE ARCHITECTURE FOR THE DIGITAL WORLD™

Increased Processor Performance

One Processor Architecture



Home Media Centres

Digital TV

Digital Set Top Box

PDA’s

Smart Phones

Router/Firewall

Cable XDSL Modems

PC Network Cards

Digital Camcorders

Digital Cameras

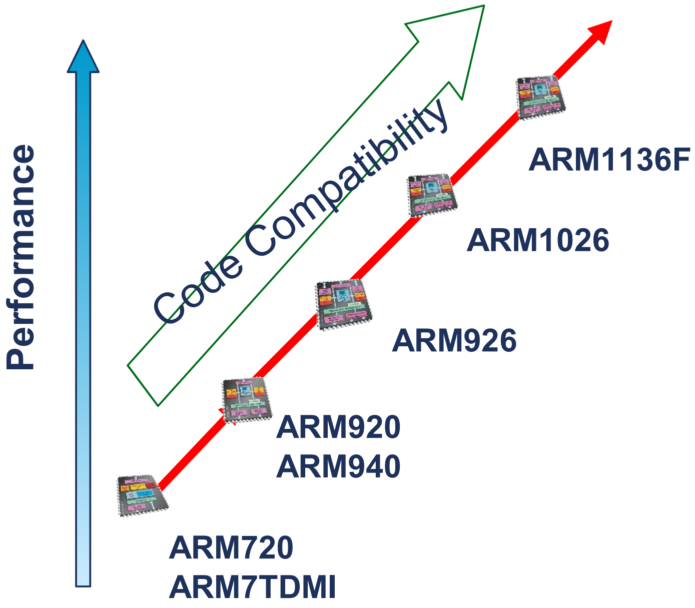
Digital Audio players

Digital Photo Frames

THE ARCHITECTURE FOR THE DIGITAL WORLD™

Performance Gains

- Hardware optimizations for
  - MMU and Cache management
  - Interrupt handling
  - Real-Time
  - Code density
  - Multi-Processor
- Compiler and tool chain
  - Instruction scheduling
  - Use of new instructions
  - Code density
- Linux support
  - Optimize Linux kernel to fully utilize new architectural features

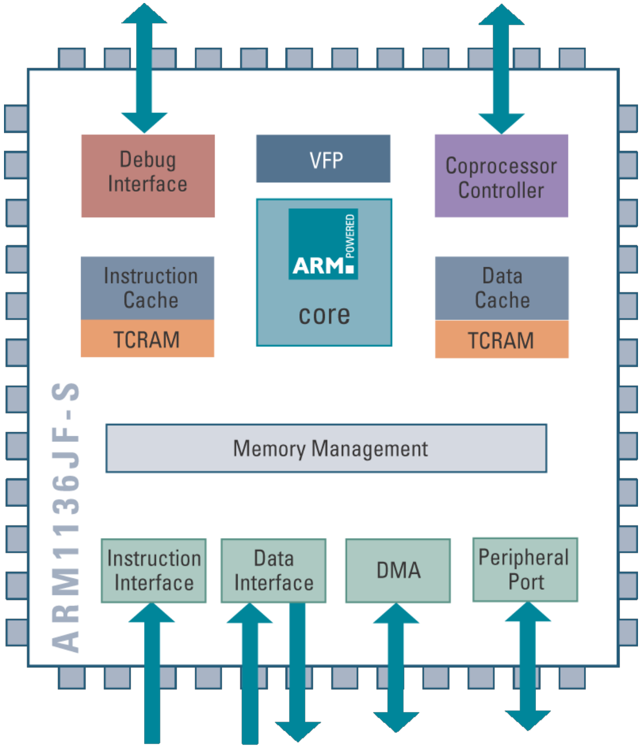


ARMv6 Architecture

- Compatibility with previous ARM architectures
- SIMD Media Instructions
  - 1.75X faster at media processing compared to ARMv5
- Improved Memory Management
  - Boost system performance by up to 30%
- Improved Mixed Endian and Unaligned data support
  - Improved processing of Big Endian data (eg. TCP/IP) in Little Endian (LE) systems
- Improved Interrupt latency for real time systems
  - Improved from 35 cycle worst case to 11 cycles in v6

The ARM11 Processor Family

- Based on ARMv6 architecture
  - Media SIMD
  - Fast interrupt modes
  - Jazelle™
  - Three power modes
    (Full, Standby and Dormant)
  - Tightly Coupled Memory (TCM)



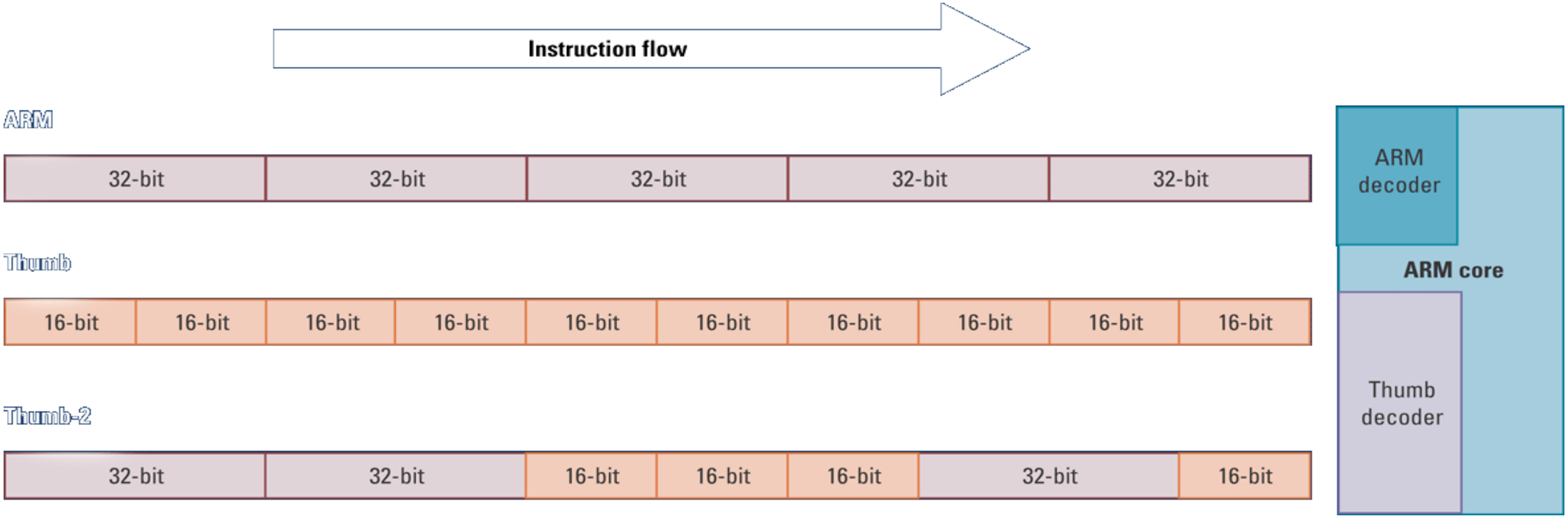
- High speed, performance
  targeting embedded and application processing

Enhancements from ARM1136J-S™ Core

- ARM TrustZone™ architecture extensions for CPU and system security
  - New secure state enabling creation of a trusted computing environment
  - Enables protection of code and data across entire memory hierarchy
- AMBA™ 3.0 (AXI) System Bus Interface
  - Higher data bandwidth, easier timing closure
  - Supports access to secure-aware memory and peripherals
- Intelligent Energy Manager (IEM) Compatible
  - Allows dynamic voltage and frequency setting under OS control to optimize energy usage / battery life
  - Supports multiple voltage domains for power-saving modes

Thumb-2 & Embedded Processors

- Thumb-2 core technology is an enhancement to the ARM architecture version 6.
- Thumb-2 core technology consists of:
  - new 16-bit Thumb instructions for improved program flow
  - new 32-bit Thumb instructions for improved performance and code size
  - new 32-bit ARM instructions for improved data handling



Linux Kernel – ARMv6 Support

- Optimize memory and cache handling
  - Minimise cache flushing
    - Benefits from Physically tagged cache
  - Prevent cache aliasing incoherencies
- Faster interrupt handling
  - Use of new CPS instruction to reduce number of cycles needed to handle interrupts
- Use Application Space Identifiers (ASIDs)
  - Optimize context switch time
  - Avoid need to flush on-chip translation buffers

Areas of Optimizations

- Real-Time support and performance
  - Open source and proprietary projects
    - Scheduling policies, interrupt handling, threading model etc.
  - Use regression test suites to validate and improve kernel performance and reliability
- Libraries
  - Reduced size and choice of optimised libraries
    - Floating point libraries, C libraries etc.
    - ARM ABI will allow more choices
- Power Management
  - Intelligent Energy Management (IEM)
  - Montavista Dynamic Power Management (DPM)
- Security and reliability
  - Encryption and protection mechanisms
  - Build on TrustZone technology
- SMP support
  - Add changes in kernel to support multiprocessor platforms
    - Synchronization, interrupt handling…

Key ARM Software with Linux

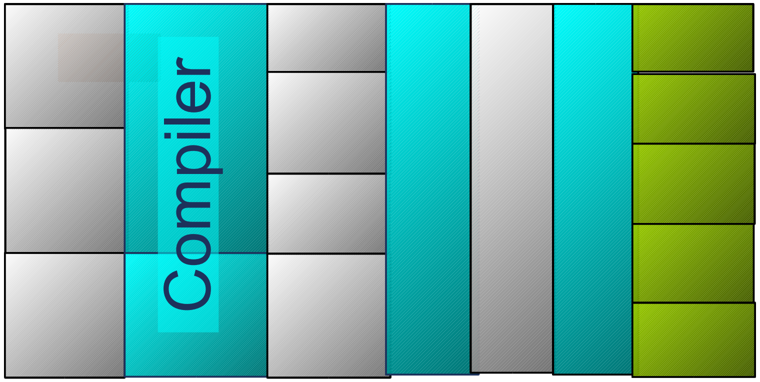
- Jazelle for Java bytecode acceleration
  - 3x to 8x time faster Java bytecode execution
  - Execute some parts of the Java Virtual Machine in hardware
- Power Management
  - IEM allowing savings up to 25% of battery life
  - Scale CPU frequency and voltage based on monitoring of the system activity
- 3D Graphics
  - Swerve: Industry-leading JSR-184 for 3D content
  - Also take benefit of hardware VFP support
- Security
  - TrustZone for device integrity and secure transactions
  - Partition and control the execution environment to prevent illegal access to critical code or data

Linux & Development Tool Chain

- Compiler is a key element in generating efficient and compact code
  - Requires in-depth knowledge of the micro-architecture
  - Support for latest architectural features
  - Requires extensive testing and validation
- Choice of development tools
  - New ARM Application Binary Interface (ABI) aims at providing compatibility between multiple tool chains
  - Allow re-use of libraries and existing code base
    - Can mix GNU based objects with libraries or objects optimized with other proprietary tool chains
  - Closely linked with debug and profiling tools

Supporting GCC and Linux for ARM

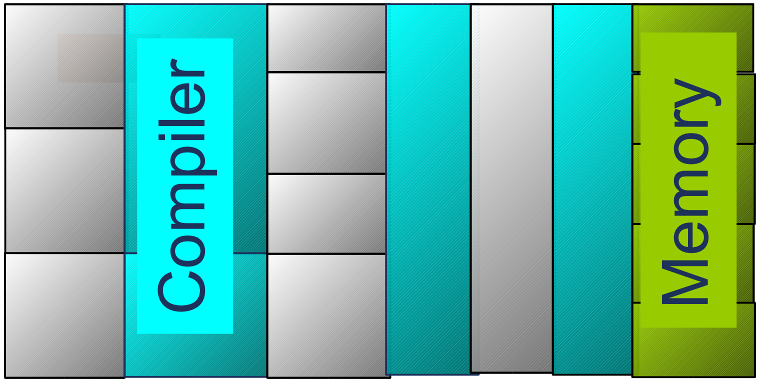
- ARM enabling GNU
  - Formal collaborative program to create a professionally supported ARM GNU Compiler



- Goals of the GCC project
  - Create stable releases of the ARM GCC compiler
  - Improve ARM architecture and micro-architecture support
  - Comply with the ABI for the ARM architecture
    - Enables inter-working of GCC and the RealView Developer Suite RVCT compilation Tools
    - Enables mixing of object code from both tool chains
  - Produce a binary release every 6 months
  - Enable support for targeting embedded Linux systems
- Available publicly through CodeSourcery's website

RealView Creating Optimal Reliable Code

- Processor-specific optimizations
  - Code scheduled to make best use of pipeline structure of the processor
  - Peephole optimization to generate optimal code sequences



- Selectable optimization levels
  - Allows choice of best debug view or best code view
  - Orthogonal to debug flag, so can produce debug capable, optimized code
  - Choice of optimization for speed or code size to suit system requirements

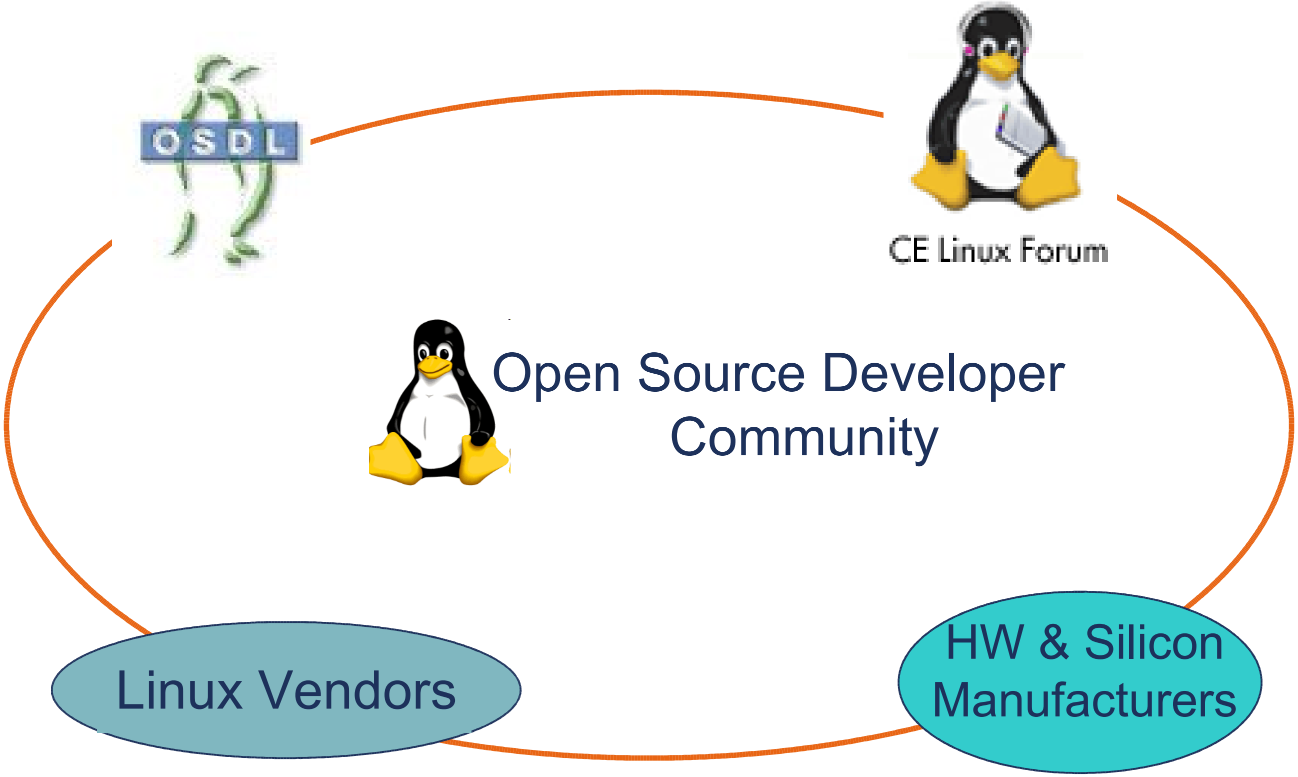
RealView - Optimizations

- Removal of unused code
  - The compiler removes code sequences that are never executed, thus saving memory
  - The linker removes unused code sections and unused functions, thus saving memory
- Reducing the Power Consumption
  - With extensive performance optimizations
    - Increase instruction-throughput with no increase in clock frequency
  - With powerful code size optimizations
    - Small code size makes better use of I-Cache
    - Small code size reduces instructions to execute

Summary

- Each component plays an important role in achieving optimum performance
  - Processor, compiler, kernel, libraries and applications
  - Each must cooperate to optimize use of hardware resources
  - Optimizations are domain specific as each environment has specific performance and resource requirements
    - Adapt Linux kernel accordingly
    - Tools need to address performance requirements
    - Choice of the processor according to the targeted product
- Test and validation play a key role in maintaining and improving code quality and performance
  - Access to standard maintenance and validation test suites

Linux Open Source Community



Improving Linux through cooperation!