Secure Virtualization on Single- and Multi-core ARM Processors

Brad Suessmith
Senior VP & GM
TRANGO Virtual Processors

John Goodacre
ARM Multiprocessing Program Manager
ARM Limited
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
TRANGO Overview

TRANGO Company
- Headquartered in Grenoble, France
- Sales and support offices in US, Europe, and Japan
- Technology created by former CPU designer

TRANGO Hypervisor
- Secure virtualization IP for SoC vendors and OEMs
- Support ARMv4, ARMv5, ARMv6 architectures
- Secure Execution of multiple Operating Systems and Applications on single- and multi-core platforms

ELSYS Design Holdings
- SoC hardware and software design services
- 35M€ annual revenue
- 500 engineers
## Target Markets

<table>
<thead>
<tr>
<th>Wireless Handset</th>
<th>Point of Sales Terminal</th>
<th>Network Equipment</th>
<th>Consumer Electronics</th>
<th>Automotive Infotainment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>BOM reduction</td>
<td>Load balancing</td>
<td>Security</td>
<td>Real-time capability</td>
</tr>
<tr>
<td>GPL enabler</td>
<td>Maintain security</td>
<td>Quality of service</td>
<td>GPL enabler</td>
<td>Complex software enabler</td>
</tr>
<tr>
<td>Real-time</td>
<td>Faster certification</td>
<td>Device</td>
<td>BOM reduction</td>
<td></td>
</tr>
<tr>
<td>capability</td>
<td></td>
<td>management</td>
<td>Complex software</td>
<td></td>
</tr>
<tr>
<td>BOM reduction</td>
<td></td>
<td>enabler</td>
<td>enabler</td>
<td></td>
</tr>
<tr>
<td>Complex software</td>
<td></td>
<td>Software re-use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enabler</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*Increase revenues, add new product features with faster time-to-market and lower costs*
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
Typical Heterogenous System

- High cost: development, validation, silicon area
- Long time to market
Availability of high-performance ARM CPUs enables consolidation of functions

- Decreased development, validation, silicon area, time to market
## Emulation vs. Virtualization

### Emulation
- Emulate ISA
- Execute non-native OS & apps

### Full-Virtualization
- Execute multiple unmodified OS’s on single CPU
- High overhead & large memory footprint
- Suitable for servers/desktops

### Paravirtualization
- Small overhead & memory footprint
- Execute multiple ‘ported’ OS’s on single CPU
- Real-time performance
- Best fit for embedded applications

---

**TRANGO Hypervisor**

**CPU CORE**
- Memory
- Peripherals

**Ported OS**
Terminology

- **Hypervisor**
  - Coined by IBM
  - Also known as a “Virtual Machine Monitor”

- **VMM: Virtual Machine Monitor**

- **VM: Virtual Machine**
  - ‘Container’ or ‘sandbox’ that guest OS and applications execute in
  - Not directly related to Java JVM

- **VPU: Virtual Processor Unit**
  - TRANGO term for guest OS view of virtual machine
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
Technical Overview

TRANGO Hypervisor Secure Virtualization
- Design inspired by RISC
- Virtual Machine isolation
- Run any OS, RTOS, application
- GPL and open-source license isolation

ARM CPUs
- ARM9/ARM10/ARM11 families
- Multi-core
- MPU or MMU required!

Features
- 25 - 30KB code size
- < 2% performance overhead
- User application binary compatibility
- No emulation
- Paravirtualization
Hypervisor ISA

Hypervisor

Hardware

Kernel Mode

User Mode

CPU

VPRs

R/W

RO

Interrupts

Exceptions

Privileged Operations

VM

Hypercalls

User Instructions

ARM Developers' Conference
& Design Pavilion 2007

TRANGO Virtual Processors
Security Architecture

- **Strongly Partitioned Spaces**
  - Virtual User Mode
  - Virtual User Mode
  - Virtual User Mode

- **MMU barrier**

- **Privileged/Unprivileged Barrier**

**_TRANGO Hypervisor (25 - 30KB)_**

- Only TRANGO Hypervisor has access to CPU privileged mechanisms (mode, MMU...)
- Virtual Machine isolation ensured by the MMU/MPU & managed by the Hypervisor
- All events (interrupts, exceptions) are trapped by the Hypervisor
TRANGO Leverages TrustZone™

- Hypervisor operates in Monitor Mode
- Security critical VMs run on-chip
  - Protection against hardware attacks
  - Control of AMBA™ AXI S/NS bit
- Normal VMs run off-chip
- Privileged mode not used
Eclipse Tools Suite

Features:
- Intuitive and user friendly graphical interface
- VM development project
- System integration project
- Integrated project builders
- Shared network connection to boards
- Remote launch
- Remote Virtual Machine consoles
- Real-time system monitoring
- JTAG-like secure debug
- Command line interface option
- Flexible development framework
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
“....there are serious problems with the design and implementation of security on the iPhone. The most glaring is that all processes of interest run with administrative privileges. This implies that a compromise of any application gives an attacker full access to the device.”

Proposed Secure iPhone Architecture

Security Critical Functions

Multimedia Open Software

Quick Time
Safari Web Browser

iTunes
Media Keys
FairPlay DRM
Contacts
Voicemail

OS X

TRANGO Hypervisor

ARM CORE

HARDWARE PERIPHERALS

Software

CPU User Mode

TRANGO Virtual
User Mode

CPU Kernel Mode

Hardware

TRANGO Virtual
Kernel Mode

Software

TRANGO Virtual
Basic Single-Core Wireless Smartphone

Today's Smartphone

- Telecom Stack
  - CPU Core
    - Telecom Peripherals
  - Application Stack
    - CPU Core
      - USB
      - Camera
      - ...

Smartphone with TRANGO

- Telecom Stack
  - TRANGO HYPERVERSOR
  - Application Stack
    - CPU Core
      - Telecom Peripherals
      - USB
      - Camera
      - ...

- Bill of materials reduction with use of a single core
- Integration of standard “Rich” OS (Windows Mobile, Symbian, Linux, etc.)
- Legacy software re-use
- Robust device platform architecture with HW resource partitioning
Advanced Single-core Smartphone

Today's Smartphone

- CPU Core
  - Telecom Peripherals
- Open OS
- Telecom Stack

Smartphone with TRANGO

- CPU Core
  - Telecom Peripherals
  - USB
  - Camera
  - ... 
- Open OS
- Telecom Stack
- Security Stack
- TRANGO HYPERVISOR
  - Crypto

- Isolated security stack
- Secure deployment of new services: DRM, crypto, e-commerce, updates...
- Scalable over full spectrum of wireless handsets and services

ARM Developers’ Conference & Design Pavilion 2007
NTT DoCoMo: OSTI Specification

Standard Handset Features

Always-on and Highly Reliable Environments

Environment #1:
• Core services proprietary code
• Critical code for security app (DRM...)

Environment #2:
• Build on Operator OS
• User personal data
• Operator services

Customized Features

Open Environment
• Enterprise/User OS
• Enterprise managed environment
• Enterprise business applications
• User personalized game software

Improved Performance:
• 1,000,000 times faster than OS switching (switch time is 2μs vs 2s)
• Better CPU optimization and power management compared to a dual-core SoC
• Scalable security

Reduce Architecture Associated Costs:
• BOM reduction, e.g. silicon area
• Development and validation

http://www.nttdocomo.co.jp/english/corporate/technology/osti/

ARM Developers’ Conference & Design Pavilion 2007
Point-of-Sale Terminals

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Dual-Core/Chip</th>
<th>Single-Core with TRANGO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E.g. ARM7 + ARM9</td>
<td>E.g. ARM9</td>
</tr>
<tr>
<td>Certification process</td>
<td>Equivalent certification process</td>
<td></td>
</tr>
<tr>
<td>Additional features</td>
<td></td>
<td>Modular and scalable architecture</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td>30% BOM reduction</td>
</tr>
</tbody>
</table>

Critical HW and SW for PCI-PED Certification

EMV Application
Application CPU
Smart Card
LCP
Keypad
Ethernet
UART
(USB)...
GPL Code Isolation

GPL constraints:

- **v2**: Disclosure of Linux-based product source code (Linux kernel, drivers and modules)
- **v3**: Provide way to update and reflash applications in product

Enable execution of Linux in a separated environment, TRANGO:

- Protect proprietary IP & critical product features
- Secure GPL-based product business models
- Ensure compliance with GPL license and GPL spirit
- Continue to benefit from open-source software

ARM Developers’ Conference & Design Pavilion 2007
Migration from Legacy OS

- Reuse legacy code & ensure continuity with existing services
- Reduce migration costs to new SMP OS

Validated legacy OS Re-use
Isolated Proprietary Code from GPL

Market Standard OS Integration

Binary Compatibility for Applications
A packaged system solution provides benefits across
the whole value chain:
- Modularity
- Cost reduction
- Easier S&M
- Protection of proprietary software
- Increased profits
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
Multi-core Hypervisor Detail

**Single-core hypervisors + comms:**
- One TRANGO hypervisor per CPU core
- Enables hardware and peripheral partitioning
- Provide multi-VPU architecture, capable of running multi-core (SMP) Operating Systems
- Same hypervisor properties
- Same validation, reliability and predictability
Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
Basic Multi-core System

- One execution environment per core
- Under-utilized CPU core resources
Hypervisor-enabled Multi-core System (1)

- Support of SMP OS and RTOS
- CPU core resource sharing
- Dynamic allocation of CPU resources
- Optimal load balancing
One TRANGO Hypervisor per core

Each Hypervisor can create multiple virtual processors

Execution Environment can share resources from different CPU cores
Hypervisor-enabled Multi-core system (3)

- Ability to switch off core
- CPU resources dynamically allocated according to Execution Environment requirements
Hypervisor-enabled Multi-core System (4)

- Ability to migrate all Execution Environments to one core
- CPU resources dynamically allocated according to Execution Environment requirements
- Large power savings
Multi-core: High RAS

- Redundancy of OS’s on the same hardware
  - If *Active Linux* fails, *Health Monitor* activates *Silent Linux*
- On-the-fly updates with very low down time:
  - Health Monitor updates *Silent Linux* in the background
  - Then switches *Active Linux*/*Silent Linux*
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
Paravirtualization Enhancements to the ARMv7 Architecture

John Goodacre,
Program Manager, Multiprocessing,
ARM Ltd

2nd October 2007
ARM Multicore Technology

- ARMv7 is the instruction set architecture (ISA) common to the ARM Cortex™ processors

- **ARMv7 MP Extensions** for next generation multicore processors
  - Hardware cache coherence
  - Enhanced communication
  - Hardware processor coherence
  - **Multicore paravirtualization**

- Targeting both increased processing efficiency and software simplification
ARM TrustZone® and GIC Architectures

- Previously extended the ARM architecture to include additional privilege levels for the execution of an additional OS within the secured software domain.
- Along with ARMv7MP the architecture has been further extended to manage and secure processor interrupts between the normal and secured software domains.
  - Known as the ARM Generic Interrupt Controller (“GIC”) Architecture.

Together, a TrustZone and GIC enabled ARMv7MP processor can:
- Restrict and control interrupts between the two software domains.
- Partition and control access to system resource between the normal/secure domains.
- Host ‘management’ software within the secured domain without altering the normal OS.

![Diagram showing the relationship between Interrupts, GIC normal/secure routing, Privileged (OS), User (apps), Non-secure Privileged, Non-secure User, Secure Privileged, and Secure User.]

GIC provides an architected way to isolate and control which software domain receives which interrupts.

TrustZone adds a “second OS context” to allow trusted programs and data to be safely separated from the open operating system and applications.
ARMv7MP: Improved Paravirtualization

- Today’s Paravirtualization solutions
  - Provide the ability to run multiple independent OS/RTOS on a single processor
  - Requires a notable modification of a OS port to appropriately defer all privileged operations to a virtual machine manager (VMM) managing processor resource sharing

- ARMv7MP processors
  - Enable concurrent execution of multiple paravirtualized operating systems
  - Providing improved real-time response and dynamic load balancing

- ARM TrustZone Architecture
  - Allows the open OS to maintain their User and Privilege states and run the VMM in the privilege contexts of the secured software domain
  - Provides an AMBA architecture mechanism to signal TrustZone context accesses to peripherals and any system based memory protection units
  - Cache state is maintained and secured between the open and TrustZone OS context during virtualization traps and request forwarding

- ARM GIC Architecture
  - Allows the VMM to manage and arbitrate access to drivers for each open OS

- Together providing an accelerated and simplified paravirtualization solution
Conclusions

The ARMv7 MP extensions set to further enhance ARM multicore processors

- Hardware support for Processor Coherence
- Extending L1 cache coherence to system accelerators
- Enhanced support for OS paravirtualization

Increasing performance scalability and further reducing the power associated with maintaining SMP

ARMv7: The ISA architecture of the ARM Cortex processors
Agenda

- Company and market overview
- Introduction to virtualization
- Single-core systems
  - Technical overview
  - Use cases
- Multi-core systems
  - Technical overview
  - Use cases
  - ARMv7MP virtualization enhancements
- Summary
Summary

- Consolidation of multiple CPU cores into one
  - Reduced cost
- Partitioning of execution environments
  - Improved security
  - GPL code isolation
  - Improved software packaging
- Multiprocessor systems
  - Improved load-balancing & CPU utilization
  - Increased RAS
  - Power savings through dynamic VM allocation
- ARM adding architectural hardware support
  - Improve performance
  - Minimize guest OS modifications