Android on Power Architecture

Speaker: Nina Wilner, Power.org

ELC, Grenobles, October 16th 2009

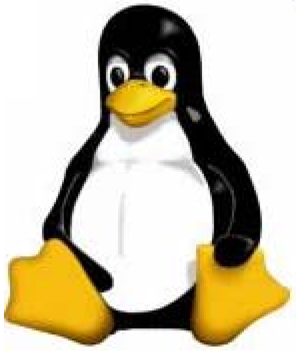
Agenda

- What Android IS (and ISN'T)
- Android Overview and Positioning
- Android Invasion
- What Makes Android Interesting and what Android can do
- Porting Android to Power Architecture
- Demo: MPC8536E-Android Platform – The underlying hardware

What IS and ISN’T Android

What IS (and ISN'T) Android

- Android is built on Linux (i.e. Linux Kernel)
  - Android is not Linux (i.e. Uses own libraries, tools, ...)
  - Android is a complete stack on top of the Linux kernel
- Software Platform versus OS
  - Android is an application framework on top of Linux
    - facilitating rapid deployment in many domains
  - Android is the enabler for a broad application developer base
  - Android is Open and Free
    - Operated under Apache License (majority of code)
  - Initially developed by Google, now managed by the Open Handset Alliance (OHA)
  - Developers write managed code in Java language via Android Java libraries
- OHA Listed Android Characteristics:
  - Open, Fast & easy application development
  - All applications are created equal, Breaking down application boundaries



Where to find Android and Android Information

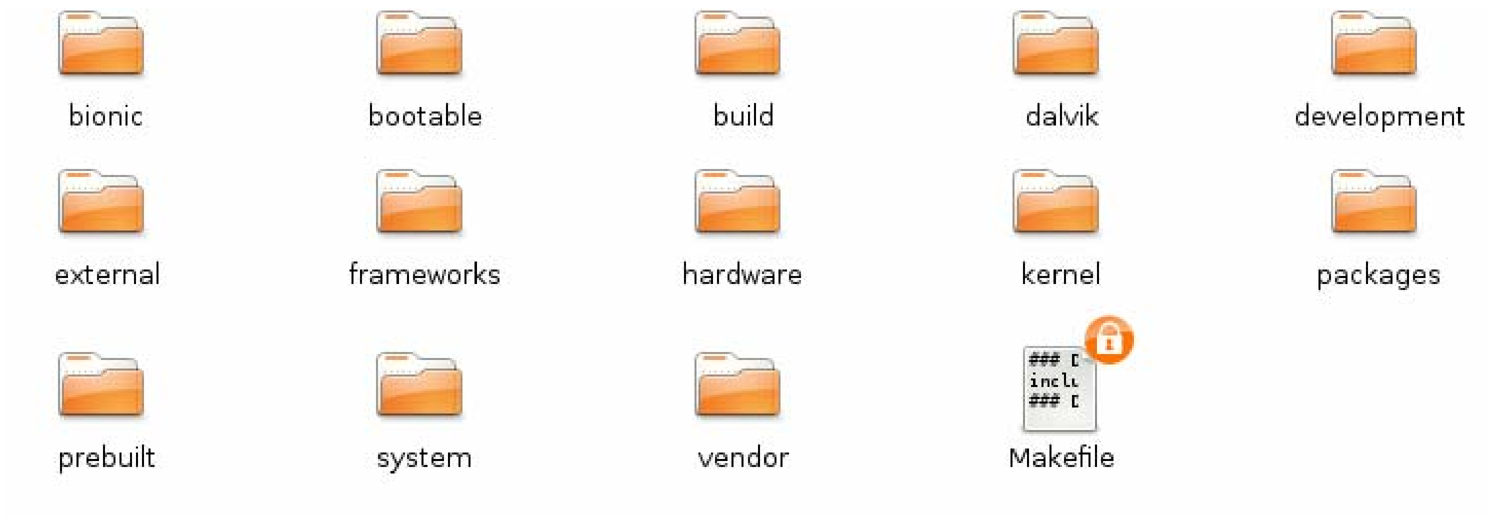
Android™ delivers a complete set of software for mobile devices: Operating system, middleware and key mobile applications.

URLs: http://www.android.com/
http://source.android.com/
http://developer.android.com/
http://androidguys.com/
http://source.android.com/roadmap/cupcake

The Android Software Development Kit (SDK) is also available:
http://developer.android.com/sdk/1.1_r1/index.html

Code

- Size: about 1.3G
- Core projects: the foundation of the Android platform
- External projects:
  - other open sources
- Packages: standard Android applications and services



Google And the Open Handset Alliance (OHA)

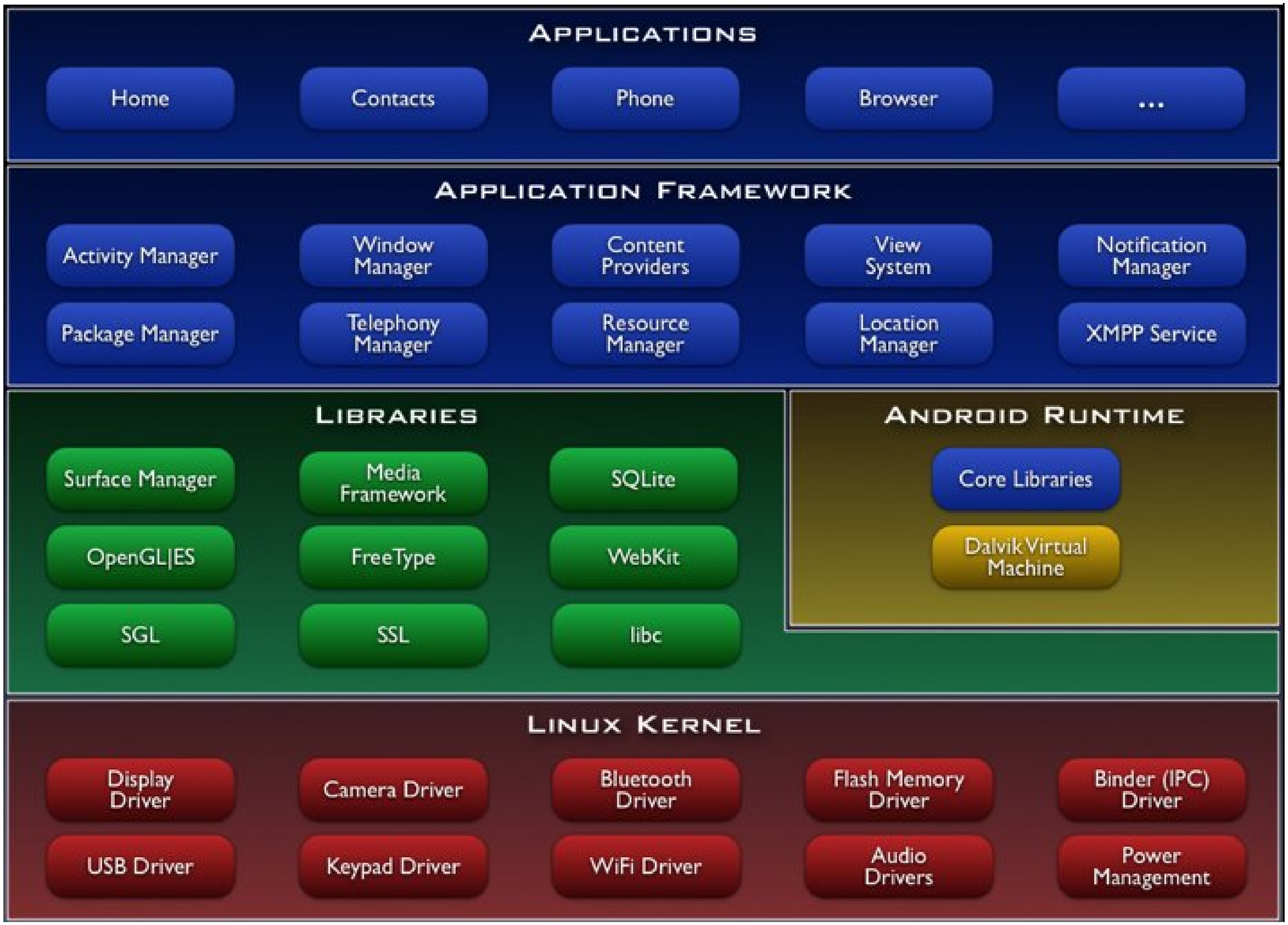
- Google Started the Android Project
- Android is not governed by the OHA
  - http://www.openhandsetalliance.com
- Members
  - http://www.openhandsetalliance.com/oha_members.html
  - 47 technology and mobile companies
  - Mobile Phone Vendors (Handset Manufacturers)
  - Chip Vendors/Semiconductor Companies
  - Graphics Vendors
  - Telecommunication Companies/Mobile Operators
  - Software Companies
- Membership Based on “Give-Back”

Android Overview

Android Design Points

- Designed Originally For Cell Phones
- Design Goals:
  - Small Footprint
  - Power Management, Reducing Energy Consumption
  - Screen Resolution, Size and Management
- Platform Independence
- No Bluescreen of Death

Android Architecture Overview



Applications

Home
Contacts
Phone
Browser
...

Application Framework

Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
XMPP Service

Libraries

Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
libc

Android Runtime

Core Libraries
Dalvik Virtual Machine

Linux Kernel

Display Driver
Camera Driver
Bluetooth Driver
Flash Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Google Code Projects/Repositories

	projects	function
Core projects	bionic/*	libc, libm,libdl
	prebuilt/*	binaries to support Linux (gdbserver),toolchain
	kernel/*	Linux kernel
	hardware/*	hardware abstraction library(API)
	system/*	debugging/inspection tools (adb)
	bootable/*	boot reference code
	vendor/*	board level configure
	build/*	build system, makefiles(envsetup.mk, etc.)
	frameworks/*	framework libraries
	dalvik/*	dalvik VM
	development/*	development tools, SDK,sample
Externel projects	external/*	other open sources(jpeg, webkit, qemu)
Packages	packages/*	application(camera, calculator)

Android Components (1 of 2)

- Linux 2.6 Kernel plus 115 some Patches
  - Providing Basic system Functionality
    - Memory Management, Process Management, Security
    - Networking, Device Drivers, Multithreading
- Libraries
  - i.e. Bionic C-Library
    - Unique Component designed for Android
    - Only 200k (half the size of std libc)
  - Audio, Video Media Support
  - Graphics Support
  - Lightweight Database (Storage and Sharing of Data)
- Android Runtime
  - DalvikVM
    - Specifically Designed for Android
    - See Later Page
- Application Framework
  - Provides higher-level Services to Applications using Java Classes

Android Components (2 of 2)

- Applications, Android SDK
  - Standard Applications come with System
  - SDK to develop new applications
  - Applications can be written in Java using DalvikVM
    - Efficient, accommodates interoperability, Portability
  - C/C++ Applications can be run as well
  - Eclipse based Application Development Environment (with special plug-in)
- Key Capability: Sharing of Functionality Amongst Applications
  - Applications can export functionality
    - Promotes Software Re-Use
    - Consistent User experience
- As mentioned earlier:
  - All applications are equal – Standard ones are not special
- For more Information: “Mentor White Paper” and Android Website
  - Android Website, and
  - http://www.mentor.com/products/embedded_software/techpubs/getting-started-with-android-development-for-embedded-systems-51770

DalvikVM: Taking a Closer Look

- DalvikVM
  - Spefically Designed for Android
  - Based on an Open Source JavaVM and Class Libraries Called Harmony
  - Designed to meet mobile performance/size requirements
- Based on Harmony: Apache's open source Java SE http://harmony.apache.org
- Some unusual characteristics:
  - Dalvik is register based (JavaVM is stackbased)
  - Dalvik does not support JIT compilation (not yet)
  - Each application runs in separate VM Instance
  - Memory is shared between instances
  - Startup-time improvement: "zygote" pre-initialization of VM with fork
- Optimized for Multiple Instantiation – Applications carry private copy
- Designed for Memory Efficiency – Register Based rather than stack based
- Uses it's own bytecode implementation
- Makes full use of Linux Kernel

Licensing Examples Per Module

Open Source Module	**License**
Linux Kernel	GPL
NetBSD C Library	BSD
DBUS	GPL2
OpenBinder (core)	GPL2
YAFFS2	GPL
SQLite	GPL2
Webkit	BSD (including LGPL)
WebCore	LGPL
SDL	LGPL
SGL	Google(Skia)
OpenGL -ES	SGI OpenGL(BSD/MPL)

Newest Android Developments

- Versions: Some one has a sweet tooth...
  - Cupcake (Current)
  - Donut
  - Éclair
  - Flan
  - ....
- Latest Applications: Google pinyin
- Lots of new phones running Android
- Companies announcing Netbooks running Android

Positioning: Android versus ChromeOS

- Google Chrome OS is a new project, separate from Android.
- Android was designed from the beginning to work across a variety of devices from phones to set-top boxes to netbooks
- Google Chrome OS is being created for people who spend most of their time on the web, and is being designed to power computers ranging from small netbooks to full-size desktop systems.
- While there are areas where Google Chrome OS and Android overlap, “we” (Google) believe choice will drive innovation for the benefit of everyone, including Google.
- ChromeOS Source (for developers) not avail till late 2009
- First implementations to be seen late 2010

Android "Invasion"
Expanding into New Markets

Android Invasion: Target Market Expanding

Original Target: Handheld/Mobile Devices, cell Phones

Currently: Netbooks, STBs,

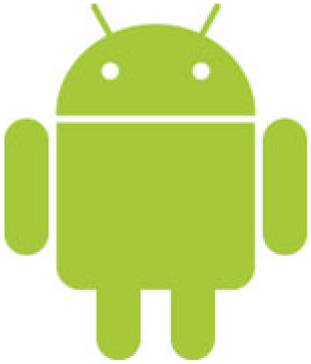
Near Future: The Sky is the limit: Car Computers (Automotive), Medical Instruments, Industrial Equipment/Control, Office Automation, Multimedia Player, Video Surveillance, Storage and Networking Appliance, Home automation, Instrumentation

Some Design Aspects Apply, Some Could Be Made to Apply

- As some design aspects from the mobile phone market may not apply, others may be used advantageously
- Energy/Battery saving functions in Android for example Could be of benefit
  - Household Vampire Energy Could be Reduced
  - Even though Standby energy on household items is usually low
    - It does add up
- Device Heat development Requires more cooling
  - Lower heat outputs will require less cooling, i.e A/C usage
- Combined with Power Architecture reduced energy requirements and heat development are further enhanced
- Android is designed for Small Footprint requiring less disk space less memory
  - Power Architecture Demo System uses an SD card only
- Android features would allow to build environmentally friendly household devices
  - such as STBs, Digital TVs, etc,
  - especially when using Power Architecture

What Makes Android Interesting
What Android Can do for Embedded
What Android Can do for “Open”

Agenda

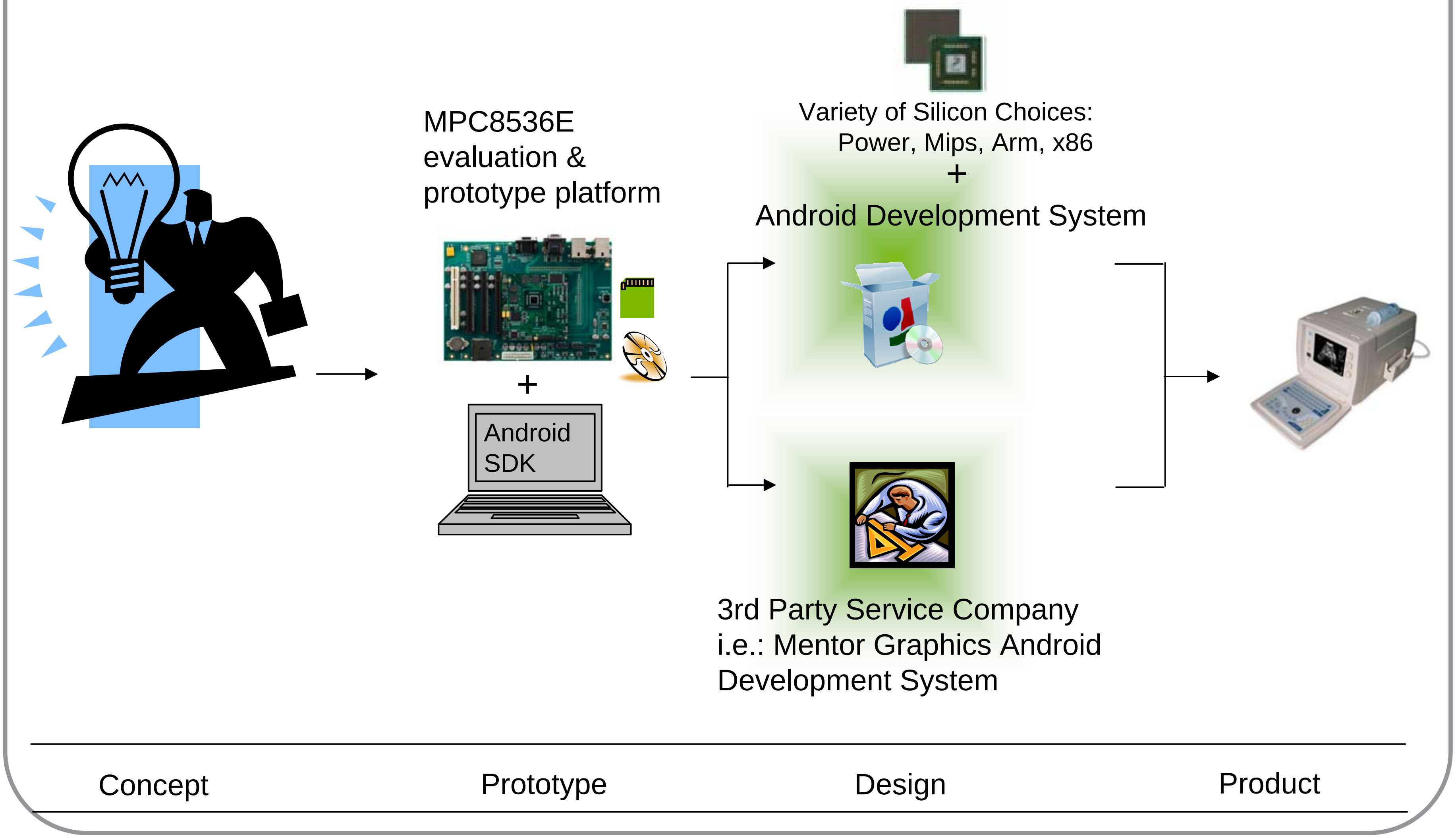


- What Makes Android Interesting
- What Android Can do for the Embedded Community
- What Android Can do for the Open Community
- Android is Disruptive Technology

What makes Android Interesting

- Aims to bring Internet-style innovation and openness to mobile/embedded equipment
- Provides a Rich development environment
  - Device emulator
  - Tools for debugging
  - Memory and performance profiling
  - Plugin for the Eclipse IDE
- Application framework enabling reuse and replacement of components
  - Dalvik Virtual machine optimized for mobile devices
  - Integrated browser
  - Optimized graphics
  - SQLite for structured data storage
  - Media support for common audio, video, and image formats
  - Communication support
- Enable customers to focus on
  - Target Device
  - Application Development
  - UI and other value add development
- Brings set of agreed upon conventions

Android Can Help Streamline the Embedded Design Processes



MPC8536E evaluation & prototype platform

+

Android SDK

Variety of Silicon Choices: Power, Mips, Arm, x86

+

Android Development System

3rd Party Service Company i.e.: Mentor Graphics Android Development System

Concept

Prototype

Design

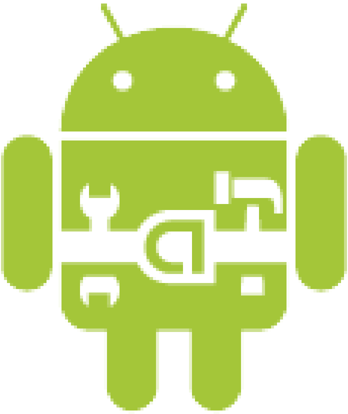
Product

What Android Brings to Openness

- Linux Based Open Source
- Adds set of “Conventions”
- Standard sets of libraries, tools etc
- Streamlined for embedded Device developer
- Easy and consistent to use for end-user
- No blue screen of death
- For the Embedded Space:
  - Android Could be to Linux what Linux was to Unix

Android is Disruptive Technology

- Could this threaten Windows?
- Initial Confrontation: Android vs WindowsCE
- Maybe AndroidPC will threaten Windows some day?
  - I.e. Android on netbooks
  - May make Linux more appealing to non-techies
    - Conventions and standards may make it less flexible for the seasoned Linux user
    - Conventions will help the novice “non-windows” device user
  - May offer an alternative to Windows
  - At least no blue-screen of Death
- The GOOGLE name behind it won’t hurt
- Google-Android sources freely available
- Android developers Community(ies) is/are growing



Porting Android to Power Architecture

Mentor Graphics on Android

Late July 2009 Glenn Perry, Mentor Graphics Embedded Systems Division GM, stated:

- "Mentor's strategy acknowledges two strong trends we see in embedded device development today. One is a huge demand for Google's Android platform in new, complex devices beyond the mobile phones for which Android was originally developed. The other is the growing use on multi-core processors of multiple operating systems, usually Linux and an RTOS like Nucleus."

Porting Agenda

- Porting Skills Needed for Each Component and Porting Considerations
- Components that need to be considered for Performance Improvements/Tuning
- Porting Steps and Phases - The Porting Process: Android Power Architecture
- The Result: Android on Power Architecture
- Demo: MPC8536E-Android Platform – The underlying hardware

Porting Skills Needed Per Component

Number	Subteams	Responsibilities	Skills
1	**System** (/bootable/*, /kernel/*, /hardware/*, /build/*, /prebuilt/*, /system/*,vendor/*)	1. bootloader porting 2. kernel, file system 3. drivers needed 4. HAL 5. cross-compiler	1. master bootloader 2. master drivers transpl. 3. familiar with linux kernel. 4. familiar with compiler 5. cooperation, communication
2	**Bionic** (/bionic/*)	1.port C runtime on PowerPC	1. master Power Arch. 2. master Assembly languaage 3. familiar with python. 4. know about kernel kno. 5. cooperation, communication
3	**App** (/dalvik/*, /external/*(not including qemu/*), /frameworks/*,/packages/*,)	1. dalvik transplant 2. frameworks modification	1. master C++, JAVA 2. cooperation, communication
4	**SDK** (/development/*, /externel/qemu/*)	1. offer a phone simulator kit based on Power	1. master Power Arch. 2. master C language 3. know about the Java language 4. cooperation, communication

First Porting Steps

- Integrating Android-specific Linux kernel patches into the Board’s BSP
- Porting the DalvikVM
  - including architecture and build support
  - Including optimization for Dalvik acceleration
- Extending bionic run-time library/linker support to accommodate Power Architecture
- Utilizing Power Architecture FPU and other performance enhancement across all software modules
- Integrating and testing board support and industry-specific device drivers, codecs, and other middleware
- Supporting Power Architecture technology in the Android Software Development Kit (SDK) and Android targets in the customizable Development System
- Platform and integration testing of Android stack components and shrink-wrap Android applications
- The Android port is based on the Open Handset Alliance (OHA) Android 1.5 ("Cupcake") release
- The initial Mentor Graphics toolkit supports the Freescale MPC8536E (PowerQUICC III) development system
  - "complemented by customer-tailored support from Mentor Graphics."

Porting Considerations per Component

- General
  - Port based on Android Cupcake Version
  - Approximately 4000 lines of code that need to be touched
- Linux Kernel
  - Kernel Patches need to be applied to BSP Software that comes with the board
  - Linux Kernel porting after that is pretty straight forward
  - Some Endianess issues travel all through Android though
- DalvikVM
  - DalvikVM's initial port can be done porting generic C code
  - CPU architecture dependent optimizations required for better performance
    - Requires Assembler Level Support of Architecture needed for full optimization
- Bionic is Androids C- Library
  - libc, libdl, linker, libm, libc++
  - OpenBSD/ NetBSD based
  - Has obviously lots of architecture/system specific areas
    - Responsible for shared library loading, process forking which all varies based on architecture
- Hardware Specific adjustments needed to leverage audio, video, graphics

Components to Consider For Optimization

- DalvikVM
- Bionic
- Webkit
- OpenGL-ES leveraging Hardware Acceleration
- H.264 and other hardware Acceleration

The Result: Android on Power Architecture

- Provides the latest stable Android for Power Architecture
  - Includes optimized Libraries for Architecture and CPU core
  - Includes optimized Dalvik VM for the Architecture and the CPU core
- Platform – Libraries, Kernel
  - Developing and adding board ports and custom drivers
  - Developing and adding product specific C/C++ Libraries
  - Enables customizing HAL for hardware specific needs
- Runtime – Dalvik, JNI
  - Developing and adding product specific JNI
- Application Framework
  - Developing and adding product specific classes
  - Enables customizing the UI, theme, and icons
- Application SDK and Emulator (Eclipse plug in)
  - Create product specific Application SDK
  - Enables customizing the emulator to match the product (hardware, behavior and skins)
- Supporting Android on additonal Power Architecture Cores/SoCs should be simple

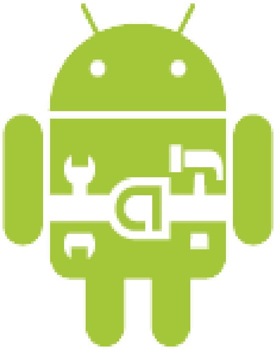
Next Steps

- What's still being Worked on
  - Bit Blitting Routine
  - Media applications and their optimization
  - More DalvikVM Optimization
  - Graphics Optimization
  - Use of Special Hardware Accelerators (i.e Media Streaming)
- Possible Next Steps
  - More Tuning/Testing (Target Device Specific, not just SoC Specific)
  - More Applications
  - Support for Multiple Cores
  - Supporting other Power Architecture Cores
  - Expand set of applications by responding to market specific requirements

Who Did the Port : Mentor’s Android Development System (1 of 2)

- Mentor Graphics – Development System for Android
  - Supplementing the MPC8536-Android system, Mentor Graphics offers a comprehensive development system for embedded Android developer.
  - Support for cross development of system-level and application software for devices running the Android platform
  - Up-to-date production kernels, run-time libraries and other core components
  - Optimized Dalvik Virtual Machine and JNI (Java Native Interface)
  - Android Application Framework and SDK (Eclipse Plug-in)
  - Customizable Android Device Emulator for Eclipse
- Mentor Graphics Embedded Target Development system
  - Development System seats available to license
  - Per project perpetual licenses (independent of user number)
  - Enterprise business models also available

Who Did the Port: Mentor’s Android Development System (2 of 2)



- Additional Mentor Graphics Services
  - Custom board ports
  - Extend Android I/O support with JNI and Class support
  - Integrate product/industry specific middleware
  - Turnkey solutions
  - Application development
  - Integration and Support services
- Contact Mentor Graphics for pricing and business model details
  - 800.468.6853 or 251.208.3400
  - www.mentor.com/products/embeddedsoftware

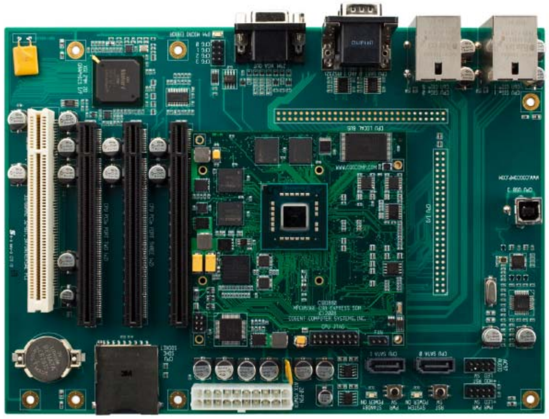
Demo

Freescale Power Architecture QorIQ and PowerQUICC III

- Previewed in 2008, the MPC8536 is billed by Freescale as a "highly integrated PowerQUICC III with advanced power management." The QorIQ SoC, meanwhile, is considered a next-generation heir to the PowerQUICC III line for high-end networking and other demanding embedded applications.
- Announced in June 2008, QorIQ is said to be pin- and software-compatible with PowerQUICC. Based on one to eight e500 cores clocked from 400MHz to 1.5GHz, QorIQ is fabricated with 45nm process technology, leading to greater claimed power efficiency. The SoC was expected to start sampling this summer.
- Stated Kamal Khouri, senior manager, Platform Product Management for Freescale's Networking and Multimedia Group, "Our collaboration with Mentor Graphics allows OEMs to build smart, feature-rich devices based on Freescale processors using either their own value-added applications or shrink-wrapped software from the Android Marketplace."

MPC8536E Android Evaluation and Prototype System

- **Hardware**
  - Based on MPC8536RDK Development System
- **Android Runtime Software**
  - Prebuilt Android runtime image
  - Optimized Android Virtual Machine (Dalvik) e500v2 on MPC8536 (later P1022)
  - Android Library (Bionic) support and optimized for e500v2
  - Linux 2.6.29 patched and configured for Android
  - Android version 1.5 (Cupcake)
  - VGA graphics mode up to 1600x1200
  - 2D graphics library, 3D graphics library based on OpenGL ES 1.0 specifications
  - Ethernet connectivity
  - Web browser, WebKit application framework.
  - Media support H.263, H.264, MPEG-4 SP, AMR, AAC,, MP3, WAV, JPEG, PNG, GIF, BMP
  - USB Mouse and keyboard
  - Demonstration Android applications
- **Android Build Software**
  - Source directory, make files, and tools to build Android



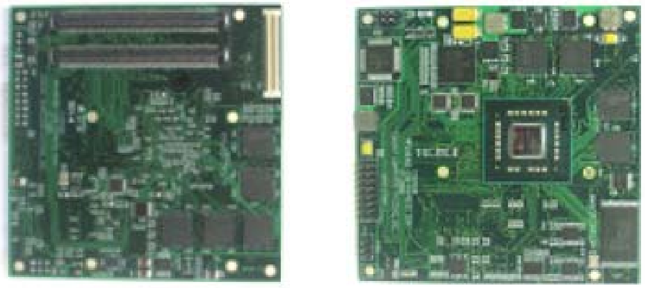
MPC8536E-Android Hardware Platform Features

CSB1880 CSB Based on COM Express Form Factor includes:

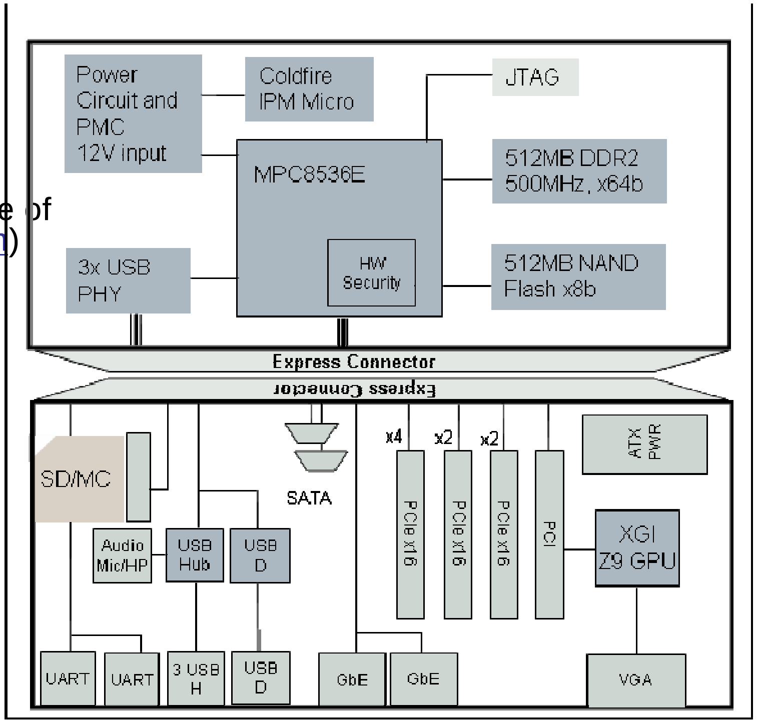
- 1.25 GHz MPC8536E processor
- 512 MB 64-bit DDR2-500 memory with ECC
- 512 MB NAND flash memory
- Bootable 4-bit SD/MMC Port (SDIO-compliant)
- Three PCI Express ports configurable
  - one x8, two x4 or one x4 and two x2 \
- Dual RGMII
- Three 480 Mb USB 2.0 ULPI PHYS
- Two serial ports and one I2C port
- Eight GPIO lines and four PCI interrupts
- Standard 16-pin JTAG
- Onboard MCF51QE32 IPM microcontroller for power sequencing, boot configuration, fan control and thermal monitoring
- CSB1880 module provides additional flexibility for direct purchase of the module from Cogent Computer Systems (www.cogcomp.com)

CSB1801 µATX Carrier Board includes:

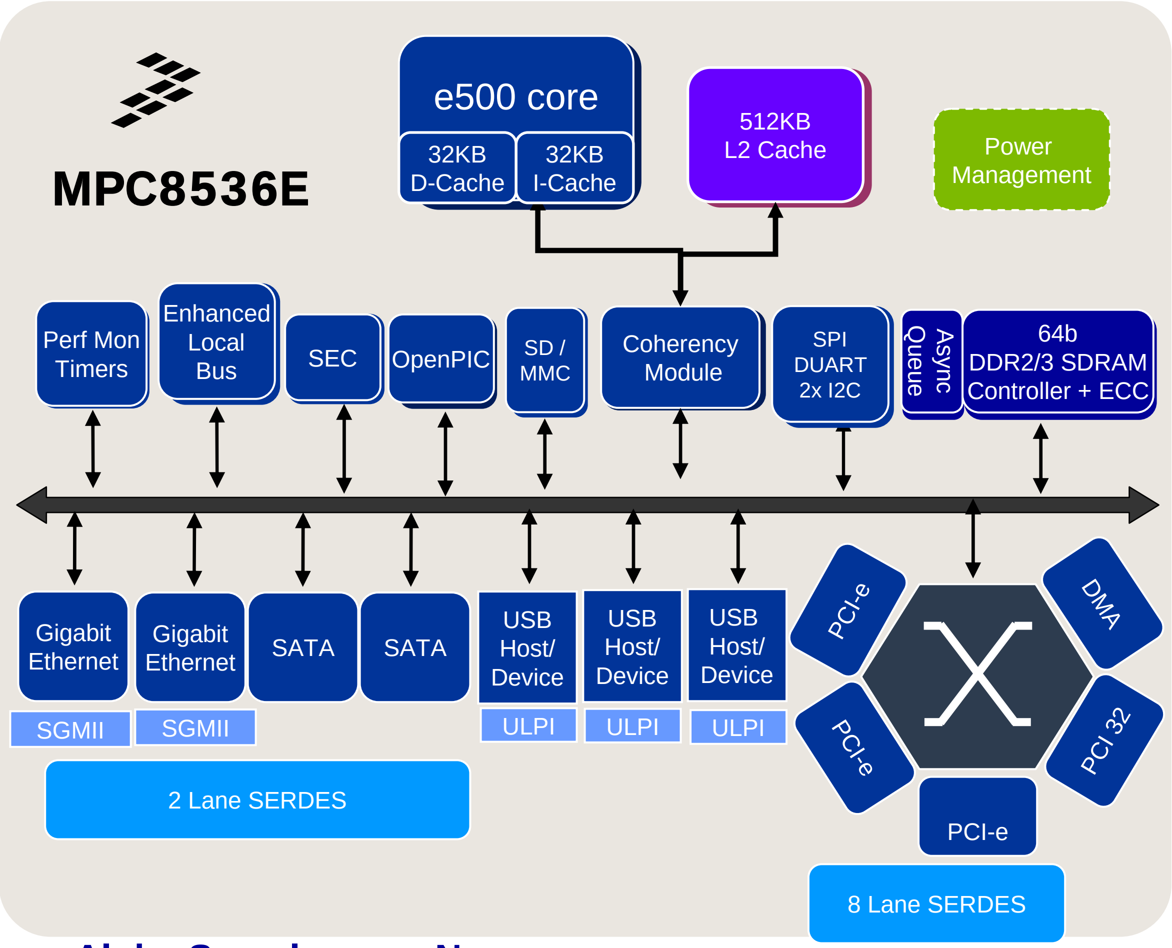
- Flex/MicroATX form factor
- High-speed 440-pin CSB Express connector
- Three x16 PCI Express connectors (x4 + x2 + x2)
- 32-bit 3.3V PCI edge connector
- Dual SATA connectors
- USB-B connector (from SOM USB 2)
- Dual 10/100/1000 Ethernet
- Four-port USB host interface
- AC97 header for headphone and mic
- XGI Z9M 2D graphics controller with 8 MB frame buffer



3.74“W x 3.74“W



MPC8536E Block Diagram



- **e500v2 PowerPC core up to 1.5 GHz**
  - 512KB L2 Cache w/ECC
  - 36 bit physical addressing
  - DP-FPU, SPE
- **System Unit**
  - 64/32b DDR2/3 up to 667 MHz data rate w/ECC
  - Integrated Security Engine
  - Triple USB 2.0 High Speed Host/Device
  - SD/MMC Flash Interfaces
  - SPI & Enhanced Local Bus
  - Dual 10/100/1000 Ethernet Controllers
  - High Speed Interconnect
    - Triple PCI Express
    - PCI
  - Dual SATA controller
- **Advanced Power Management Controller**
  - Nap, Doze, Sleep,
    Deep Sleep (@0.95W @ 35C)
  - Power off to core and cache
  - Wake on LAN,/USB/GPIO/timer/external signal
  - Isolated Power Planes
- **Low Power Dynamic Operation**
  (typical power estimate)
  - CPU/Platform
  - 800/400 – @ 4.5W
  - 1000/500 – @ 5.2W
  - 1500/500 – @ 6.7W

- **Alpha Samples: Now**
- **Production: Now**

Key Features

- Low Heat Development Based on Power Architecture Core
- No Fan Required
- Whole System on SD Card – no hard drive with spinning heads needed
  - uboot + OS + Libraries + Applications
- Only Small Power Supply needed
- GPU integrated into main board
- Small Form Factor
  - Can be much smaller than what’s shown today

Thank You!

Questions?

Power.ORG ™ Grenobles, ELC 2009