ARM The Architecture for Android

Tom Wang
Manager, ARM Marketing
Android is developed for, and is exclusively shipping on ARM

Android is heavily optimized for ARM
- 1,200 files of assembler code.
- 20,000 lines, or 28% of Bionic libraries are ARM assembler
- WebM with 11,000 lines of ARM NEON assembler
- 6,300 C & 4,900 C++ files many with ARM optimizations
- 268 organizations contributing to the ARM Android codebase

Android Marketplace development targets Dalvik and Native ARM code
- Marketplace now exceeds 100K apps with more than 1.2B downloads
Continued ARM Momentum on Android

Demonstrating continued Google Investment in ARM

- V8 Javascript engine for ARM
  - World’s **fastest** mobile browser
- ARM Dalvik JIT support
  - **2-5x** performance increase
- Complete ARMv7 Support
  - ARMv7 native apps & debug
  - ARM Thumb®-2 code generation
  - ARM NEON™ SIMD instruction support
- Support for pure native apps

Google source releases include all ARM enhancements

260+ ARM Powered Android Handsets
Driven by market opportunity and robust partner ecosystem

- 50+ phones running Android applications
- 70+ members of the ARM Connected Community™ working on Android solutions

Mobile Computing

Connected Appliances/
General Embedded

Washing Machine &
Microwave Oven

Home
Digital TV’s
Digital Media Adapters
Set Top Boxes
Android Applications Growing Rapidly

- ARM-based Android Devices run rate 100,000 per day
- 180,000 Developers
- 50,000 Android Apps
- Android App Store is 57% free

Android Enabling Consumer Devices

- Complete and ready to deploy across multiple platforms
- Extensible by developers and users
- Insulates applications from rapidly evolving open source
  - Brings cost savings
  - Quicker time to market
- Apps stores easily deployed using NDK & Dalvik
- ARM optimizing the Linux Kernel for 12 years
- ARM working with CodeSourcery on GNU Tools (GCC 4.4)
Google Delivers Another Tasty Treat for ARM

- Support for pure native ARM applications
  - Designed to deliver compelling apps
- WebM (VP8) codec optimized for ARM NEON with 11,000 lines of assembler code
- Further ARM optimizations for Dalvik
  - Focus on garbage collection
- Further ARM optimizations for V8 javascript engine
  - Google now working on delivering a further 50% improvement with “Crankshaft” optimized for ARM
Only ARM Passes CTS & Run all applications

- As per Google Android CTS document
  - Device implementations MUST accurately report the native Application Binary. Interface (ABI) supported by the device, via the android.os.Build.CPU_ABI API. The ABI MUST be one of the entries documented in the latest version of the Android NDK, in the file docs/CPU-ARCH-ABIS.txt. Note that additional releases of the Android NDK may introduce support for additional ABIs.

- Supported production ABIs
  - ARM V5
  - ARM V7A

- The value of Android is destroyed
  - If you do not use the latest highly optimized and tested version
  - Can not run all the latest and greatest applications
What is Android?

- Android codebase is large and valuable
- Is it Android if you just use some components?
- Do not underestimate the benefits of the SDK and NDK tools

- Powerful applications focused on Smartphones
- Valuable libraries including Webkit
- Android Kernel very efficient and mainly Apache 2
- Dalvik VM to sandbox apps
- Application Framework, mobile centric
- Apps designed for mobile HID & UI
# Know Your Android Device Categories

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cat 1</th>
<th>Cat 2</th>
<th>Cat 3</th>
<th>Cat 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worked with Google directly on device</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allowed to use “Google” in branding</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Applications (with agreement)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access to app store (with agreement)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Passes compliance test suite</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Android OS phone</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
# Best Fit For Device

<table>
<thead>
<tr>
<th>Chrome or Android</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Smartphone</strong></td>
<td>Android is designed for Smartphones with full mobile phone features</td>
</tr>
<tr>
<td><strong>Netbooks &amp; Smartbooks</strong></td>
<td>Chrome is the long term as Chrome OS is designed for net books and has a multi windowed UI. Android in the short term</td>
</tr>
<tr>
<td><strong>Tablets</strong></td>
<td>Android Tablet to be launched Q1 2011? Chrome tablet plans unclear</td>
</tr>
<tr>
<td><strong>PMP</strong></td>
<td>PND’s &amp; PMP’s fundamentally has the same physical UI as a smartphone</td>
</tr>
<tr>
<td><strong>Consumer</strong></td>
<td>Android in the short term. Google TV in the longer term. Chrome maybe used in as well.</td>
</tr>
<tr>
<td><strong>Other Devices</strong></td>
<td>Depends on device UI and memory requirements.</td>
</tr>
</tbody>
</table>
Google Making Money out of Android

- How does Google make money out of Android?
  - $0 too good to be true?
- Google is an advertising company, the more they know about you, the more targeted and more valuable the adverts.
- What does Google know about an Android user?
  - Your location, what you have searched for, contents of your gmail
- Targeted adverts via applications and location based search
- Google also makes money from the marketplace
- Google projected to make anything from $1 to $10 per month per user
  - Revenue share model in place with Operators
- Google wants to ensure that Android reaches the widest number of eyeballs worldwide
Working with Android

- Google cares about volume
  - Has designs with every major OEM except Apple, RIM, & Nokia
- Android has a unique engagement model
  - Works with a launch OEM and Silicon partner
    - In some rare instances they may have two
  - Codebase and features kept private
  - Once “launch device” is made available, then the source code is released
    - Normally a delay as code is cleaned up
- Android works on two releases per year
  - “Dads & Grad” release
  - Winter Holiday release
    - Only one OEM will meet the holidays with this release
    - Releases are fixed in time, not in features
- Android team is not set up to engage with a wide customer base
  - Unless you are one of the select few
  - Best way to establish contact is via Google Groups
- Android roadmap limited to a 6 month view
2.3 Release About Enabling ARM Apps

- NDK is a comprehensive tool kit to enable application developers to write directly for the ARM processor
  - Available exclusively for ARM
- 2.3 (Gingerbread) NDKr5 highlights
  - New NativeActivity feature removes requirement to use any Java
  - EGL library to create and manage OpenGL ES textures and services
  - Addition of default C++ STL implementation
  - Updated toolchain, based on GCC 4.4.3
  - New API’s
    - Input subsystem, sensor data
    - Windows, surface subsystem
    - Audio API’s based on OpenSL ES
    - Direct access to graphics assets in an .apk

Android applications can be written in Java, native ARM code, or a combination of the two.
NativeActivity for Pure Native Apps

- New class called NativeActivity enables creation of strictly C and ASM level applications
  - Moves “main()” from Java to C
  - Apps still run in a Dalvik sandbox, but **no Java code required**

- Two methods
  - Modify the NativeActivity class
    - Full flexibility of main app thread control
  - Use native app glue helper library built on top of NativeActivity class
    - Provides ready made infrastructure for threading blocking behavior

---

Example main function written for native code using NDKr5

```c
void android_main(struct android_app* state) {
    // Struct unique to this app, includes callback function pointers
    struct engine engine;

    // Make sure glue isn't stripped.
    app_dummy();

    // Setup state
    state->userData = &engine;
    state->onAppExit = engine_handle_end;
    state->onInputEvent = engine_handle_input;
    state->app = state;

    // Loop waiting for stuff to do.
    while (1) {
        // Application behavior, event processing
        
    }
}
```
Know Your Android Versions

Cupcake: Android 1.5 Released Q2 2009

Donut: Android 1.6 Released Q3 2009

Éclair: Android 2.1 Released Q1 2010
Android 2.0 was only ever released on the Motorola Droid

FroYo (Frozen Yogurt): Android 2.2 Released Q2 2010

Gingerbread: Android 2.3 Released Q4 2010
Rapid Transition To Latest Android Versions

- Market is moving rapidly to the latest V7A versions of Android
- Platform API Level Distribution
  - Android 1.5 6.3%
  - Android 1.6 10.6%
  - Android 2.1 39.6%
  - Android 2.2 43.4%
  - Data collected during two weeks ending on December 1, 2010
- Historical Distribution
  - Last historical dataset collected during two weeks ending on December 1, 2010
Android Roadmap

**Éclair (2.0/2.1)**
ARM V6/V7
- NEON Skia 2D acceleration
- GLES 2.0 support via ARM native code
- Browser support for HTML5

**FroYo (2.2)**
May 2010 Release
ARM V6/V7
- New multimedia framework
  - Flash 10.1
  - Dalvik T2 JIT
  - V8 Javascript
  - Full V7 Native Development kit for apps

**Gingerbread**
Released Dec 2010
- Extra Large Screen support
- NDKr5 enhanced with full native apps
  - VP8/WebM
  - Full transition to Stagefright
    - Dalvik optimizations
  - V8 JavaScript optimizations
  - 2.6.35 Linux kernel

**“HoneyComb”**
Future Release*
Expected 2011
- SMP as primary platform target
- UI enhancements for larger screens

*Listed features for unreleased versions represent current intentions and are subject to change.
Android Architecture Evolution on ARM

1.5x Javascript Improvement

Cortex A5, Cortex A8, Cortex A9, QC Snapdragon

ARM9

VFPv2

Jazelle®

ARM11

TrustZone™

SIMD

VFPv3

NEON™ Adv SIMD

Thumb®-2

SMP Support

Improved Media (2xV5)

Scalable Multi Processor Support

Improved Media and DSP (2xV6)

Improved code density, approx 30%

ARM9

ARM11

V5

V6

V7A

Improved Media (2xV6)

The Architecture for the Digital World®
Android is ARM Highly Optimized

- Dalvik JIT Support
- 1,200 files of ARM Assembler
- NDK
- Ogg Theora
- WebM 11,000 lines of NEON
- ARM Native Applications
- NDK Tools
- V8 Javascript & Crankshaft
- Chrome Browser
- WebKit Browser
- Only ARM access to CTS

Google is investing significant effort to optimize both Android and Chrome for the ARM architecture
Linaro Providing the Basis for Android

“To make open source development easier by delivering a common software foundation, across multiple distributions and vertical segments, deployed by the industry's leading OEM, MNO and Semiconductor companies”

(Linaro™ is a Not For Profit UK company)
How Linaro Will Benefit Android

- Linaro works with upstream projects
- Linaro creates a validated kernel and toolchain supported by multiple chip companies
- Linaro will provide V7 optimized Kernels and BSP’s
- Linaro will provide high quality tools for Android
- Significant core
Java and JavaScript Optimizations

- Dalvik Java Virtual Machine
  - Interpreter support for ARMv7T
    - Instructions/co-procs for interpreter
      - UXFB, TBB... VFP, NEON
  - Tracing JIT
    - Focus on hot code paths
    - Increases performance by 2-5x
    - Minimal code bloat approx 100K
    - Thumb or Thumb-2 code generation
    - Support for VFP or Soft FP

- V8 JavaScript VM
  - Enabling true Web 2.0 apps on ARM-based devices
  - VERY fast pure JIT generating ARM code
  - ARM and open source community driving performance higher 60% improvement
ARM and Google working on V8

- V8 is a pure JIT, no interpreter, no intermediate code
- Not one JIT but many JITs - one for each JavaScript construct
- Intel codebase mature and contains optimizers for many JITs
  - ARM codebase has some way to go
  - Opportunity to converge on Intel over the next 6-9 months
- ARM works directly with Google V8 team
  - ARM’s 2 man team have extensive knowledge of JIT code generators
  - Small regular incremental optimization patches to the V8 codebase
  - Bi-Weekly calls with between ARM and Google V8 Team
    - No extensive roadmap for optimization

\[
\text{REPEAT} \\
\text{profile; find_bottleneck; optimize} \\
\text{UNTIL law_of_diminishing_returns == True}
\]
ARM contributed first V8 patch in March 2010
   - 15% improvement

ARM and Google *together* have converged on Intel in <5 months clock-for-clock
   - *Both* Intel and ARM now complete the V8 suite in < 2 Secs (1Ghz A9/1.6Ghz N450)

New register allocators will increase performance a little (maybe 5%)
   - Intel has this today
   - Google and ARM adding register allocators for ARM V8 *now*

**Now 50% faster than V8 in Android Froyo**

Google now working on Crankshaft
   - Another 50% uplift on V8
   - Work will be done on ARM & x86
     - No other architectures
ARM Native Code and GPU Support

- **ARM optimized Skia**
  - Provides up to 30% performance uplift on browser use-cases
  - NEON intrinsics (and assembler)
  - Available in Skia Open Source codebase

- **Enhanced NDK r4b Support for**
  - ARM Thumb®-2
  - VFP hardware FPU instructions
  - ARM NEON™ SIMD Engine
  - OpenGL® ES 2.0 (ARM Mali GPU Support)

Examples of applications that leverage the NDK and optimized Skia
ARM Native Code Applications

- Increasing number of applications now run native ARM code
- Native code required for performance sensitive applications, including
  - 3D Graphics using OpenGL ES 2.0
  - Multimedia
  - Browsers
  - Games
  - VoIP
NEON Support In Android

- ARM optimized Skia
  - Provides up to 30% performance uplift on browser use-cases
  - NEON intrinsics (and assembler)
  - Available in Skia Open Source codebase

- NEON optimized PixelFlinger
  - Android S/W renderer
  - 2x uplift in performance

- NEON Optimized codecs
  - Work sponsored by Google
  - OGG Theora
  - VP8
WebM Open Media Project

- WebM (VP8) Codec now in Android 2.3
- Launched May of this year to address lack of open video standard for the web, WebM includes:
  - VP8 video codec licensed under a BSD-style, royalty-free license
  - The Vorbis open source audio codec
  - A container format based on a subset of the Matroska media container
- Launched with 11,000 lines of ARM NEON optimized code

Source: ARM benchmarking of opensource libvpx codec running on 400MHz Cortex-A9 testchip with 200MHz DDR
Android SMP Support

- ARM is working with its partners to enable SMP support on Android
  - Validating SMP safeness
  - Investigating SMP optimizations
  - Changes have been checked into Android codebase

- Android is ideally suited for SMP
  - Linux Kernel is already SMP aware
  - Dalvik applications each have their own thread

- SMP Android already demonstrated by Nvidia and STE:
  - Nvidia reference tablet design
  - U8500 evaluation board
Adobe Flash and AIR Support in Froyo

- Adobe Flash 10.1 Public Beta in Froyo
  - Enhanced ActionScript Engine
  - Enhanced graphics renderer OpenGL ES 2.0

- Adobe AIR Developer Release in Froyo
  - Rich Internet Applications
Who Is Behind Android Optimizations on ARM?

- Some of the companies behind Android on ARM and the supporting technologies
Planning for Future Android Roadmaps

- High end Android devices are easy to plan for
  - Follow the high end Cortex A roadmap
  - 1GHz A8 this year
  - 2x 700MHz A9 2011
  - 2x Eagle 2012
  - Multimedia is HD video and GLES 2.0
    - VP8 support would be a bonus
- Low end devices are harder to plan for
  - Ensure that you have the right performance to support Android
  - Ensure that you can follow the roadmap features
  - Ensure that you can keep the BOM low
  - But there is no roadmap to plan against
Cortex A5 Benefits Over ARM 11

- 15% per MHz uplift over ARM 11
- ARM 9 power and size
- 3x Memcopy performance increase over ARM 11
- Thumb 2 enables 30% code density improvements
  - Minimal performance impact
  - Flash memory BOM saving for OEM’s
- Software compatibility with high end v7A Smartphones
  - Access to great V7A optimized S/W library
- SMP provides performance scalability
  - Ability to offer parts to OEM’s with different performance profiles

2011 ARM11 designs cannot compete against Cortex A5 designs
Cortex A5 Benefits for Android

- Access to the latest and greatest Android optimizations
  - Not all optimizations are being done for V5/V6
  - SKIA UI acceleration
  - Next generation V8 optimizations
  - Next generation Dalvik optimizations
  - VP8 & Ogg Theora
- Thumb 2 code compression
  - Helps save that valuable Flash and DDR usage
- Improved memcopy memory throughput
  - 3x over ARM 11
  - Memory B/W caused bottleneck in ARM 11 designs
- Plus power and performance benefits of Cortex A5
Cortex-A5: Delivering A8-class Performance for Entry-Level Smartphones

Web browser performance

(page load time - smaller is better)

(RTL, scaled to 600MHz)

- Cortex-A5 at 600MHz delivers comparable performance to 2010 smartphones (Cortex-A8 1GHz)
## Entry Level Solutions For Android

<table>
<thead>
<tr>
<th></th>
<th>Android 2.1</th>
<th>Android 2.2</th>
<th>Ginger Bread</th>
<th>Honey Comb</th>
<th>Screen Res</th>
<th>Flash Player 10.1</th>
<th>Ogg &amp; VP8/WebM</th>
<th>3D Gaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM11 no VFP, 0 L2</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>480 x 320</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ARM11 no VFP, 0 L2</td>
<td>Yes</td>
<td>Yes</td>
<td>?</td>
<td>?</td>
<td>480 x 320</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ARM11 VFP, 128K L2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (S/W UI)</td>
<td>Yes (S/W UI)</td>
<td>480 x 320</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cortex A5 VFP, 128K L2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (S/W UI)</td>
<td>Yes (S/W UI)</td>
<td>480 x 320</td>
<td>Yes (S/W) based</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Cortex A5 VFP, 128K L2, NEON, GLES 2.0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, Full UI</td>
<td>Yes, Full UI</td>
<td>848 x 480</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
SCA Now 80+ and Growing

- The ARM Solution Center for Android continues to grow the ecosystem of content and services
- Now over 80 participating SCA companies sharing their expertise, solutions and services:
  - Resources for building devices
  - Development tools
  - Porting guides
  - White papers
  - Android training

Expect more from the SCA in 2H2010

www.arm.com/solution-center-android
Android has a wide range of resources on-line

Source code:

Porting Developer’s Guide
- Great resource for porting Android to a new SoC or Platform

Compatibility Test Suite
- Great resource for helping you determine if your device is going to be able to access the Android Market
- CTS Document is critical read if you are interested in Android

Support
- Google does not provide direct support unless you are a handpicked launch partner.
- Best route to support is via the discussions groups, please see