ARM® update
www.arm.com

Spring 2015
Robert Boys
bob.boys@arm.com
Development Solutions Group, ARM

V 7.1
Agenda

- Roadmap
- Architectures
- What is NEW !
  - Instruction Sets
  - big.LITTLE™
  - What’s New !
  - Cortex®-M features
  - SIMD & NEON™
  - Linaro
  - Debug and Trace
The Cortex Processor Roadmap in 2008

ARM Cortex
“Intelligent Computing”

Performance, Functionality

Future

ARM7
ARM7TDMI
ARM9
ARM11
Cortex-M1
Cortex-M3
SC300
Cortex-R4
Cortex-R4F
Cortex-A8
Cortex-A9

Application
Real-time
Microcontroller
ARM 2015 Processor Roadmap

ARM Cortex
“Intelligent Computing”

Up to 3+ GHz
Cortex-A72
V8-A (64 Bit)
Cortex-A57 Cortex-A53 Cortex-A17

Up to 2.5 GHz
Cortex-A15

Up to 2 GHz
Cortex-A9 (MPCore)

~600 to 1 GHz
Cortex-A8
Cortex-A5
Cortex-A7

200+ MHz
Cortex-R4F
Cortex-R4
Cortex-R5
Cortex-R4

200+ MHz
Cortex-M4
Cortex-M5
Cortex-M6
Cortex-M7

72 – 150 + MHz
Cortex-M3
Cortex-M1
Cortex-M0

50 MHz
Cortex-M0 SC300

MMU
No MMU

ARM11
MPCore
ARM926EJ-S
ARM11
MPCore
ARM9
ARM7TDMI
ARM7

ARM7, 9, 11
Application
Real-time
Microcontroller

Not to scale

Performance, Functionality

Future

KEIL Tools by ARM

The Architecture for the Digital World®
Versions, cores and architectures?

- What is the difference between ARM7™ and ARMv7?
- Search for ARM architecture on Wikipedia to get the full list.

<table>
<thead>
<tr>
<th>Family</th>
<th>Architecture</th>
<th>Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM7TDMI</td>
<td>ARMv4T</td>
<td>ARM7TDMI(S)</td>
</tr>
<tr>
<td>ARM9 ARM9E</td>
<td>ARMv5TE(J)</td>
<td>ARM926EJ-S, ARM966E-S</td>
</tr>
<tr>
<td>ARM11</td>
<td>ARMv6 (T2)</td>
<td>ARM1136(F), 1156T2(F)-S, 1176JZ(F), ARM11 MPCore™</td>
</tr>
<tr>
<td>Cortex-A</td>
<td>ARMv7-A</td>
<td>Cortex-A5, A7, A8, A9, A12, A15, A17</td>
</tr>
<tr>
<td>Cortex-R</td>
<td>ARMv7-R</td>
<td>Cortex-R4(F), Cortex-R5, R7</td>
</tr>
<tr>
<td>Cortex-M</td>
<td>ARMv7-M</td>
<td>Cortex-M3, M4, M7 (M7 is ARMv7-ME)</td>
</tr>
<tr>
<td></td>
<td>ARMv6-M</td>
<td>Cortex-M1, M0, M0+</td>
</tr>
<tr>
<td><strong>NEW!</strong></td>
<td>ARMv8-R</td>
<td>32 Bit: Cortex-Rx</td>
</tr>
<tr>
<td><strong>NEW!</strong></td>
<td>ARMv8-A</td>
<td>64 Bit: Cortex-A53/57, Cortex-A72</td>
</tr>
</tbody>
</table>
Processor Licenses  (list is not complete or up to date)

- **ARMv8-A**: 27! AMD, Broadcom, Huawei (HiSilicon), STMicroelectronics, Samsung, MediaTek, Huawei, Altera, Qualcomm and Rockchip.
- **Cortex-A15**: Broadcom, HiSilicon, Texas Instruments, Samsung, nVIDIA
- **Cortex-A9**: Xilinx, Altera, NEC, nVIDIA, STMicroelectronics, Toshiba, Broadcom Corporation, Freescale, NEC, Texas Instruments, Toshiba, Mindspeed Technologies, ZiLABS, Open-Silicon, eSilicon
- **Cortex-A8**: Broadcom Corporation, Freescale, Panasonic, Samsung, STMicroelectronics, Texas Instruments, PMC-Sierra, Matsushita
- **Cortex-A7**: Broadcom, Freescale, Fujitsu, HiSilicon, LGE, Samsung, STEricsson, Texas Instruments
- **Cortex-A5**: AMD, Atmel, Freescale, Cambridge Silicon Radio, Open-Silicon, eSilicon
- **Cortex-R**: Broadcom, Texas Instruments, Toshiba, Infineon, Open-Silicon, eSilicon, Samsung, Marvell, LSI, Fujitsu, Spansion
- **NEW! Cortex-M7**: Freescale, Atmel, ST.
- **Cortex-M4**: Freescale, NXP, Atmel, ST, Texas Instruments, Open-Silicon, eSilicon, Spansion
- **Cortex-M3**: Microsemi (Actel), Broadcom, Energy Micro, NXP, ST, TI, Toshiba, Zilog, Accent Srl, Broadcom Corporation, Cypress Semiconductor, Ember, Fuzhou Rockchip Electronics CO. Ltd., Open-Silicon, eSilicon, Spansion (Fujitsu)
- **Cortex-M0**: Austriamicrosystems, Chungbuk Technopark, NXP, Triad Semiconductor, Melfas, Open-Silicon, eSilicon, Cypress, Infineon, Nuvoton, STMicroelectronics
- **Cortex-M0+**: Freescale, NXP, Atmel, Spansion, Silicon Labs

**Total:** Cortex-A 179, Cortex-R 45, Cortex-M 240, ARM7 172, ARM9 271, ARM11 82
Feature Set of Various ARM Processors

- ARM 9
- ARM 11
- Cortex-M,R,A
- 64 Bit
Instruction Sets

- ARM (32 bit) now referred as AArch32
- Thumb® (16 bit)
- Thumb2: Cortex-Mx processors. Cortex-R, A have Thumb2 + ARM.
- A64 (64 bit) referred as AArch64
Other ARM Instruction Sets:

- NEON™ (aka Advanced SIMD)
- Cryptographic
- DSP
- Floating Point
- Jazelle® (deprecated)
big.LITTLE™

- Two or more processors – share workload and save power.
- Processors not always the same.
- Cortex-A15 & Cortex-A7 are the original configuration.
- Cortex-A57 & A15 or A17. Cortex-A72 & A53 or dual A53.
- Freescale: Vybrid A5 & M4
- NXP LPC4300 M4 & M0
- Plenty of others now and more coming…
What is New?

- Cortex-M7 and Cortex-A72 announced. ARM v8-R
- More low cost Cortex-M boards: $12 to $20 with debugger.
- Samsung Exynos 7: A57 + A53, Exynos 5 Octa: 4 A15 + 4 A7
- Raspberry Pi 2 (4 Cortex-A7 900MHz)
- MediaTek Cortex-A72 (have chips)
- Cavium ThunderX: ARMv8 48 cores @ 2.5 GHz.
- Qualcomm Snapdragon 618 & 620: 4 A72 + 4 A53
- Xilinx Zynq UltraScale+: 4 Cortex-A53, 2 Cortex-R5nx
- Altera Stratix10SOC 4 Cortex-A53
- HP Moonshot using X-Gene ARMv8
- HiKey 96Boards: 8 Cortex-A53 1.2 GHz. $129. www.96boards.org
Cortex-M Adds:

- Simpler register set.
- More interrupts and exceptions. NVIC
- More breakpoints (to 6)
- 4 Watchpoints.
- SYSTICK timer for RTOSs.
- Fixed address space.
- Bit manipulation.
- Upgrade and downgrade paths.
Cortex-M Exceptions

- Exception handling order is defined by programmable priority
  - Reset, Non Maskable Interrupt (NMI) and Hard Fault have predefined pre-emption.
  - NVIC catches exceptions and pre-empts current task based on priority
  - Program Counter set to exception address in vector table which directs to handler code

<table>
<thead>
<tr>
<th>Exception</th>
<th>Name</th>
<th>Priority</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reset</td>
<td>-3 (Highest)</td>
<td>Reset</td>
</tr>
<tr>
<td>2</td>
<td>NMI</td>
<td>-2</td>
<td>Non-Maskable Interrupt</td>
</tr>
<tr>
<td>3</td>
<td>Hard Fault</td>
<td>-1</td>
<td>Default fault if other handler not implemented</td>
</tr>
<tr>
<td>4</td>
<td>MemManage Fault</td>
<td>Programmable</td>
<td>MPU violation or access to illegal locations</td>
</tr>
<tr>
<td>5</td>
<td>Bus Fault</td>
<td>Programmable</td>
<td>Fault if AHB interface receives error</td>
</tr>
<tr>
<td>6</td>
<td>Usage Fault</td>
<td>Programmable</td>
<td>Exceptions due to program errors</td>
</tr>
<tr>
<td>11</td>
<td>SVCall</td>
<td>Programmable</td>
<td>System SerVice call</td>
</tr>
<tr>
<td>12</td>
<td>Debug Monitor</td>
<td>Programmable</td>
<td>Break points, watch points, external debug</td>
</tr>
<tr>
<td>14</td>
<td>PendSV</td>
<td>Programmable</td>
<td>Pendable SerVice request for System Device</td>
</tr>
<tr>
<td>15</td>
<td>Systick</td>
<td>Programmable</td>
<td>System Tick Timer</td>
</tr>
<tr>
<td>16</td>
<td>Interrupt #0</td>
<td>Programmable</td>
<td>External Interrupt #0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>255</td>
<td>Interrupt #239</td>
<td>Programmable</td>
<td>External Interrupt #239</td>
</tr>
</tbody>
</table>
SIMD – Single Instruction Multiple Data

- SIMD is a set of instructions that can operate on multiple data sets contained in a register.
- Registers used are the 32 bit general purpose R in ARM.
- Some of the SIMD instructions start with a “S” (signed) or a “U” (unsigned) and with a suffix denoting the size of the operand (I16). An example is UADDI16.

Here is the SADD16 instruction:

Two sets of 16 bit operands are taken from 32 bit registers R3 and R0 and are added together (signed) and the 16 bit result stored in the 32 bit register R1:
NEON™:

- Up to 128 bit registers.
- Granularity down to 8 bits.
- Here are some examples of various data sizes displayed in ARM DS-5.
NEON: is also called Advanced SIMD:

- NEON is both a) another set of more instructions that operate on b) 32 special 64 bit registers.
- NEON works on a 128-bit data path. So on the Cortex-A9, it normally uses two of the 64-bit NEON registers for each of the operands and puts the result back in one or two of them.
- NEON share many instructions with a FPU…and each have their own instructions.
- NEON instructions begin with a “V”.
- VADD, VABS, VCGE, VCGT, VEOR, VQADD…
- See ARM appnote DHT0002A “Introducing Neon”.
- Cortex-A8, A9, A15, A64, some ARM11 devices.
How to program SIMD and NEON:

- Programs are written using assembly language.
- You can also use Intrinsics.
- The compiler can also use automatic vectorization on C or C++ code. (you have to tell the compiler it can do this)
- NEON libraries: This is a good way to do this to avoid writing in assembly. OpenMax and Ne10 are two. The executable code will use SIMD and NEON whenever possible.
- See DHT0004A for more information.
- At this time, all ARM Cortex-A series processors have NEON.
- SIMD and NEON used in Video encode/decode, 2D/3D graphics, audio processing...
- Cortex-A8, A9, A15, A64, some ARM11 devices.
Mali™ Graphics:

- Graphics Processing Units (GPUs)
- Part of the silicon…not a separate chip.
- There are several versions of Mali.
- Samsung Galaxy S uses Mali plus many more products…
have proven software ready before new hardware technology

The Linaro 15.02 release is now available for download!

Linaro Stable Kernel - Linux 3.10 Download

Latest Toolchain - Linaro GCC 4.8 (stable) Download
Operating Systems

- **Linux**
- WinCE
- Android
- Micrium, ExpressLogic, Quadros, QNX and so on.
- Big differentiator between processors is the MMU.
- Linaro – an open source community supported by ARM.

- Keil RTX has BSD license now – means this is free.
- CMSIS-Core – header files and startup files
- CMSIS-DSP – DSP libraries
- CMSIS-RTOS – for the RTOS market.
- CMSIS-Packs – distribution of processor files and examples.
DesignStart™

- A way to utilize ARM cores at low cost.
- ARM 926-EJ, Cortex-M0.
- www.arm.com & search for DesignStart
- Cores are initially free – good for universities and startups.
- If successful pay a royalty or license regular M0 or M3 part.
- Debug: Can probe registers, printf via UART.
- FIUBA (Argentina), U of Pittsburgh, Loughborough (UK), NC State University, U of Plymouth (UK), U of Waterloo, UC Berkeley, and Rochester Inst. of Technology.
CoreSight™ Debug & Trace

- CoreSight debug technology delivers enhanced debugging modes and features

- Serial Wire Debug (SWD) Mode
  - 2 wire interface instead of 4 or 5. Most have JTAG too.

- Serial Wire Viewer (SWV)
  - Data R/W, Exceptions, PC Samples

- Enhanced Trace Macrocell (ETM)
  - Adds all the program counters.
  - Provides Code Coverage, Timings, Performance Analysis, crash reports.

- Program Trace Macrocell (PTM)
  - Cortex-A9, A15. All program counters.

- Embedded Trace Buffer: small 4 – 8K trace buffer.
Serial Wire Viewer: Cortex-M3
Tools: Keil MDK™ with µVision®:

- For Cortex-M and Cortex-R processors.
- Proprietary IDE µVision
- ARM compiler, assembler and linker. Can use GCC.
- ULINK™2, ULINKpro, CMSIS-DAP + more debug adapters.
- Many board support packages (BSP) and examples.
- MDK Professional: TCP/IP. CAN, USB & Flash middleware.
- Serial Wire Viewer and ETM, MTB & ETB Trace supported.
- Evaluation version is free from www.keil.com/arm.
- Is complete turn-key package: no add-ons needed to buy.
- Valuable technical support and updates are included for one year. Can be easily extended.
- Keil RTX RTOS included free with source code. BSD license.
Tools: ARM DS-5™:

- Linux, Android aware multi-core debugger.
- Focus on Cortex-A: also supports Cortex-M and Cortex-R.
- DSTREAM or CMSIS-DAP debug adapters supported.
- Eclipse IDE, ARM debugger, Streamline performance analyzer.
- ARM and or GCC compiler, assembler and linker.
- Many board support packages (BSP) and examples.
- ETM, ETB and PTM Trace supported.
- ARM RTSM Fast Models included.
- Evaluation version is free 30 days from [www.arm.com/ds5](http://www.arm.com/ds5)
- Is complete turn-key package: no add-ons needed to buy.
- Valuable technical support included for one year. Can be easily extended.
Instruction Trace Support in DS-5™

- Trace capture via DSTREAM™
  - 4GB trace buffer for ETM/PTM
  - Trace streaming for ITM/STM

- Trace enables
  - Non-intrusive analysis of complex timing-related HW and SW bugs
  - Non-intrusive performance analysis of critical bits of code
**ARM Streamline Performance Analyzer**

- Sample-based analysis of Linux kernel and applications
  - Profiling, call chain, call graph, and timeline views
  - Support for software events and performance counters
  - Requires only a network (Eth, USB) connection to the target or a dump to a file system

- ARM Streamline provides:
  - Code Coverage
  - Performance Analysis
Energy Meter and Powerprofiling

- Software developers aware of energy
- How code is designed affects power.
- Energy readable by applications
- Measurement on multiple power rails
- API that support on-chip Energy info

![Prototype Image]

---

<table>
<thead>
<tr>
<th>Cycles</th>
<th>Power</th>
<th>Instructions</th>
<th>Cache</th>
<th>CPU Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>6 M</td>
<td>4 M</td>
<td>180 K</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.987,924</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.796</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,151,289</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>24,648</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10,202</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.258</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimize Idle → Sleep

Peripherals off

Not optimized Linux Kernel

Optimized Linux Kernel
Challenges and Opportunities:

- Android is has leaked ing out of the cellphone market.
- Multi-core – is coming now.... Cortex M4+M0, Dual Cortex-A9
- Cross-triggering of core is important.
- Challenge is getting all the cores working together efficiently: this is a software issue.
- GCC is really good now. ARM actively supports GCC.
- ARM Compiler 6 builds on LLVM technology.
- ARM DS-5, Keil MDK, many other toolchains…
END