ARM Microcontroller Updates - Markets, Technologies and Trends

Joseph Yiu
Embedded Technology Specialist, ARM
From 2004 to 2013

- Cortex®-M3 was announced in 2004
- First product released in 2006
- In 2013
  
  - 160+ licensees of ARM Cortex-M processors
  - Over 2000 Cortex-M processor-based MCU types
  - Tens of thousands Cortex-M based applications
Cortex-M Microcontroller Processors

Cortex-M0
- “8/16-bit” applications
- Lowest cost

Cortex-M0+
- “8/16-bit” applications
- Lowest power
- Outstanding Energy efficiency

Cortex-M3
- “16/32-bit” applications
- Performance efficiency
- Feature rich connectivity

Cortex-M4
- “32-bit/DSC” applications
- MCU plus DSP
- Accelerated SIMD, FP & SP

Energy Efficient
Configurable
Deterministic
Low Silicon Area

Coherent architecture across all applications

WIDELY ADOPTED
MARKET PROVEN
HIGH VOLUME

The Architecture for the Digital World®
Clear Market Trend

- Expected volume in 2013 – 4.5 billion units (23% of 19.1 billion units)

<table>
<thead>
<tr>
<th>Units (Bn)</th>
<th>% share</th>
<th>Growth%</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/8-bit</td>
<td>6.7</td>
<td>35</td>
</tr>
<tr>
<td>16-bit</td>
<td>7.9</td>
<td>41</td>
</tr>
<tr>
<td>32-bit</td>
<td>4.5</td>
<td>23</td>
</tr>
</tbody>
</table>

- Estimated ARM MCU market share
  - 2011 – 15%
  - 2012 – 18%

Report from IC InSights (www.icinsights.com)

Data from ARM Q1 result presentations 2011,12
MCU Market Moving to 32-bit

- Shipment data shows a growing share of 32-bit MCU since late 90’s

- Growth in 32-bit overtake 8/16-bits

- Traditional 8/16-bit applications migrating to 32-bit

Source: WSTS (Worldwide Semiconductor Trade Statistics) 
Courtesy of Freescale
ARM Cortex-M Reaching Beyond MCU

- Total Cortex-M volume
  - 2010 = 376M
  - 2012 = 2.2Bn (~6x growth)
- Cumulatively >3.5B Cortex-M processors shipped
- >160 Licensees
- Wireless, SoCs, Automotive
- Emerging markets
  - Sensors
  - MEMS
  - Power Management IC (PMIC)
  - Touch controllers
  - Connectivity - Internet of Things (IoT)
Reasons for 32-bit Momentum

- Technology
  - Performance & capability
  - Energy Efficiency & low power
  - Easy to use – save development time !!!
  - Portable & Scalability – Safe guard your software investment

<table>
<thead>
<tr>
<th>Processor</th>
<th>CoreMark/MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-M4</td>
<td>3.40</td>
</tr>
<tr>
<td>Cortex-M3</td>
<td>3.32</td>
</tr>
<tr>
<td>Cortex-M0+</td>
<td>2.15</td>
</tr>
<tr>
<td>Cortex-M0</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Active current (µA/MHz)

- 200
- 150
- 130
- 110
- 90

Power/MHz
- CPU
- Flash
- SRAM
- Peripherals
Price of ARM Microcontrollers

- Comparable price level to 8/16-bit products

- Save cost by using ARM MCUs
  - Competitive microcontroller prices
  - Reducing time to market
  - Better software reusability
  - Better ecosystem
What has changed in the last 12 months?

TECHNOLOGIES
Compilers

- **ARM Compilers and Tools**
  - ARM DS-5™ – Performance improvement in version 5.14
  - Keil™ MDK-ARM™ 5.0 beta – CMSIS-PACK software component package support

- **gcc** ([https://launchpad.net/gcc-arm-embedded](https://launchpad.net/gcc-arm-embedded))
  - Version 4.7 release (end of 2012)
    - Newlib-nano for code size reduction
    - MAC installation tarball
  - Version 4.7 update 1
    - Link Time Optimization (LTO)
    - Basic Cortex-A profile multilib
    - Ubuntu PPA based package

- **IAR Embedded Workbench**
  - Certified for functional safety development (IEC 61508 & ISO 26262)
Cortex Microcontroller Standard (CMSIS)

- Software layers for all Cortex-M processor-based devices
  - CMSIS-CORE : API for Cortex-M processor and core peripherals
  - CMSIS-DSP : DSP Library with 61 functions for Cortex-M
  - CMSIS-SVD : XML system view description for peripherals
  - CMSIS-RTOS : API for RTOS integration
  - CMSIS-DAP : Firmware for Debug Access Port
CMSIS-DAP

- Provides a standardized interface for debuggers
  - Access to all CoreSight™ registers + units (Cortex-A, Cortex-R and Cortex-M profile)
  - Supports multi-core debugging

- Easy to implement on different debug hardware units
  - Debug hardware can be implemented on vendor board using vendor silicon

- Provided for free by ARM
  - Specification and reference implementation available from arm.com
mbed.org – SDK 2.0 released

- Easy to use, low cost
- Web base development
- Open source (from 2.0)
- RTOS support
Cooperation with MathWorks

Matlab/Simulink generates optimized code using CMSIS-DSP Library
- Important for customer with control applications
- Cortex-M4 can replace dedicated DSP in many scenarios

Currently in Review
- Planned availability: H2 2013
What will happen next?

TRENDS
Microcontroller Market Landscape

- Increasing competition on low cost microcontrollers
  - 8-bit and 16-bit no longer necessarily cheaper, especially when factor in:
    - Code density
    - Easy of use / development time
    - Middleware
    - Energy efficiently / Battery life
  - Key differentiations for microcontrollers
    - Software offering, low power, innovations

- Definitions of “Microcontrollers” is changing
  - Cortex-A processors for embedded/industrial applications
  - Mixed signal designs and microcontrollers are converging
Convergence of MCU and Mixed Signal

Microcontrollers are getting more analogue components

Analogue components are getting more intelligence

Sensors / MEMS / Analog
CMSIS Roadmap Continues To Evolve…

- New proposals being discussed with partners
  - CMSIS-PACK: Accelerate software development
    - Software packages in ZIP format
    - XML based description files
    - Easier middleware integration
    - Better software reuse
    - Include toolchain and compatibility information
    - Demonstrated with Keil MDK 5 beta
      (http://www2.keil.com/mdk5/)

- CMSIS-Driver: Unified device driver API
  - Consistent driver collection for middleware
  - Potentially on-top of silicon vendor’s drivers
  - Still allow for differentiation
Pack Description File Example

```xml
<package schemaVersion="1.0" xmlns:xs="..." xs:noNamespaceSchemaLocation="PACK.xsd">
  <vendor>Keil</vendor>
  <name>CMSIS_RTX</name>
  <description>RTX is a CMSIS-RTOS compliant RTOS for Cortex-M based devices</description>
  <license>License.txt</license>
  <url>http://www.keil.com/demo/eval/rtx.htm</url>

  <releases>
    <release version="4.70.0">
      Updates:
      - osTimerCreate can be called prior to osKernelStart (but after osKernelInitialize)
      - initialization of external timer corrected for Cortex-M0/M0+/M1
      - Message/Mail Queue behaviour corrected when timeout expires
    </release>
  </releases>

  <conditions>
    <condition id="CMSIS_Core">
      <description>This component requires the CMSIS CORE component</description>
      <require Cclass="CMSIS" Cgroup="CORE"/>
    </condition>

    <!-- ARMCC -->
    <condition id="CM0_LE_ARMCC">
      <description>Cortex-M0 or Cortex-M0+ or SC000 processor based device in little endian mode for the ARM Compiler</description>
      <accept Dcore="Cortex-M0"/>
      <accept Dcore="Cortex-M0+/"/>
      <accept Dcore="SC000"/>
      <require Dendian="Little-endian"/>
      <require Tcompiler="ARMCC"/>
    </condition>
  </conditions>
</package>
```

Supplier and release information

Dependency on other component

Dependency on core, endianness and toolchain
Pack Description File Example

```
<components>

<component Cclass="CMSIS" Cgroup="RTOS" Csub="Keil RTX" condition="CMSIS_Core">
    <description>RTX is a CMSIS RTOS implementation for Cortex-M, processor based devices.</description>
    <files>
        <!-- CPU and Compiler independent -->
        <file category="doc" name="Doc\index.html"/>
        <file category="header" name="INC\cmsis_os.h"/>
        <file category="source" name="Templates\RTX\Conf_CM.c" copy="true"/>
        <!-- CPU and Compiler dependent -->
        <!-- ARMCC -->
        <file category="library" condition="CM0_LE_ARMCC" name="Lib\ARM\RTX_CM0.lib"/>
        <file category="library" condition="CM0_BE_ARMCC" name="Lib\ARM\RTX_CM0_B.lib"/>
        <file category="library" condition="CM3_LE_ARMCC" name="Lib\ARM\RTX_CM3.lib"/>
        <file category="library" condition="CM3_BE_ARMCC" name="Lib\ARM\RTX_CM3_B.lib"/>
        <file category="library" condition="CM4F_LE_ARMCC" name="Lib\ARM\RTX_CM4.lib"/>
        <file category="library" condition="CM4F_BE_ARMCC" name="Lib\ARM\RTX_CM4_B.lib"/>
        <!-- GCC -->
        <file category="library" condition="CM0_LE_GCC" name="Lib\GCC\libRTX_CM0.a"/>
        <file category="library" condition="CM0_BE_GCC" name="Lib\GCC\libRTX_CM0_B.a"/>
        <!-- IAR -->
        <file category="library" condition="CM4F_LE_IAR" name="Lib\IAR\RTX_CM4.a"/>
        <file category="library" condition="CM4F_BE_IAR" name="Lib\IAR\RTX_CM4_B.a"/>
    </files>
</component>
</components>
```

Common files

Files with processor dependencies for ARM Compiler

Files with processor dependencies for GCC Compiler

Files with processor dependencies for IAR Compiler
Energy Efficiency is The Next King

- Many new ARM based microcontrollers are under 100uA/MHz
- New benchmarks being developed:
  - EEMBC and VOLKSWAGEN to Develop Benchmarking Standards to Quantify Microcontroller Energy Efficiency (May 2013)
    - Performance (system level - including peripherals)
    - Power / energy efficiency
    - Latency
  - EEMBC aims to create ultra low power microcontroller benchmark (Jan 13)
    - Processing efficiency
- Low power on-chip memories is the next frontier
  - e.g. Flash sampling technique in ATMEF SAM4L (www.atmel.com/images/Redefining_the_Power_Benchmark.pdf)
  - MRAM / FRAM technologies entering MCU designs
Internet of Things (IoT)

- Many pieces of the puzzle are ready / almost ready …
  - **Ultra low power microcontrollers** - for always on systems
  - **Weightless** SIG – v1.0 of the specification has arrived (April 2013)
    - Low power wireless communication standard targets at M2M
    - Low cost
    - >10Km range
  - Software ecosystem building up…. e.g.
    - **Oracle Java ME 8** – expect Q1 2014
      - Smaller memory footprint
      - Peripheral APIs
  - Increasing demand for middleware
    - Protocol stack, GUI library, file system, crypto, multimedia…
  - Still lots of issues to solve
    - E.g. Application level M2M protocols

http://www.weightless.org
Where are the Opportunities?

- **Sensors**
  - Energy management, security, monitoring
- **Motors**
  - Preventative maintenance, performance monitoring
- **Medical monitoring**
  - Chronic disease management, on-body wellness tracking
- **Asset/Goods Tracking**
  - Hospital equipment, RFID “from a distance”
- **The Previously Unconnected**
  - Streetlights, traffic signs/lights, parking meters, vending machines
- **Middleware**
  - Communication protocols, crypto, GUI, multimedia processing
- **IoT Platforms**
THANK YOU!