Designing ARM Cortex-M0 Processor into a Mixed Signal Application

Shyam Sadasivan
Analog Mixed Signal Requirement

Analog

MCU

Digital

Analog
Broader Market Challenges

- **8/16-bit running out of performance headroom**
  - As complexity rises so does frequency and memory requirement

- **Need for greater energy efficiency**
  - Wireless sensors, motor control, metering

- **Increased features at lower cost**
  - Increasing connectivity (e.g. USB, Ethernet, 802.15, NFC)
  - Drive for increased code reuse and more richer tools environment
  - Analog devices with increased processing and communication needs
ARM Cortex-M0 Processor

- Wake up Interrupt Controller
- Nested Vectored Interrupt Controller
- Efficient 3-stage Processor Pipeline
- Optional CoreSight-compliant Debug
- Thumb-2 code density technology
- Single AHB-Lite Master Interface
- CoreSight
- AMBA AHB-lite Interface
- Bus Matrix
- Debug interface
- NVIC
- ARM Core
- WIC
- Debug
## Cortex-M0 Processor Benefits

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy efficiency</td>
<td><strong>Energy efficiency</strong>&lt;br&gt;Lower energy costs&lt;br&gt;Low power implementation&lt;br&gt;Sleep mode support&lt;br&gt;Wake-up Interrupt Controller&lt;br&gt;Increased intelligence at node</td>
</tr>
<tr>
<td>Ease of use</td>
<td><strong>Ease of use</strong>&lt;br&gt;Lower software costs&lt;br&gt;Broad tools and OS support&lt;br&gt;Binary compatible roadmap&lt;br&gt;Pure C target</td>
</tr>
<tr>
<td>High performance</td>
<td><strong>High performance</strong>&lt;br&gt;Competitive products&lt;br&gt;32-bit RISC architecture&lt;br&gt;High efficiency processor cores&lt;br&gt;Integrated Interrupt Controller (NVIC)</td>
</tr>
<tr>
<td>Reduced system cost</td>
<td><strong>Reduced system cost</strong>&lt;br&gt;Lower silicon costs&lt;br&gt;Thumb®-2 code density&lt;br&gt;Area optimised designs&lt;br&gt;CoreSight support</td>
</tr>
</tbody>
</table>
Instruction Set Architecture

- **Thumb**
  - 32-bit operations in 16-bit instructions
  - Introduced in the ARM7TDMI® processor (‘T’ stands for Thumb)
  - Subsequently supported in every ARM processor developed since

- **Thumb-2**
  - Enables a performance optimised blend of 16/32-bit instructions
  - All processor operations can all be handled in ‘Thumb’ state
  - Supported across the Cortex-M processor range
## Cortex-M0 Instructions

- **Simple 16-bit Thumb ISA from ARM7TDMI**
  - Just 56 instructions, all with guaranteed execution time
  - 8, 16 or 32-bit data transfers possible in one instruction

<table>
<thead>
<tr>
<th>Thumb</th>
<th>Thumb-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>User assembly code, compiler generated</td>
<td>System, OS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADC</th>
<th>ADD</th>
<th>ADR</th>
<th>AND</th>
<th>ASR</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIC</td>
<td>BL</td>
<td>BX</td>
<td>CMN</td>
<td>CMP</td>
<td></td>
</tr>
<tr>
<td>EOR</td>
<td>LDM</td>
<td>LDR</td>
<td>LDRB</td>
<td>LDRH</td>
<td>LDRSB</td>
</tr>
<tr>
<td>LDRSH</td>
<td>LSL</td>
<td>LSR</td>
<td>MOV</td>
<td>MUL</td>
<td>MVN</td>
</tr>
<tr>
<td>ORR</td>
<td>POP</td>
<td>PUSH</td>
<td>ROR</td>
<td>RSB</td>
<td>SBC</td>
</tr>
<tr>
<td>STM</td>
<td>STR</td>
<td>STRB</td>
<td>STRH</td>
<td>SUB</td>
<td>SVC</td>
</tr>
<tr>
<td>TST</td>
<td>BKPT</td>
<td>BLX</td>
<td>CPS</td>
<td>REV</td>
<td>REV16</td>
</tr>
<tr>
<td>REVSH</td>
<td>SXTB</td>
<td>SXTH</td>
<td>UXTB</td>
<td>UXTH</td>
<td></td>
</tr>
</tbody>
</table>

- Thumb instructions present in ARM7TDMI
Program Registers

- All registers are 32-bit wide
  - The same as ARM7TDMI and other ARM processors
  - Instructions exist to efficiently support packed 8/16/32-bit data in memory

- 13 general purpose registers
  - Registers r0 – r7 (Low registers)
  - Registers r8 – r12 (High registers)

- Only 3 registers with special meaning/usage
  - Stack Pointer (SP) – r13
  - Link Register (LR) – r14
  - Program Counter (PC) – r15
Programmer’s Model – Memory Map

- 4GB linear memory space
- Standard across all Cortex-M implementations for consistency and code portability
- Divided in to regions for better portability
- No need for complex software paging schemes
Nested Vectored Interrupt Controller

- Very fast interrupt response
  - With less software effort

- ISR written directly in C
  - Interrupt table is simply a set of pointers to C routines
  - ISRs are standard C functions
  - Focus is on ease of programming

- Integrated NVIC handles:
  - Saving corruptible registers
  - Exception prioritization
  - Exception nesting

---

<table>
<thead>
<tr>
<th>8 bit example</th>
<th>Cortex-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SJMP/L JMP from vector table to handler</td>
<td></td>
</tr>
<tr>
<td>2. PUSH PSW</td>
<td></td>
</tr>
<tr>
<td>3. ORL PSW, #00001000b (to switch register bank)</td>
<td></td>
</tr>
<tr>
<td>4. Starting real handler code</td>
<td></td>
</tr>
</tbody>
</table>

1. Starting real handler code

Tail-chaining

The Architecture for the Digital World®
Interrupt Handling with NVIC

Traditional approach:

- Exception table
  - Fetch instruction to branch
- Top-level handler
  - Routine handles re-entrancy

```
IRQVECTOR
  LDR   PC, IRQHandler
```

```
IRQHandler PROC
  STMFD sp!,{r0-r4,r12,lr}
  MOV r4,#0x80000000
  LDR r0,[r4,#0]
  SUB sp,sp,#4
  CMP r0,#1
  BREQ C_int_handler
  MOV r0,#0
  STR r0,[r4,#4]
  ADD sp,sp,#4
  LDMFD sp!,{r0-r4,r12,lr}
  SUBS pc,lr,#4
ENDP
```

ARM Cortex-M family:

- NVIC automatically handles
  - Saving corruptible registers
  - Exception prioritization
  - Exception nesting
- ISR can be written directly in C
  - Pointer to C routine at vector
  - ISR is a C function
- Faster interrupt response
  - With less software effort
- WFI and Sleep On Exit provide further flexibility
Debug Extensions

- Optional CoreSight™-technology compliant debug

- CoreSight is a flexible standard debug system developed by ARM for SoC and microcontrollers

- CoreSight supports both invasive and non-invasive debug:
  - Single stepping support
  - Debugger has background access to all memory and registers
  - Profiling support to help to optimize code

- Breakpoints for instruction comparison
  - Maximum of 4 instruction comparators are supported

- Data Watchpoints for data comparison
  - Up to 2 hardware comparators are supported
Optimised Debug with 2-pin SWD

- The recommended 2 pin debug solution:
  - Optimised to access memory mapped debug devices
  - Tested and supported with ARM deliverables
  - Silicon proven since 2004
  - Widely supported by a large debug tool ecosystem
An Example AMBA AHB-Lite System

- AHB-lite
  - Single Master
  - Simple Slaves
  - No retry or split responses
- Standard AHB modules can be used
- Allows easier module design/debug

- Single clock edge operation
- Uni-directional busses
  - No tri-state signals
  ⇒ Good for synthesis
- Allows burst transfers
- Pipelined operation

Very low gate count for simple systems
Example system provided as part of the processor package
Integration and implementation guide and C source files for integration test
Includes example AMBA® AHB-lite single master, single layer interconnect
Includes example GPIO, zero wait-state SRAM/ROM ctrl, PMU components
**ARM Cortex-M0 State Machine**

- **Example software state machine implementation**
  - Events communicated via IRQ lines
  - Interrupt context saving behaviour bypassed to reduce latency
  - Core placed into low power Wait For Event state (WFE)

```assembly
WFE
LDR r0,[intpend]
TST r0,#1
BEQ eventC
.eventB
.eventC
```

- Event B within 3 cycles
- Event B within 7 cycles
- Event C within 9 cycles
Hardware State Machine to Software

- Cortex-M0 has just 56 instructions, and is C-friendly
  - Broad ecosystem of development tools, RTOS and middleware

- Several routes from hardware to software state machine
Keil Development Tools for Cortex-M0

Microcontroller Development Kit
Complete software development environment for Cortex-M0 devices.
Easy to learn and use, yet powerful enough for the most demanding embedded ARM application.

RTX and Real-Time Library
Fully featured real-time kernel
Library of middleware components to speed up software development and solve real-time and communication challenges

ULINK2 USB Adapter
On-the-fly debugging and Flash programming via JTAG or serial interface

Microcontroller Development Kit
- RealView C/C++ Compiler
- RTX RTOS Kernel Library
- μVision Device Database & IDE
- μVision Debugger & Analysis Tools
- Complete Device Simulation

Examples and Templates

Real-Time Library
- RTX RTOS Source Code
- TCPnet Networking Suite
- Flash File System
- USB Device Interface
- CAN Interface

Examples and Templates
Cortex-M0 Processor Support

- Cortex-M0 support in Keil MDK v4.11
  - Compiler, simulator and debugger
  - New µVision4 IDE
- MicroLib library
  - Reduced code size for embedded applications
  - ~50% typical code size saving
- CMSIS compliant
  - Common interface standard for Cortex-M devices
  - Applications, peripherals, RTOS, and middleware
- Cortex-M0 device support
  - Includes start-up code and Flash programming algorithms
  - Complete device simulation possible using an open API
  - Examples, templates, and Board Support Packages (BSPs)
Cortex-M0 RTOS and Library Support

- Running an RTOS on a mixed signal device may seem overkill but …
  - The footprint may only be 1 to 2K bytes
  - A simple schedule could make firmware development easier

- Many optimized solutions available:
  - RTOS with scheduling and task management facilities
  - Library support for USB, TCP/IP, File System & CAN
  - Focus on application development
  - RTX & RL-ARM offer reliable scheduling low-level operations
  - Co-developed with Cortex-M0
  - Integrated support in MDK-ARM
Cortex-M0 Starter Kits

MDK evaluation CD and ULINK-ME included to offer user fully functional development environment.

RealView MDK (Eval)

Example Portfolio (optional)

Development Board

ULINK-ME
Cortex-M0 mbed Evaluation System

Very low cost
USB / evaluation board

No Installation!

Plug it in...

Appears as USB Disk
linking to website

Save to the board and
program runs!

“Hello World!” in 5 minutes

Compile a program online

http://mbed.org/
ARM Cortex-M0 Configuration

- Cortex-M0 RTL is highly configurable
  - Tune for your application

- Consistent programmer’s model
  - Software compatibility
  - All tools remain compatible

- Interrupt options
  - 1, 2, 4, 8, 16, 24 or 32 interrupts

- Multiplier options
  - Fast or small (1 or 32 cycle)

- Optional timer
  - SysTick

- Removable hardware debug
  - 4-2 breakpoints, 2-1 watchpoints
  - JTAG or SWD interface

---

<table>
<thead>
<tr>
<th>ARM Cortex-M0 r0p0 Base</th>
<th>Area (gates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total smallest usable configuration</td>
<td>12.4k</td>
</tr>
<tr>
<td>NVIC, 1 interrupt, small multiply</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ARM Cortex-M0 r0p0 Full</th>
<th>Area* (gates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug (SWD, 1 watchpt, 2 breakpt)</td>
<td>+ 4.8k</td>
</tr>
<tr>
<td>- to 2 watchpt, 4 breakpt</td>
<td>+ 1.4k</td>
</tr>
<tr>
<td>NVIC 16 Interrupts configuration</td>
<td>+ 1.5k</td>
</tr>
<tr>
<td>- to 32 Interrupts configuration</td>
<td>+ 1.5k</td>
</tr>
<tr>
<td>Fast multiplier (1 cycle)</td>
<td>+ 2.8k</td>
</tr>
<tr>
<td>System timer</td>
<td>+ 0.9k</td>
</tr>
<tr>
<td>WIC (max. configuration of 34 lines)</td>
<td>+ 0.2k</td>
</tr>
<tr>
<td>Total</td>
<td>25.5k</td>
</tr>
</tbody>
</table>

Gate count numbers at 50MHz targeting 180ULL Metro
10% lower gate count possible on 90G Advantage
Physical IP Support for Mixed Signal

- **Attributes**
  - Cost sensitive market (Area + time to market + low mask cost)
  - Long battery life (Power: Leakage & Dynamic)
  - Medium Performance on mature processes (e.g. 180nm)

**Power Management Kit**
- Support for retention (low leakage) mode
- Power gating support

**Optimized Cell Library**
- Optimized for Density / Power
- Includes clock gating cell

**32 Bit Wide Memory**
- Byte write capability
- Optimized for Density / Power

**Via Programmable ROM**
- Reduced re-program costs
T-180uLL Logic Library Overview

- ARM provides multiple standard cell libraries for the 180uLL process
  - SC9 High Density
    - 9-track library for mainstream applications requiring a balance between speed, area and power
  - SC7 Ultra High Density
    - 7-track library for low power, high density (low cost) applications
      - Library architecture optimized for low power applications
      - Large number of drive strengths with very fine granularity
      - Tap-less cell design enables advanced power management features
  - SC7 Ultra High Density Power-Management Kit
    - Provides functionality to actively manage dynamic and leakage power like power gates and retention flip-flops
Cortex-M Low Power Technologies

- All Cortex-M processors are specifically designed for low power, with a range of complementary technologies including:
  - Integrated architectural clock gating
  - Sleep and deep sleep modes:
    - Puts the processor into a low-power state with flexible software control
  - “Sleep-on-exit” interrupt handling:
    - Enables the processor to sleep whenever all outstanding Interrupts are complete
  - Wakeup Interrupt Controller (WIC)
    - Enables advanced interrupt-controlled processing
    - Enables nW power consumption in deep sleep mode with instant wakeup
SC7 PMK ULP Sleep Modes in Cortex-M

- ARM Cortex-M family has architected support for sleep states
  - Enables ultra low-power standby operation
  - Critical for extended life battery based applications
  - Includes very low gate count Wake-Up Interrupt Controller (WIC)

- **Sleep**
  - CPU can be clock gated
  - NVIC remains sensitive to interrupts

- **Deep sleep**
  - WIC remains sensitive to selected interrupts
  - Cortex-M0 can be put into state retention

- WIC signals wake-up to PMU
  - Core can be woken almost instantaneously
  - React to critical external events

Use the SC7 PMK to implement ultra-low Power Sleep modes & retention in Cortex-M processors
180nm Cortex-M0 Processor using SC7 UHD

Ability to obtain best leakage, dynamic power and area trade-off depending on features desired!
## Advanced Design Flow Support

### EDA Specification/Package 6.2 - 130

<table>
<thead>
<tr>
<th>Deliverables</th>
<th>Standard Validation</th>
<th>ARM Products</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EDA Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>View (title)</td>
<td>Name</td>
<td>Version</td>
<td>Vendor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
<tr>
<td></td>
<td>RTL Compiler</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Liberty/NUM (v1.1b)</td>
<td>8.1</td>
<td>Cadence</td>
</tr>
<tr>
<td></td>
<td>Synopsys</td>
<td></td>
<td>Mentor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VCS</td>
</tr>
</tbody>
</table>

ARM products are validated with tools from leading EDA vendors
Existing Cortex-M3 AMS Devices

- Several analog plus microcontroller devices available
- Two ARM Cortex-M3 processor-based product announcements this year so far:

Actel SmartFusion™

Cypress PSoC® 5
NXP LPC1100 Family

- Key Features:
  - ARM Cortex-M0 processor
  - 50-MHz operation
  - Nested Vectored Interrupt Controller for fast deterministic interrupts
  - Wakeup Interrupt Controller enables automatic wake from any priority interrupt
  - Three reduced power modes: sleep, deep-sleep, and deep power-down
Mocha - Configurable Mixed Signal SoC

Another example of Cortex-M0 and analog

Triad Semiconductor - Mocha at a Glance
• Low power, high performance Cortex-M0
• Completely configurable mixed signal resources
• 4-week fabrication, Low NRE, Low risk

Configurable Digital
• IIC, SPI, UART
• USB, GPIO, PWM
• FPGA Replacement
• Custom Digital,…

Configurable Analog
• Op-Amps, Resistors
• Capacitors, Switches,
• Transistors
• Filters, ADCs, DACs
• Temperature Sensors
• Custom Analog,…

Applications
• Solar Power Micro-Inverter
• Automatic Meter Reading
• Implantable Medical
• Neuro-Stimulation
• Human Interface Control
• Low Power, Portable Devices

Another example of Cortex-M0 and analog
Full ARM Cortex-M Processor Family

- Forget traditional 8/16/32-bit classifications
  - A compatible architecture spanning the embedded application range

**ARM Cortex-M4**
- “32-bit/DSC” applications
- Efficient digital signal control

**ARM Cortex-M3**
- “16/32-bit” applications
- Performance efficiency

**ARM Cortex-M0**
- “8/16-bit” applications
- Low-cost & simplicity
Thank You

Please visit www.arm.com for ARM related technical details

For any queries contact < Salesinfo-IN@arm.com >