Introduction to ARM® Cortex™-M3

October 17, 2007

MCD Application Team

CONTENTS

- Introduction to the Cortex-M3
  - Architecture
  - Overview
  - Comparison to ARM7
  - Bit Banding and Unaligned data access
  - Interrupt and real time capabilities

Introduction to the Cortex-M3

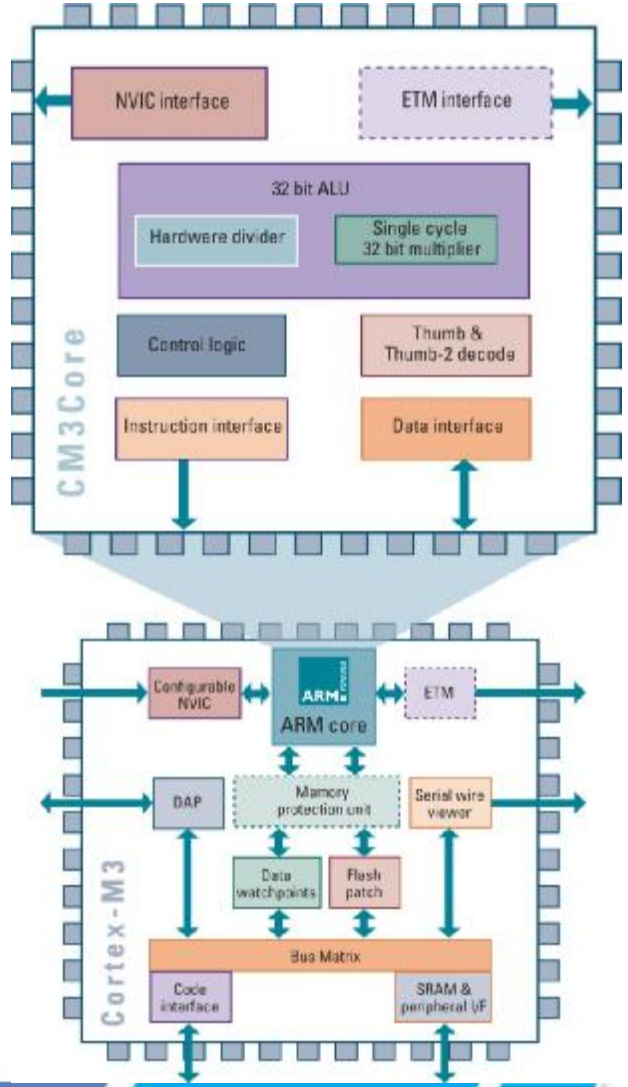
- Three distinct profiles of the ARMv7 architecture
  - A profile for sophisticated, high-end applications
  - R profile for real-time system
  - M profile for cost-sensitive and microcontroller applications.
- Cortex-M3 processor is the first ARM processor based on the ARMv7-M architecture
- Designed to achieve high system performance in power and cost-sensitive embedded application such as, automotive body systems, industrial control systems and wireless networking

CONTENTS

- Introduction to the Cortex-M3
  - Architecture
  - Overview
  - Comparison to ARM7
  - Bit Banding and Unaligned data access
  - Interrupt and real time capabilities

Cortex-M3 Processor Architecture

- Hierarchical processor integrating core and advanced system peripherals
- Cortex-M3 core
  - Harvard architecture
  - 3-stage pipeline w. branch speculation
  - Thumb®-2 and traditional Thumb
  - ALU w. H/W divide and single cycle multiply
- Cortex-M3 *Processor*
  - Cortex-M3 core
  - Configurable interrupt controller
  - Bus matrix
  - Advanced debug components
  - Optional MPU & ETM **(Not available in STM32F10x)**



CONTENTS

- Introduction to the Cortex-M3
  - Architecture
  - Overview
  - Comparison to ARM7
  - Bit Banding and Unaligned data access
  - Interrupt and real time capabilities

Cortex-M3 Processor Overview (1/2)

- ARM v7M Architecture
- Thumb-2 Instruction Set Architecture
  - Mix of 16 and 32 bit instructions for very high code density
- Harvard architecture
  - Separate I & D buses allow parallel instruction fetching & data storage
- Integrated Nested Vectored Interrupt Controller (NVIC) for low latency interrupt processing
- Vector Table is addresses, not instructions

Cortex-M3 Processor Overview (2/2)

- Designed to be fully programmed in C
  - Start Up code
  - Even reset, interrupts and exceptions
- Integrated Bus Matrix
  - Bus Arbiter
  - Bit Banding – Atomic Bit Manipulation
  - Write Buffer
  - Memory Interface (I&D) Plus System Interface & Private Peripheral Bus
- Integrated System Timer (SysTick) for Real Time OS or other scheduled tasks

CONTENTS

- Introduction to the Cortex-M3
  - Architecture
  - Overview
  - **Comparison to ARM7**
  - Bit Banding and Unaligned data access
  - Interrupt and real time capabilities

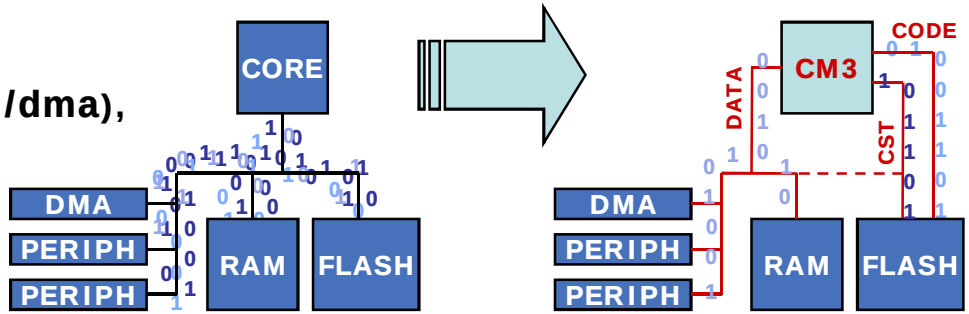
Comparison to ARM7

	ARM7TDMI-S	Cortex-M3
Architecture	v4T	v7M
ISA Support	ARM (32-bit) & Thumb (16-bit)	Thumb-2 (Merged 32/16-bit)
DMIPS/MHz	0.74 Thumb / 0.93 ARM	1.25 Thumb-2
Pipeline	3-Stage	3-Stage + Branch Speculation
Interrupts	FIQ / IRQ	NMI, SysTick and up to 240 interrupts. Integrated NVIC Interrupt Controller up to 1-255 Priorities
Interrupt Latency	24-42 Cycles (Depending on LSM)	12 Cycles (6 when Tail Chaining)
Memory Map	Undefined	Architecture Defined
System Status	PSR. 6 modes. 20 Banked regs	xPSR. 2 modes. Stacked regs (1 bank)
Sleep Modes	No	Three

High Performance CPU and Buses

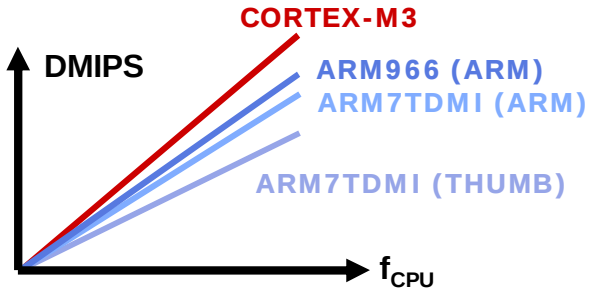
- **ARM v7M Architecture: Harvard benefits with Von Neumann single memory space**

Von Neumann "bottleneck"
Single 32bit bus for:
- code execution,
- data transfer (core/dma),
- peripheral control



Three 32bit buses for a parallel
- code execution,
- data transfer (core/dma),
- peripheral control

- **Outstanding efficiency of 1.2 DMIPS/MHz and 1.2 CPI**



THUMB-2 instruction set

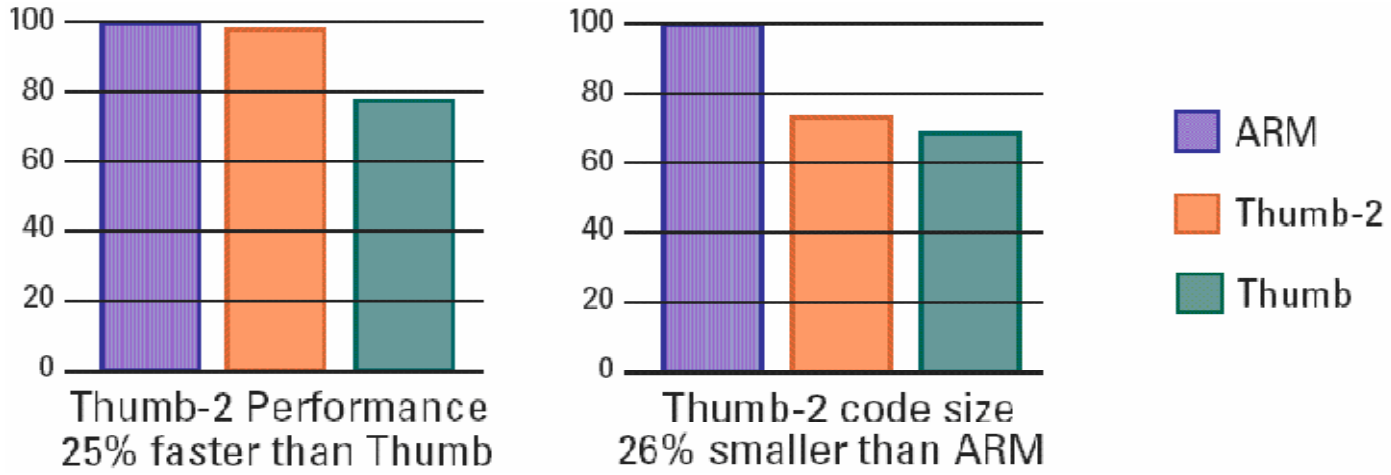
THUMB-2 instruction set provide 32bit performance with 16bit code density

- **THUMB 16bit Instruction Set** → Full THUMB compatibility
- **ARM 32bit Instruction Subset** → Complete ARM instruction set for better performance
- **New 16/32bit Instructions** → 1 cycle MAC and Hardware Divide Bit handling

THUMB-2

- Single POWERFULL instruction set
  - à No more mode switching
- Two 16bit instruction fetch per FLASH access

Reduces Flash Memory Requirement By 25% vs. ARM

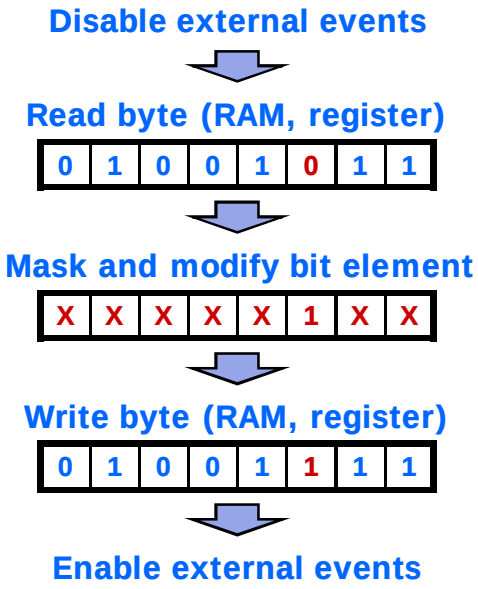


CONTENTS

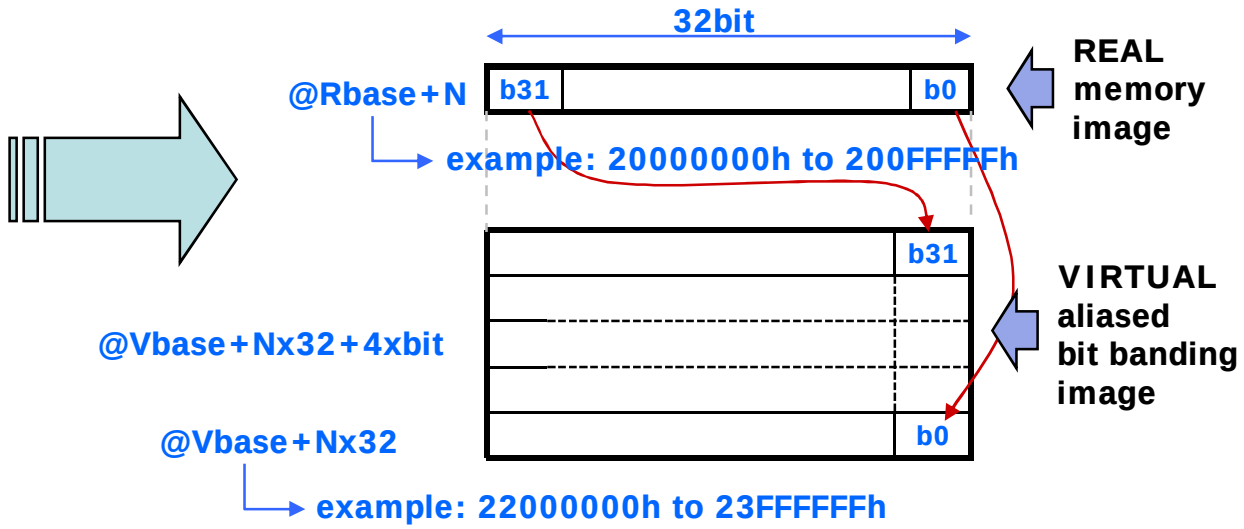
- Introduction to the Cortex-M3
  - Architecture
  - Overview
  - Comparison to ARM7
  - **Bit Banding and Unaligned data access**
  - Interrupt and real time capabilities

Bit Banding

Traditional method



Speed and code size optimized Cortex-M3 implementation



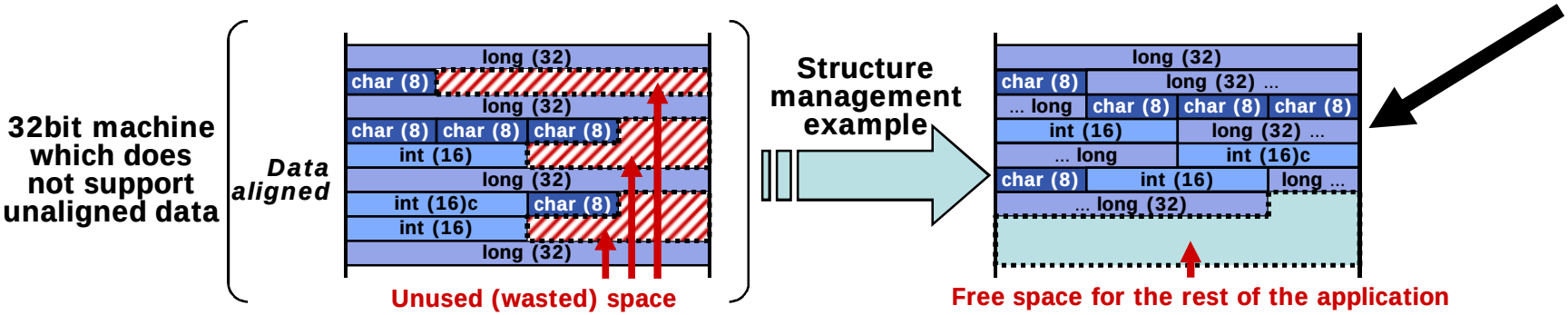
- Single instruction Read/Modify/Write (no more masking)
- No new instruction set à Use standard data one (AND, OR, XOR…)

Optimized RAM, peripherals and IOs registers accesses

Easy multi-task semaphore management

Unaligned data access

- Unaligned data access supported to improve data constant and RAM utilization



Reduces SRAM Memory Requirements By Over 25%

Less Memory - LowER Cost devices

CONTENTS

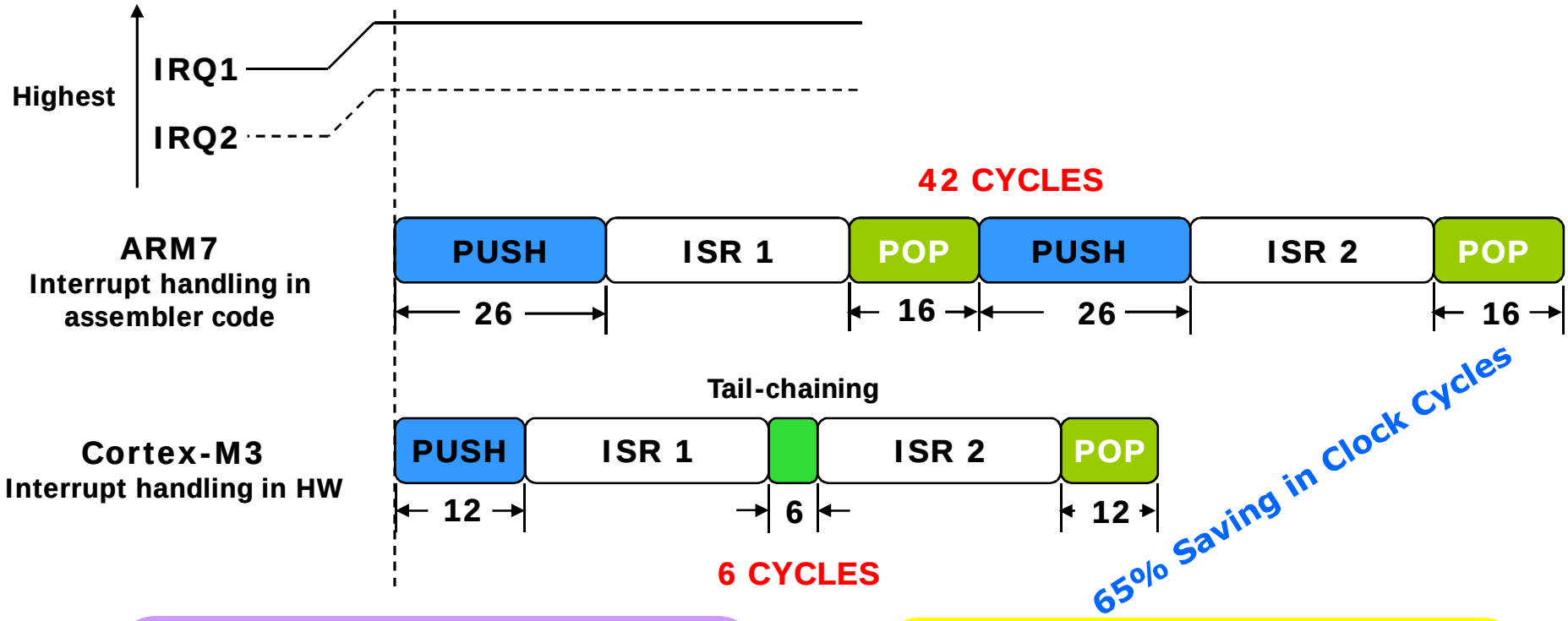
- Introduction to the Cortex-M3
  - Architecture
  - Overview
  - Comparison to ARM7
  - Bit Banding and Unaligned data access
  - **Interrupt and real time capabilities**

Exception / Interrupt Handling

- Very low latency interrupt processing
  - Exceptions processed in Privileged operation
  - Interruptible LDM/STM for low interrupt latency
  - Automatic processor state save and restore
    - Provides low latency ISR entry and exit
    - Allows handler to be written entirely in 'C'
- The Cortex-M3 processor integrates an advanced Nested Vectored Interrupt Controller (NVIC)
  - 43 maskable interrupts channels (not including 16 interrupt lines of Cortex-M3)
  - 16 programmable priority levels
  - Allows early processing of interrupts
  - Supports advanced features for next generation real-time applications
    - Tail-chaining of pending interrupts
    - Late-arrival interrupt handling and priority boosting / inversion

Exceptional Control Capabilities Through Integrated Interrupt Handling

Interrupt Response- Tail Chaining



ARM7

- 26 cycles from IRQ1 to ISR1 entered
  - Up to 42 cycles if LSM
- 42 cycles from ISR1 exit to ISR2 entry
- 16 cycles to return from ISR2

Cortex-M3

- 12 cycles from IRQ1 to ISR1 entered
  - 12 cycles if LSM
- 6 cycles from ISR1 exit to ISR2 entry
- 12 cycles to return from ISR2

Introduction to STM32F10x

October 17, 2007

MCD Application Team

CONTENTS

- STM32F10x Device
  - Block Diagram
  - Memory mapping and boot modes
  - System Architecture
- STM32F10x Peripherals
  - Main features
- STM32F10x USB Developer kit

STM32F10x : 2 first product lines

STM32F103 Performance Line

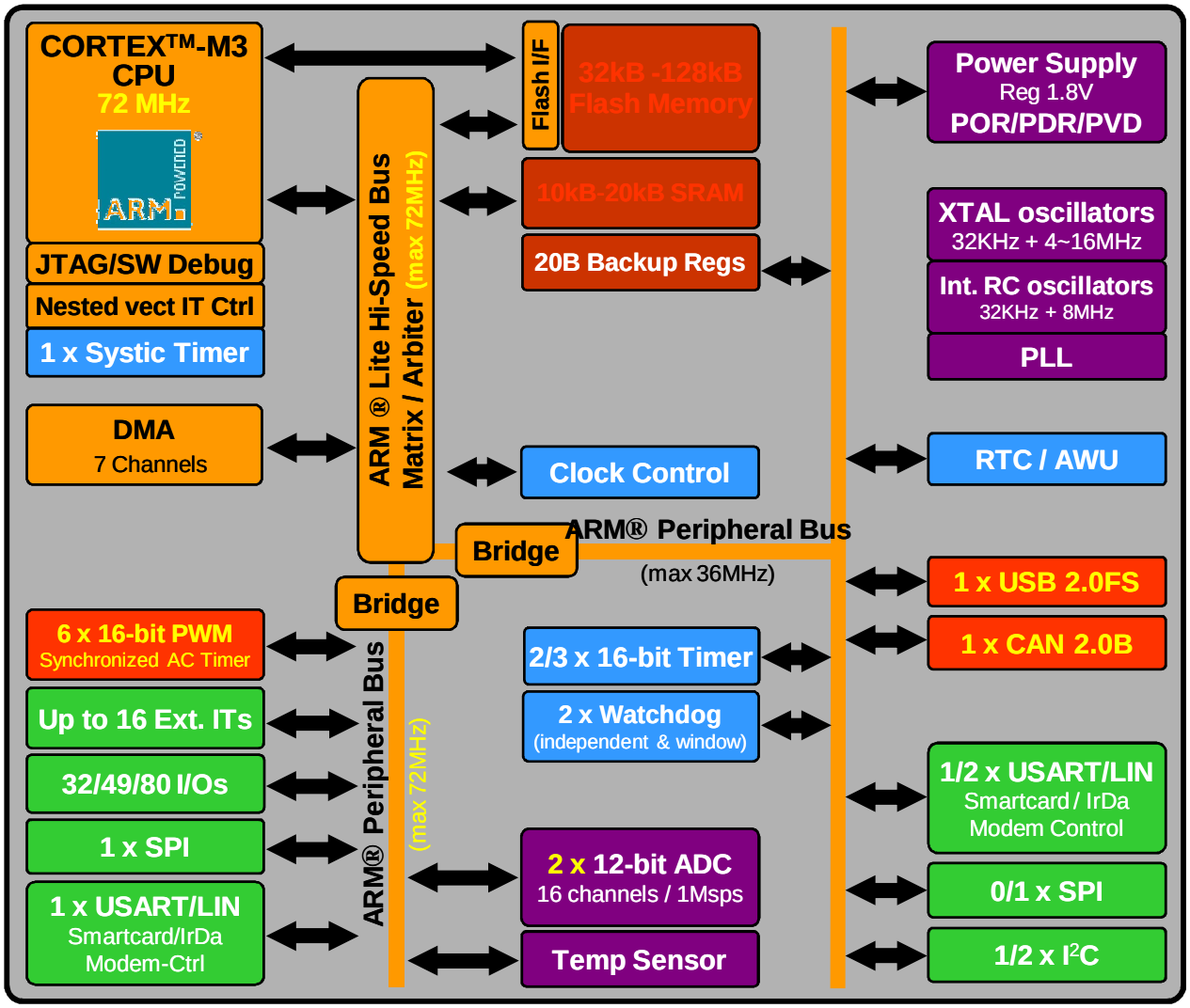
- Best in class 32-bit flash MCU
- Ability to outperform integer DSP solutions
- Superior control & connectivity
- Excellent fit for low voltage / low power applications

STM32F101 Access Line

- 32-bit performance at 16-bit Prices
- Entry point to STM32 world
- Excellent fit for low voltage / low power applications

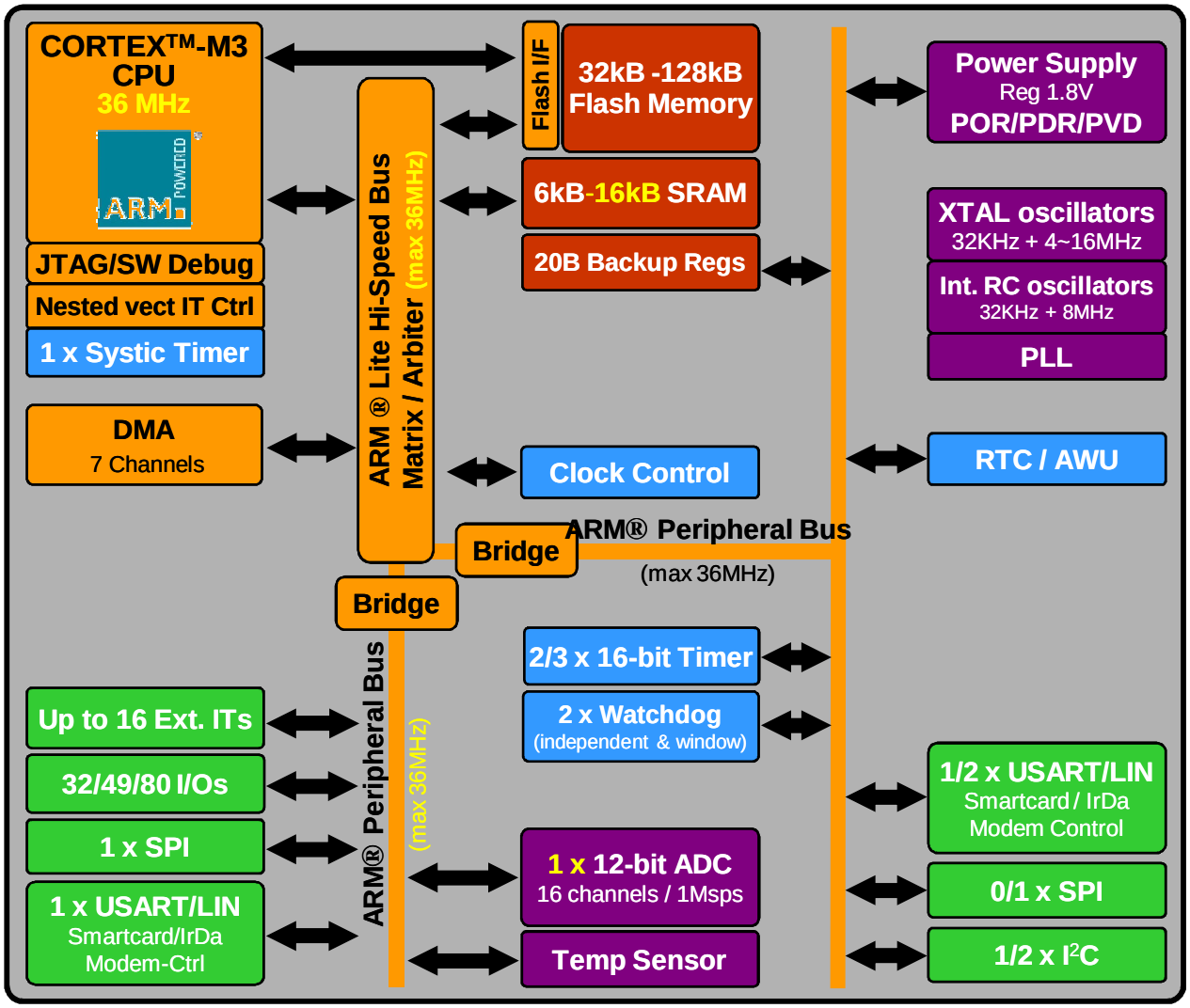
STM32F103 Performance Line

- 2V-3.6V Supply
- 5V tolerant I/Os
- Excellent safe clock modes
- Low-power modes with wake-up
- Internal RC
- Embedded reset
- -40/+105°C



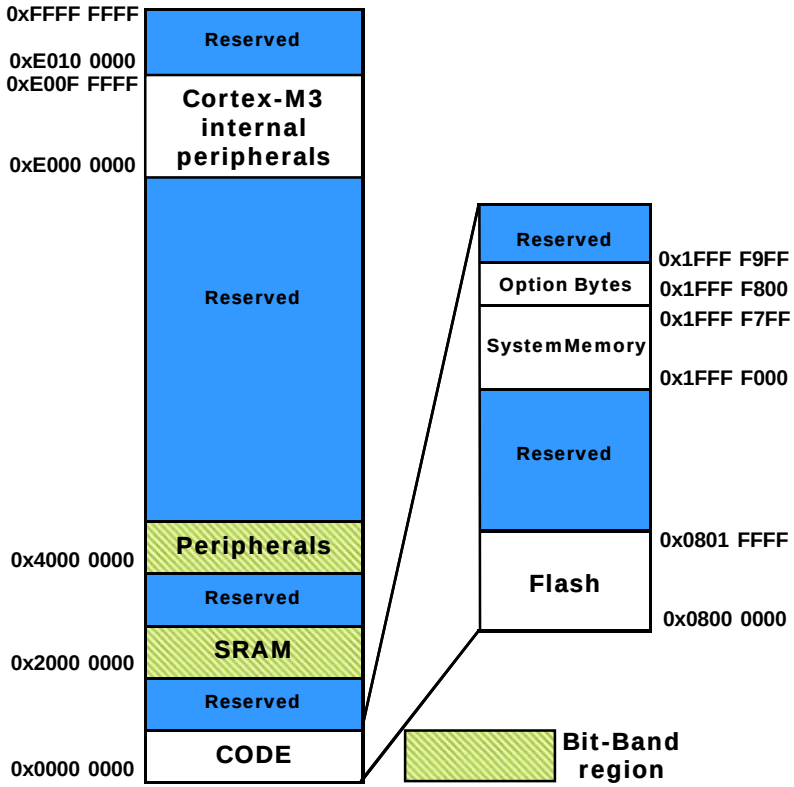
STM32F101 Access Line

- No USB/CAN/PWM TIMER
- 1xADC
- SRAM up to 16K
- -40/+85°C



Memory Mapping and Boot Modes

- Addressable memory space of 4 GBytes
- RAM : up to 20 kBytes
- FLASH : up to 128 kBytes



- **Boot modes**
  Depending on the Boot configuration, Embedded Flash Memory, System Memory or Embedded SRAM Memory is aliased at @0x00

BOOT Mode Selection Pins		Boot Mode	Aliasing
BOOT1	BOOT0		
x	0	User Flash	User Flash is selected as boot space
0	1	SystemMemory	SystemMemory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space

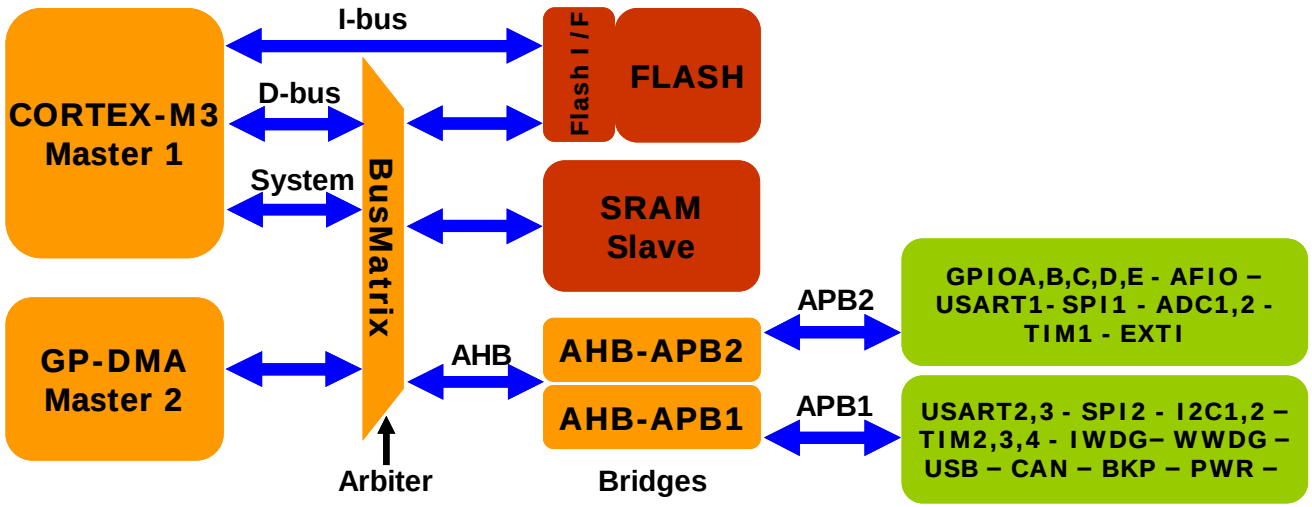
- **SystemMemory**: contains the Bootloader used to re-program the FLASH through USART.

- **Boot from SRAM :**
  In the application initialization code you have to Relocate the Vector Table in SRAM using the NVIC *Exception Table and Offset* register

System Architecture

- Multiply possibilities of bus accesses to SRAM, Flash, Peripherals, DMA
  - BusMatrix added to Harvard architecture allows parallel access
- Efficient DMA and Rapid data flow
  - Direct path to SRAM through arbiter, guarantees alternating access
  - Harvard architecture + BusMatrix allows Flash execution in parallel with DMA transfer
- Increase Peripherals Speed for better performance
  - Dual Advanced Peripheral buses (APB) architecture w/ High Speed APB (APB2) up to 72MHz and Low Speed APB (APB1) up to 36MHz

  è Allows to optimize use of peripherals (18MHz SPI, 4.5Mbps USART, 72MHz PWM Timer, 18MHz toggling I/Os)



Buses are not overloaded with data movement tasks

Embedded FLASH

Flash Features Overview

- **Flash Features:**
  - Up to 128KBytes
  - 1 KByte Page size
  - Endurance: 1000 cycles
  - Memory organization:
    - Main memory block
    - Information block
  - Access time: 35ns
  - Word(32-bit) program time: 20µs
  - Page / Mass Erase Time: 20ms

- **Flash interface (FLITF) Features:**
  - Read Interface with pre-fetch buffer
  - Option Byte loader
  - Flash program/Erase operations
  - Types of Protection:
    - Access Protection
    - Write Protection

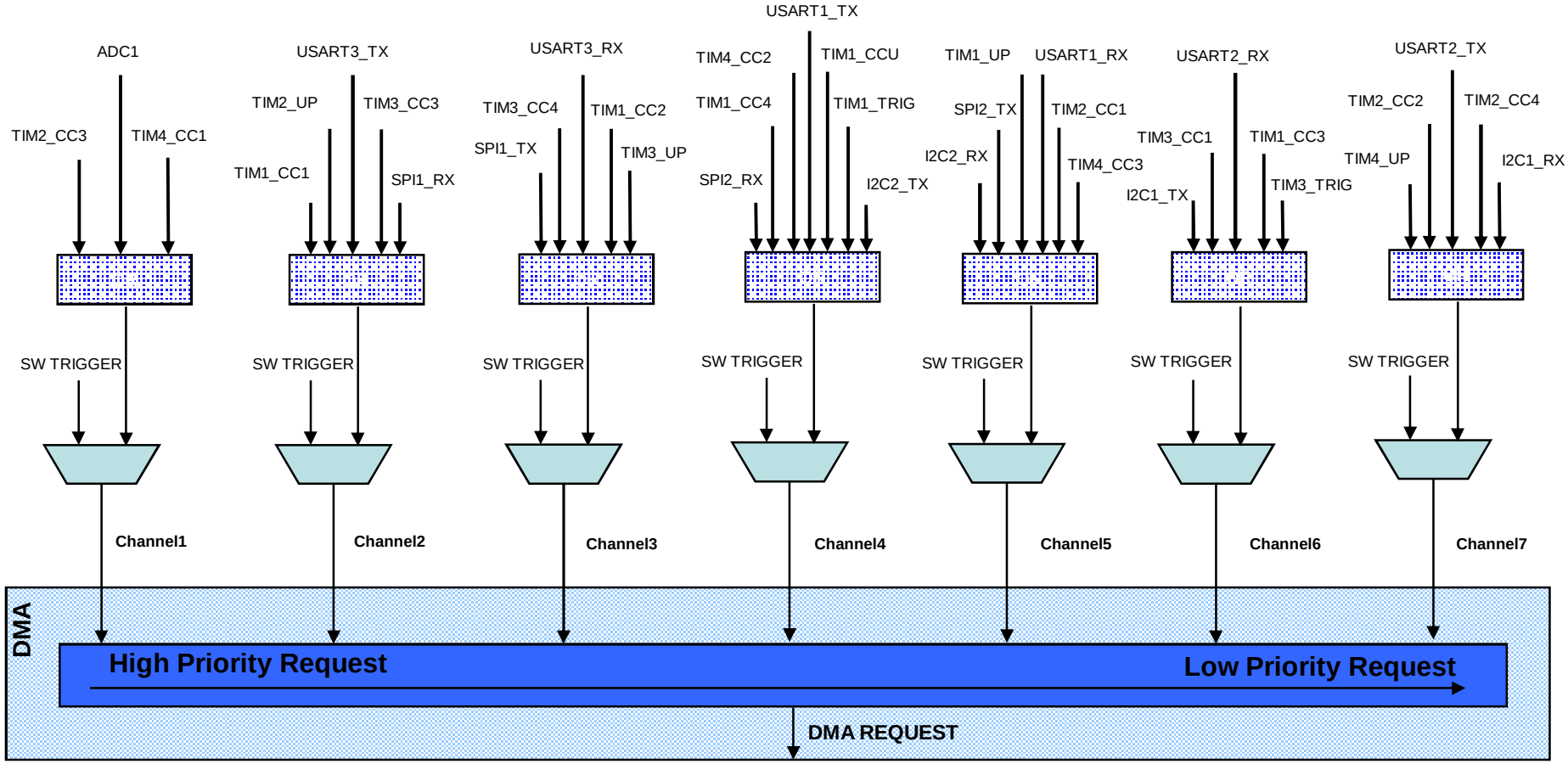
Direct Memory Access (DMA)

DMA Features

- 7 independently configurable channels: hardware requests or software trigger on each channel
- Software programmable priorities: Very high, High, Medium or Low. (Hardware priority in case of equality)
- Programmable and Independent source and destination transfer data size: Byte, Halfword or Word
- 3 event flags for each channel: DMA Half Transfer, DMA Transfer complete and DMA Transfer Error
- Memory-to-memory, peripheral-to-memory and memory-to-peripheral transfers
- Support for circular buffer management

DMA Request Mapping

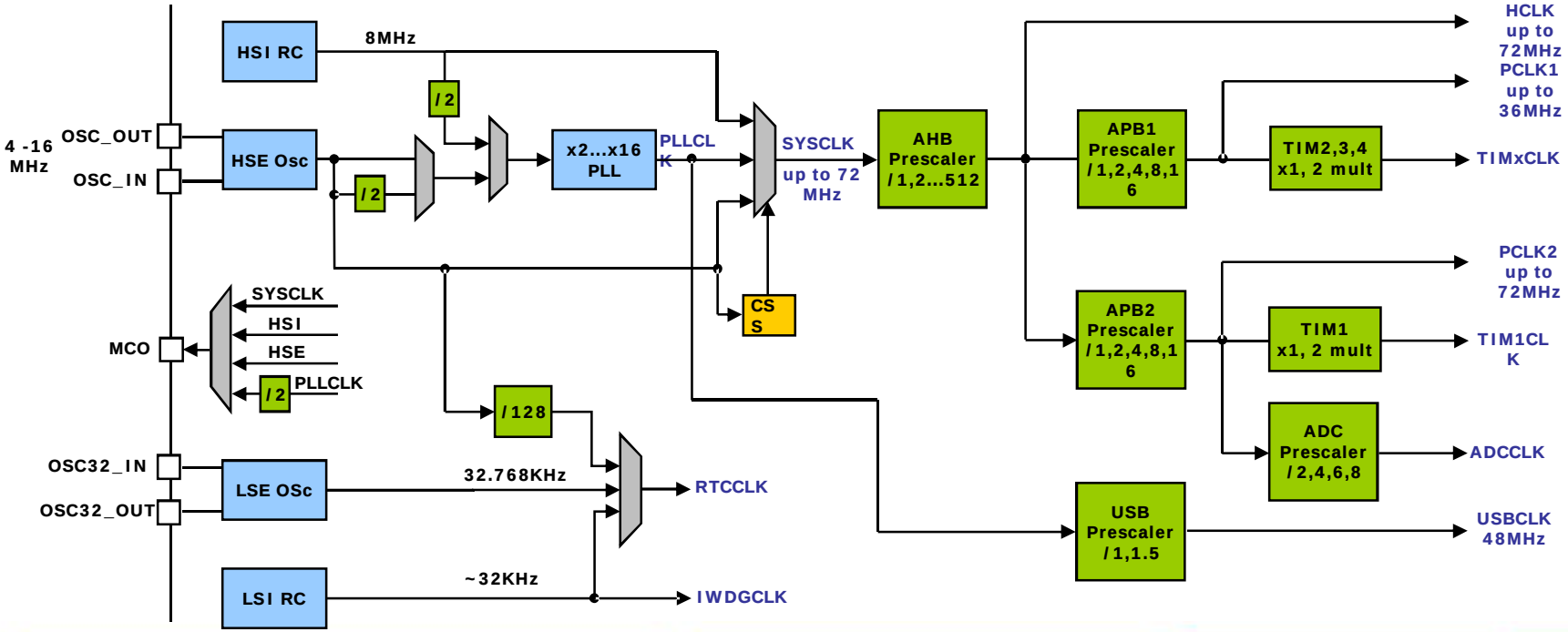
- The DMA controller provides access to 7 channels



Clock Scheme

Clock Scheme

- System Clock (SYSCLK) sources
  - HSI
  - HSE
  - PLL
- RTC Clock (RTCCLK) sources
  - LSE
  - LSI
  - HSE clock divided by 128
- USB Clock (USBCLK) provided from the internal PLL
- Clock-out capability on the MCO pin (PA.08) / max 50MHz
- Configurable dividers provides AHB, APB1/2, ADC and TIM clocks
- Clock Security System (CSS) to backup clock in case of HSE clock failure (HSI feeds the system clock)
  - Enabled by SW w/ interrupt capability linked to Cortex NMI



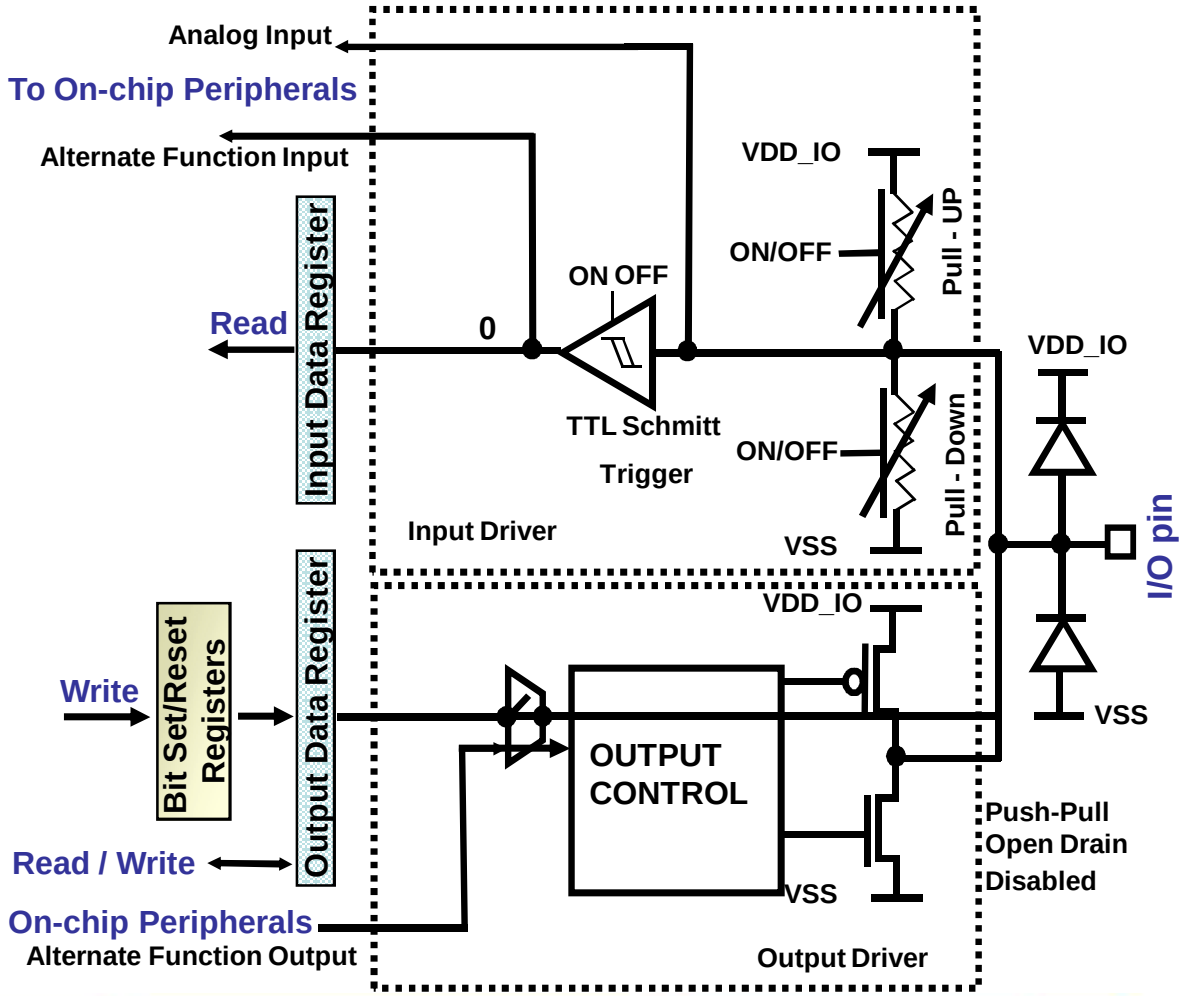
General Purpose and Alternate Function I/O (GPIO and AFIO)

GPIO Features

- 80 multifunction bi-directional I/O ports available: 80% IO ratio
  - 80 Standard I/Os (5V tolerant, 20 mA drive)
  - 18 MHz Toggling
  - Configurable Output Speed up to 50 MHz
  - Up to 16 Analog Inputs
  - Alternate Functions pins (like USARTx, TIMx, I2Cx, SPIx, CAN, USB…)
  - All I/Os can be set-up as external interrupt (up to 16 lines at time)
  - One I/O can be used as Wake-Up from STANDBY (PA.00)
  - One I/O can be set-up as Tamper Pin (PC.13)
  - All Standard I/Os are shared in 5 ports (GPIOA..GPIOE)
  - Atomic Bit Set and Bit Reset using BSRR and BRR registers
  - Locking mechanism to avoid spurious write in the IO registers
    - When the LOCK sequence has been applied on a port bit, it is no longer possible to modify the configuration of the port bit until the next reset (no write access to the CRL and CRH registers corresponding bit).

GPIO Configuration Modes

Configuration Mode	CNF1	CNF0	MOD1	MOD0
Analog Input	0	0	00	
Input Floating (Reset State)	0	1		
Input Pull-Up	1	0		
Input Pull-Down	1	0		
Output Push-Pull	0	0	00: Reserved 01: 10 MHz 10: 2 MHz 11: 50 MHz	
Output Open-Drain	0	1		
AF Push-Pull	1	0		
AF Open-Drain	1	1		



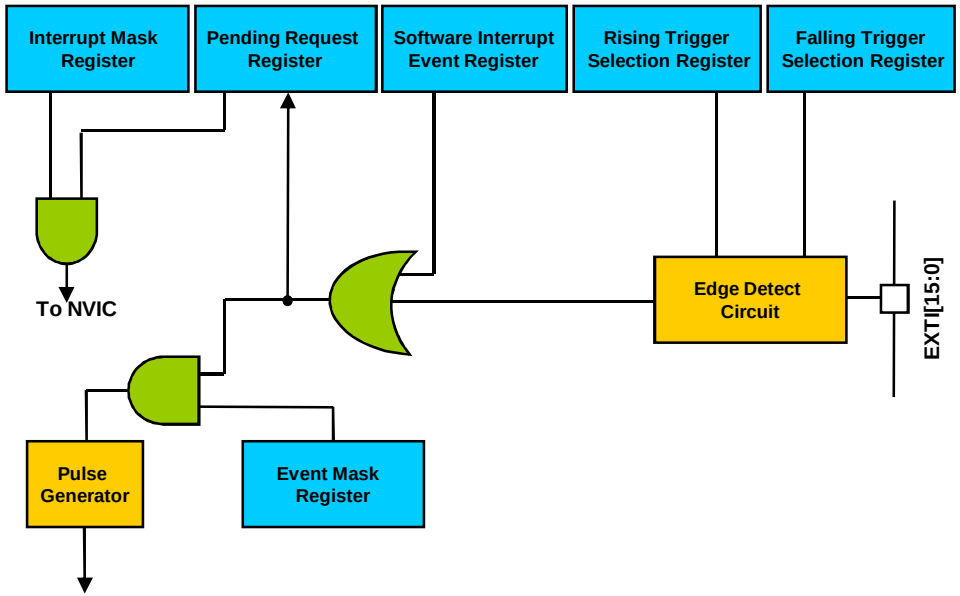
AFIO Features

- **Event Out signal generation**
  - Pulse generation with SEV instruction: to wake-up an other MCU from low power mode through its Event In signal
  - Each IO can be used as Event Out
- **GPIO Software Remapping**
  - Some Alternate function can be remapped in two different pins allowing optimization of the pin out
  - All SWJ-DP I/O pins can be used as GPIO
- **EXTI Lines Configuration**
  - Each EXTI line is shared with all GPIO ports: EXTI Linexx à GPIO[A..E].xx

External Interrupt/Event Controller (EXTI)

EXTI Features

- Up to 19 Interrupt/Events requests
  - All GPIO can be used as EXTI line(0..15)
  - EXTI line 16 connected to PVD output
  - EXTI line 17 connected to RTC Alarm event
  - EXTI line 18 connected to USB Wake-up from suspend event
- Two Configuration mode:
  - Interrupt mode: generate interrupts with external lines edges
  - Event mode: generate pulse to wake-up system from WFI, WFE and STOP modes
- Independent trigger (rising, falling, rising & falling) and mask on each interrupt/event line
- Dedicated status bit for each interrupt line
- Generation of up to 19 software interrupt/event requests



- Minimum Pulse Width : < $1*T_{PCLK2}$ (Fast APB)
- EXTI mapped on High Speed APB (APB2) to save time entering in the External Interrupt routine

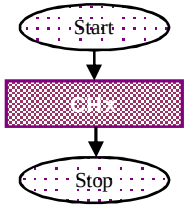
Analog-to-Digital Converter (ADC)

ADC Features

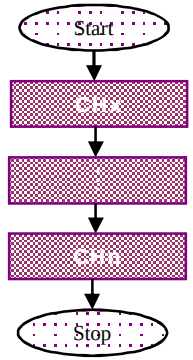
- ADC conversion rate 1 MHz and 12-bit resolution
- ADC supply requirement: 2.4V to 3.6 V
- ADC input range: VREF- ≤ VIN ≤ VREF+ (VREF+ and VREF- available only in LQFP100 package)
- Dual mode (on devices with 2 ADCs): 8 conversion mode
- Up to 18 multiplexed channels:
  - 16 external channels
  - 2 internal channels: connected to Temperature sensor and internal reference voltage (Bandgap voltage)
- Self-calibration
- DMA capability (only on ADC1)

ADC conversion modes

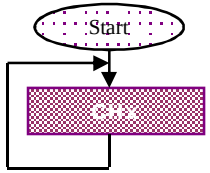
- Single and continuous conversion modes
- Four conversion mode are available:



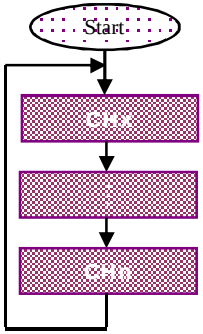
Single channel single conversion mode



Multi-channels (Scan) single conversion mode



Single channel continuous conversion mode



Multi-channels (Scan) continuous conversion mode

Serial Peripheral Interface (SPI)

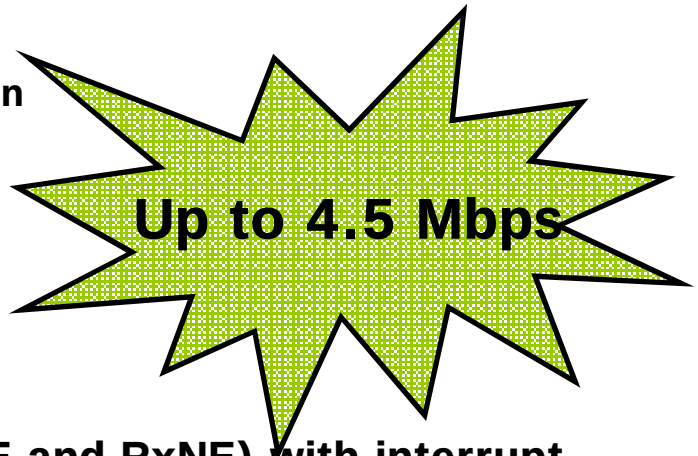
SPI Features

- Two SPIs: SPI1 on high speed APB2 and SPI2 on low speed APB1
- Full duplex synchronous transfers on 3 lines
- Simplex synchronous transfers on 2 lines with or without a bi-directional data line
- Programmable data frame size :8- or 16-bit transfer frame format selection
- Programmable data order with MSB-first or LSB-first shifting
- Master or slave operation
- Programmable bit rate: up to 18 MHz in Master/Slave mode
- NSS management by hardware or software for both master and slave: Dynamic change of Master/Slave Operations
- Support for DMA

Universal Synchronous Asynchronous Receiver Transmitter (USART)

USART Features

- Three USART: USART1 High speed APB2 and USART2,3 on Low speed APB1
- Fully-programmable serial interface characteristics:
  - Data can be 8 or 9 bits
  - Even, odd or no-parity bit generation and detection
  - 0.5, 1, 1.5 or 2 stop bit generation
  - Programmable baud rate generator
    - Integer part (12 bits)
    - Fractional part (4 bits)
  - Support hardware flow control (CTS and RTS)
- Dedicated transmission and reception flags (TxE and RxNE) with interrupt capability
- Support for DMA
  - Receive DMA request
  - Transmit DMA request

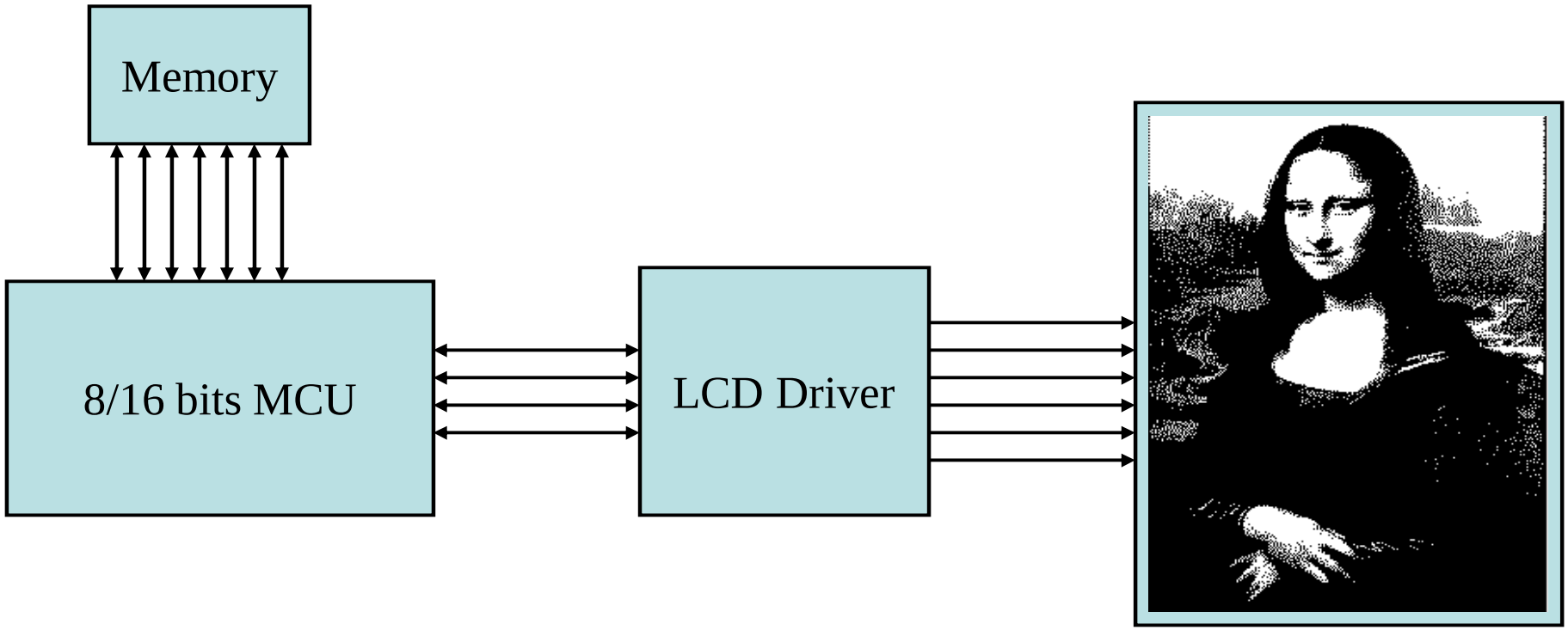


Universal Serial Bus interface (USB Device)

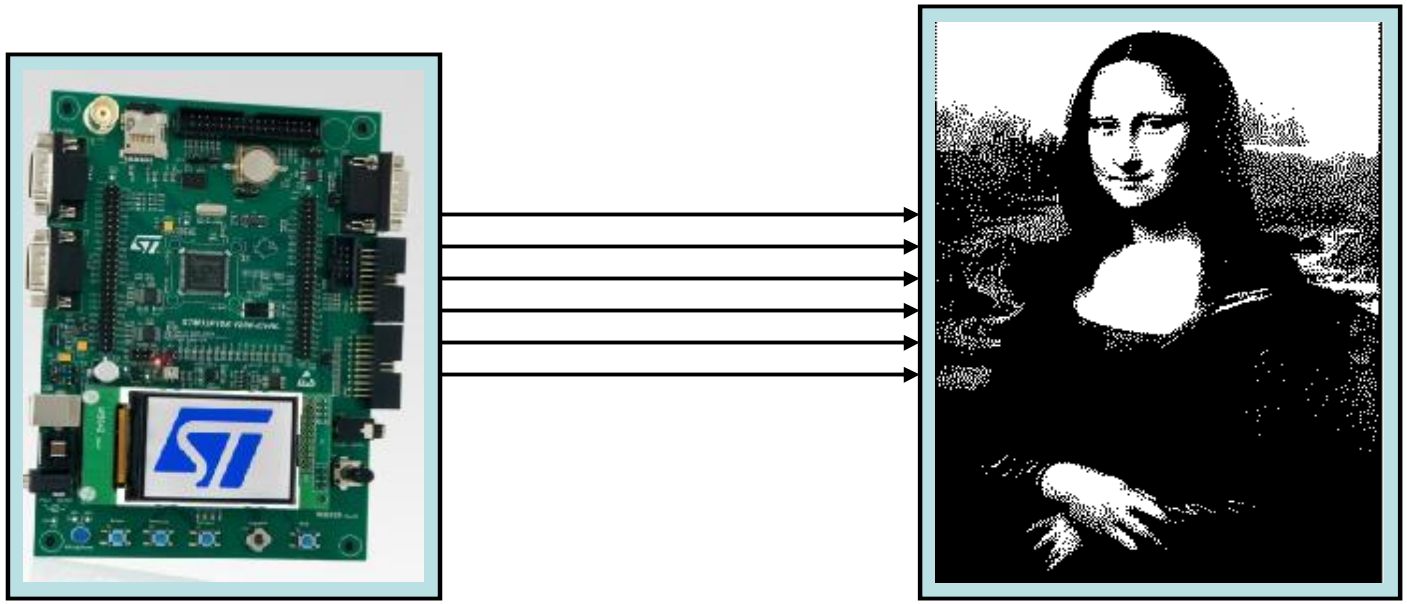
USB Developer's Kit

- Complete source file with documented, thoroughly tested C source code, compatible with major IDE toolsets for ARM
- Supports any flavor of USB firmware with:
  - Control transfer for generic device management tasks
  - Interrupt transfer with HID Mouse/Joystick
  - Bulk transfer with mass storage
  - Isochronous transfer with Voice Speaker/micro
  - DFU for firmware updates on USB
  - Virtual COM (CDC class) for emulation of RS232

Dot LCD System



Dot LCD System with STM32(Using GPIO)



Thank you!