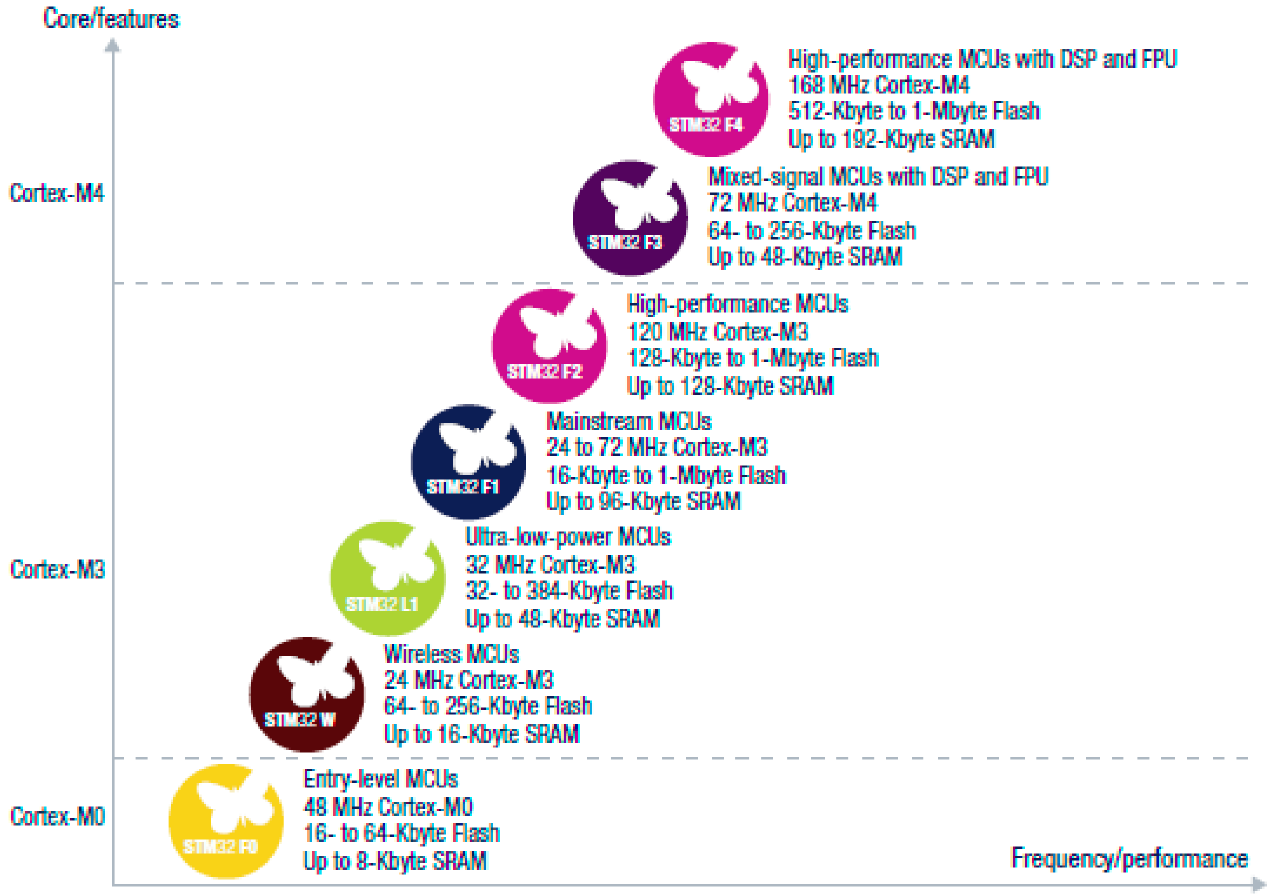
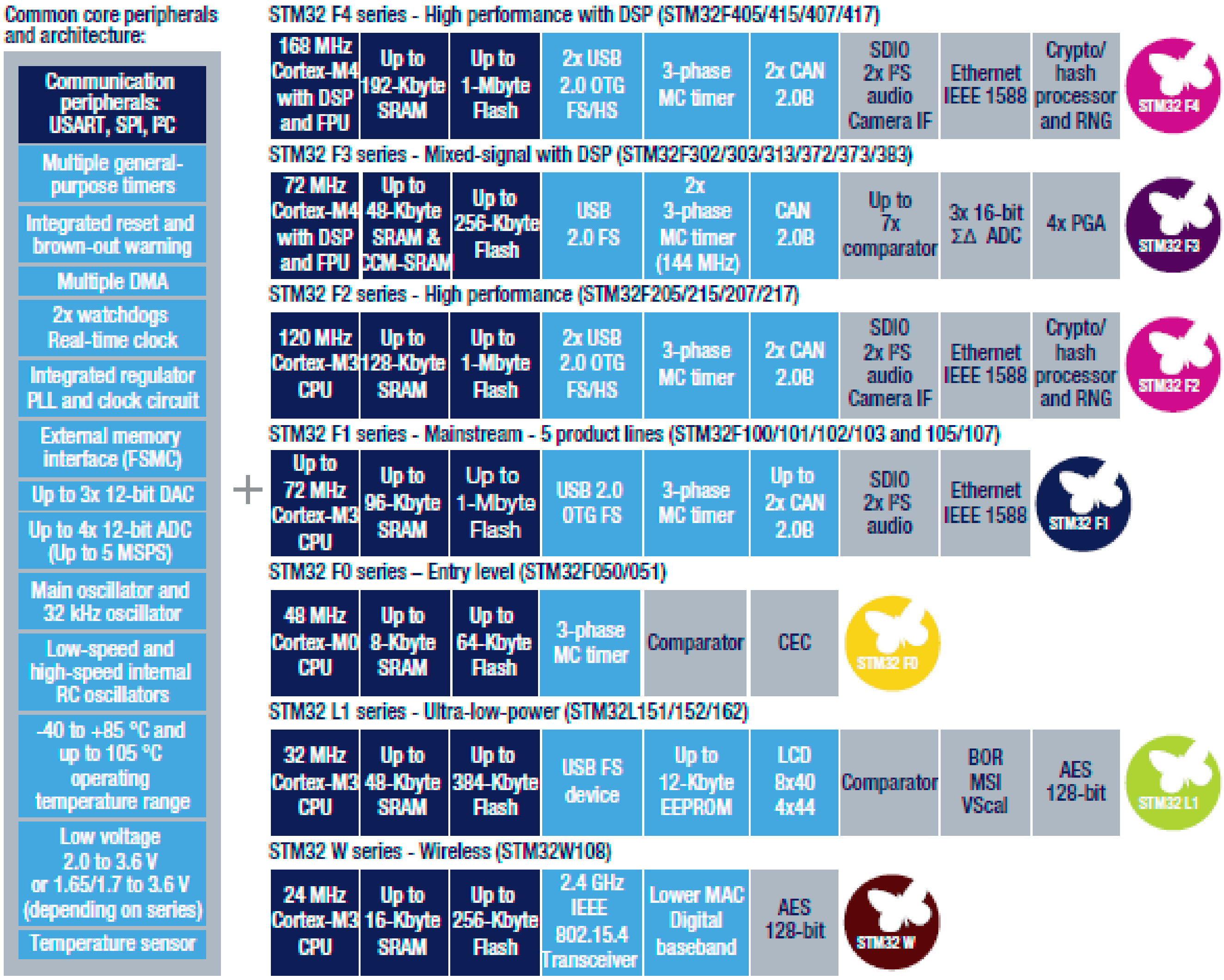
ARM Cortex board

ARM Cortex STM series



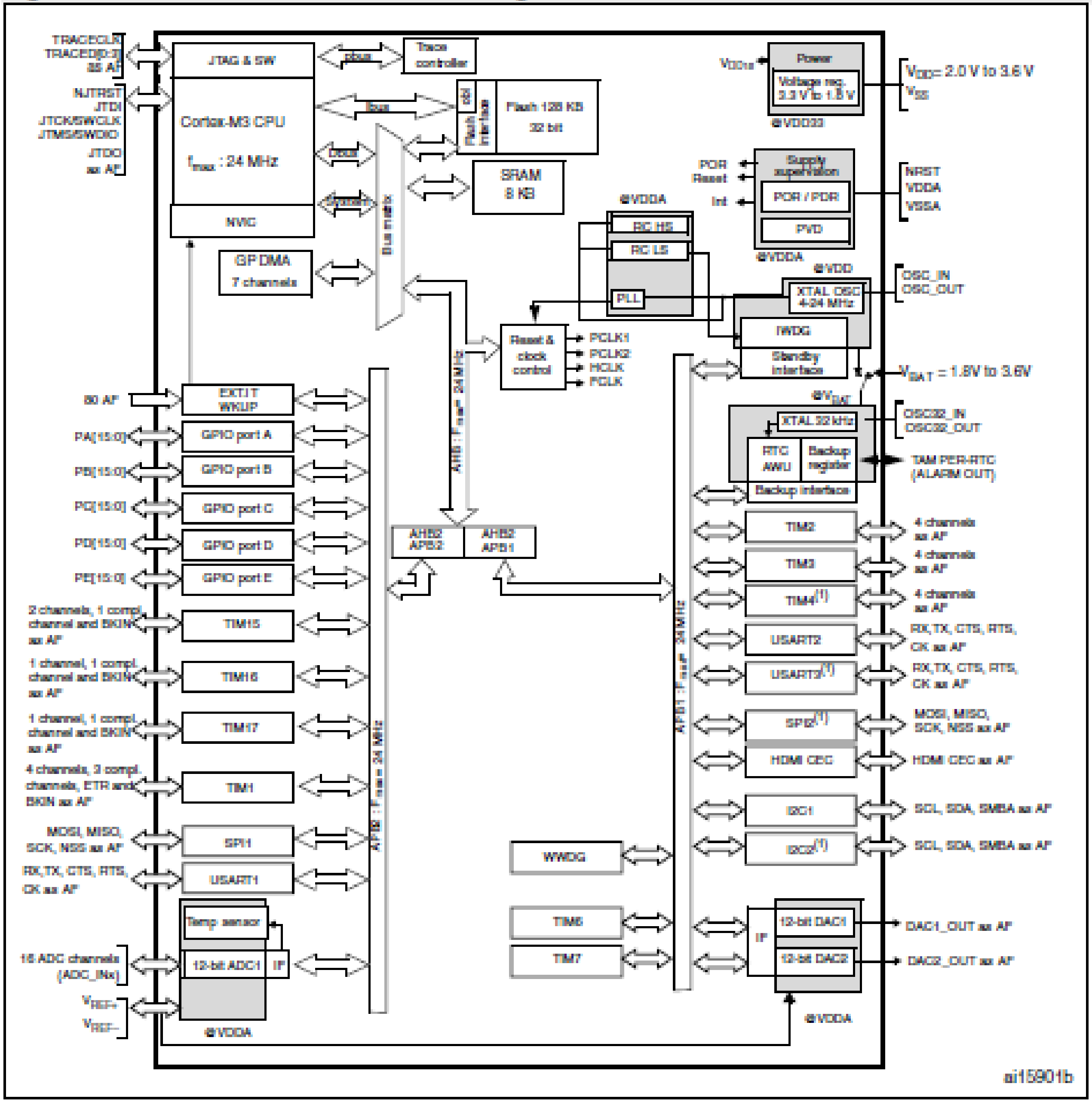
STM32 Series



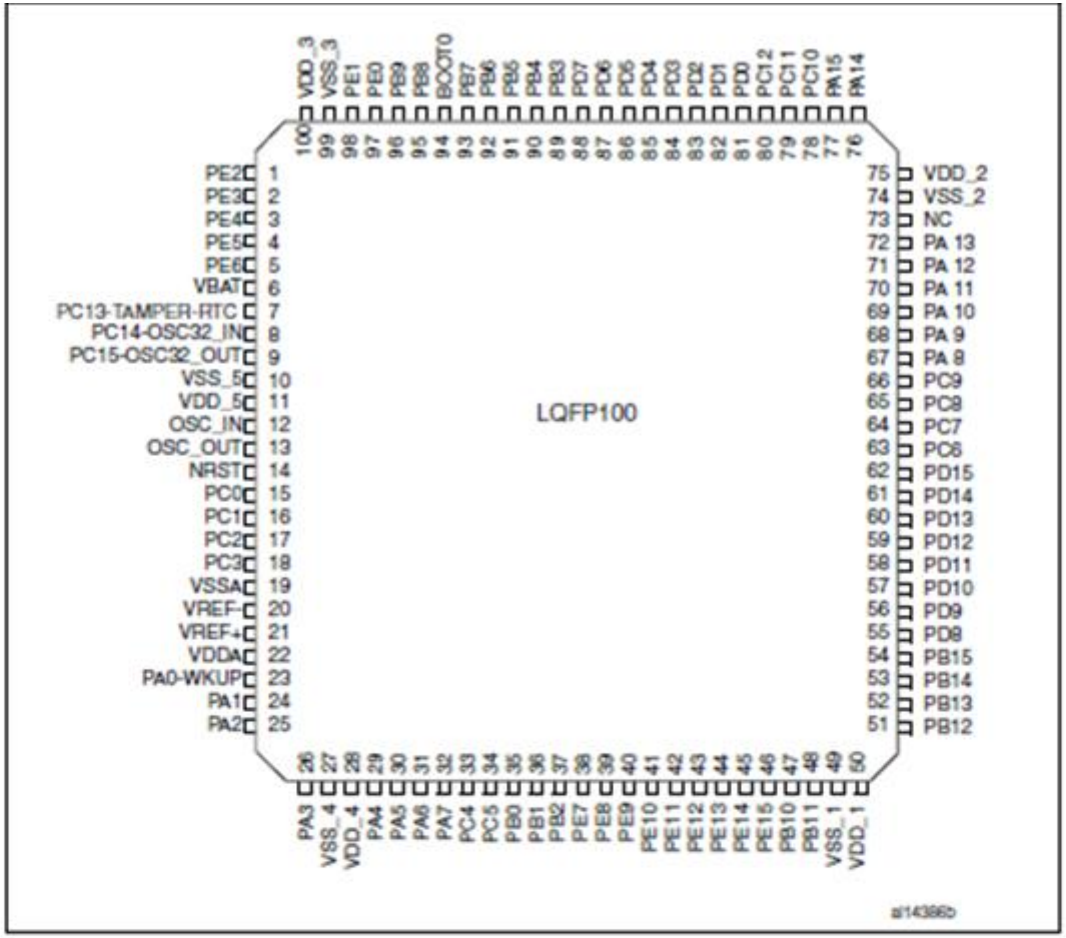
Abbreviation

- FS full speed
- HS high speed
- MC motor controller
- MSI multi speed internal oscillator
- RNG random number generator
- SDIO secure digital input output
- VScal voltage scaling
- DSC digital signal controller
- PGA programmable gain ampilier

STM32F100xx block diagram

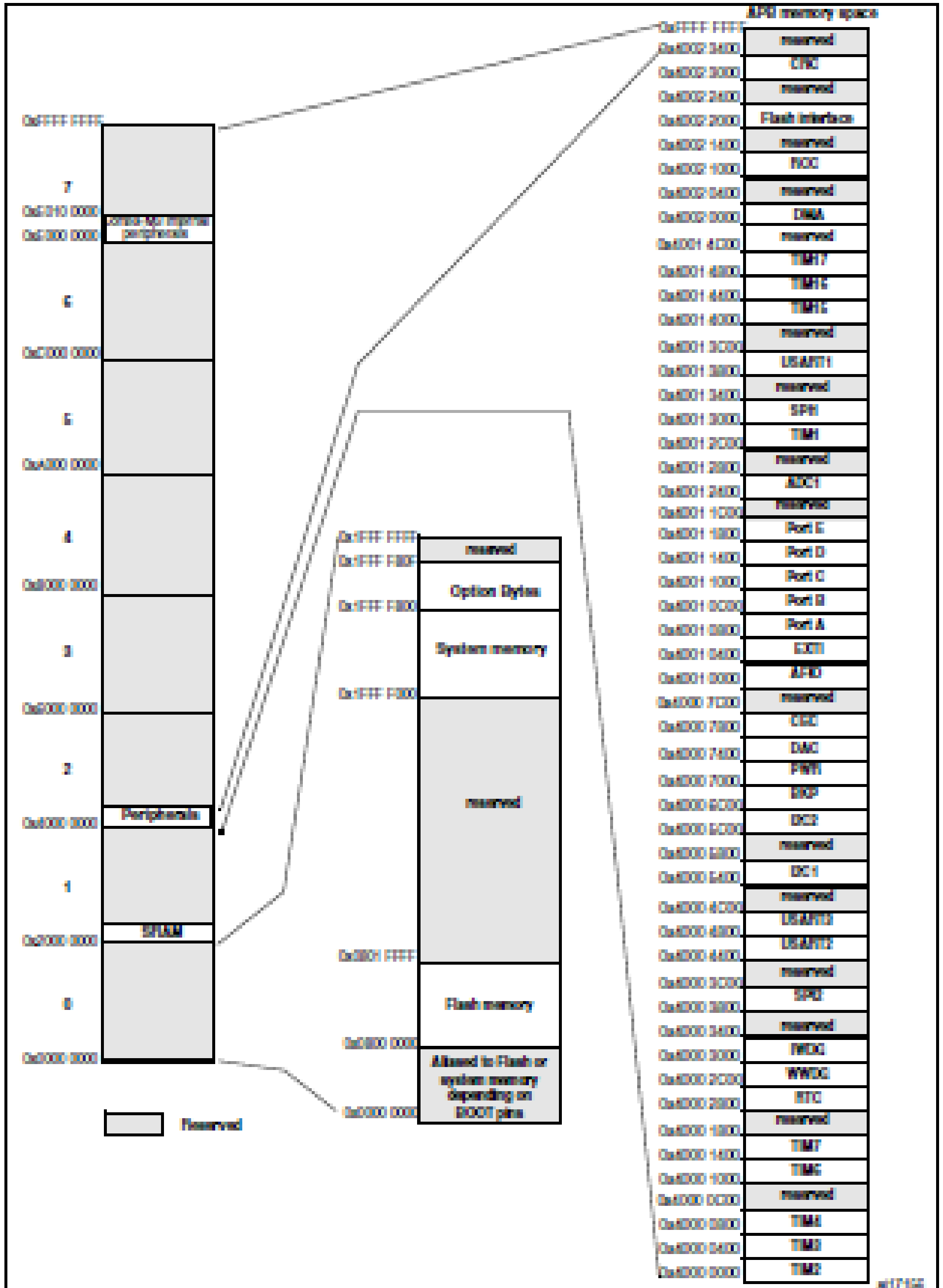


Pin-out



LQFP100
PE2 1
PE3 2
PE4 3
PE5 4
PE6 5
VBAT 6
PC13-TAMPER-RTC 7
PC14-OSC32_IN 8
PC15-OSC32_OUT 9
VSS_5 10
VDD_5 11
OSC_IN 12
OSC_OUT 13
NRST 14
PC0 15
PC1 16
PC2 17
PC3 18
VSSA 19
VREF- 20
VREF+ 21
VDDA 22
PA0-WKUP 23
PA1 24
PA2 25
PA3 26
VSS_4 27
VDD_4 28
PA4 29
PA5 30
PA6 31
PA7 32
PC4 33
PC5 34
PB0 35
PB1 36
PB2 37
PE7 38
PE8 39
PE9 40
PE10 41
PE11 42
PE12 43
PE13 44
PE14 45
PE15 46
PB10 47
PB11 48
VSS_1 49
VDD_1 50
51 PB12
52 PB13
53 PB14
54 PB15
55 PD8
56 PD9
57 PD10
58 PD11
59 PD12
60 PD13
61 PD14
62 PD15
63 PC6
64 PC7
65 PC8
66 PC9
67 PA 8
68 PA 9
69 PA 10
70 PA 11
71 PA 12
72 PA 13
73 NC
74 VSS_2
75 VDD_2
76 PA14
77 PA15
78 PC10
79 PC11
80 PC12
81 PD0
82 PD1
83 PD2
84 PD3
85 PD4
86 PD5
87 PD6
88 PD7
89 PB3
90 PB4
91 PB5
92 PB6
93 PB7
94 BOOT0
95 PB8
96 PB9
97 PE0
98 PE1
99 VSS_3
100 VDD_3

Memory map



APB memory space
Option Bytes
System memory
reserved
Flash memory
Aliased to Flash or system memory depending on BOOT pins
Peripherals
SRAM
Reserved
CRC
Flash interface
RCC
DMA
USART1
TIM1
ADC1
Port E
Port D
Port C
Port B
Port A
AFIO
DAC
PWR

Memory map (1)

Boundary address	Peripheral	Bus	Register map
0x5000 0000 - 0x5003 FFFF	USB OTG FS	AHB	Section 27.15.6 on page 861
0x4003 0000 - 0x4FFF FFFF	Reserved		
0x4002 8000 - 0x4002 9FFF	Ethernet		Section 28.8.5 on page 1021
0x4002 3400 - 0x4002 7FFF	Reserved		
0x4002 3000 - 0x4002 33FF	CRC		Section 3.4.4 on page 58
0x4002 2000 - 0x4002 23FF	Flash memory interface		
0x4002 1400 - 0x4002 1FFF	Reserved		
0x4002 1000 - 0x4002 13FF	Reset and clock control RCC		Section 6.3.11 on page 111
0x4002 0800 - 0x4002 0FFF	Reserved		
0x4002 0400 - 0x4002 07FF	DMA2		Section 10.4.7 on page 212
0x4002 0000 - 0x4002 03FF	DMA1		Section 10.4.7 on page 212
0x4001 8400 - 0x4001 7FFF	Reserved		
0x4001 8000 - 0x4001 83FF	SDIO		Section 21.9.16 on page 578

Memory map (2)

Boundary address	Peripheral	Bus	Register map
0x4001 5800 - 0x4001 7FFF	Reserved	APB2	
0x4001 5400 - 0x4001 57FF	TIM11 timer		*Section 15.5.14 on page 424*
0x4001 5000 - 0x4001 53FF	TIM10 timer		*Section 15.5.14 on page 424*
0x4001 4C00 - 0x4001 4FFF	TIM9 timer		*Section 15.5.14 on page 424*
0x4001 4000 - 0x4001 4BFF	Reserved		
0x4001 3C00 - 0x4001 3FFF	ADC3		*Section 11.12.15 on page 247*
0x4001 3800 - 0x4001 3BFF	USART1		*Section 26.6.8 on page 776*
0x4001 3400 - 0x4001 37FF	TIM8 timer		*Section 13.4.21 on page 333*
0x4001 3000 - 0x4001 33FF	SPI1		*Section 24.5 on page 692*
0x4001 2C00 - 0x4001 2FFF	TIM1 timer		*Section 13.4.21 on page 333*
0x4001 2800 - 0x4001 2BFF	ADC2		*Section 11.12.15 on page 247*
0x4001 2400 - 0x4001 27FF	ADC1		*Section 11.12.15 on page 247*
0x4001 2000 - 0x4001 23FF	GPIO Port G		*Section 8.5 on page 179*
0x4001 1C00 - 0x4001 1FFF	GPIO Port F		*Section 8.5 on page 179*
0x4001 1800 - 0x4001 1BFF	GPIO Port E		*Section 8.5 on page 179*
0x4001 1400 - 0x4001 17FF	GPIO Port D		*Section 8.5 on page 179*
0x4001 1000 - 0x4001 13FF	GPIO Port C		*Section 8.5 on page 179*
0x4001 0C00 - 0x4001 0FFF	GPIO Port B		*Section 8.5 on page 179*
0x4001 0800 - 0x4001 0BFF	GPIO Port A		*Section 8.5 on page 179*
0x4001 0400 - 0x4001 07FF	EXTI		*Section 9.3.7 on page 196*
0x4001 0000 - 0x4001 03FF	AFIO		*Section 8.5 on page 179*

Memory map(3)

Boundary address	Peripheral	Bus	Register map
0x4000 7800 - 0x4000 FFFF	Reserved	APB1	
0x4000 7400 - 0x4000 77FF	DAC		Section 12.5.14 on page 268
0x4000 7000 - 0x4000 73FF	Power control PWR		Section 4.4.3 on page 72
0x4000 6C00 - 0x4000 6FFF	Backup registers (BKP)		Section 5.4.5 on page 77
0x4000 6800 - 0x4000 6BFF	Reserved		
0x4000 6400 - 0x4000 67FF	bxCAN1		Section 23.9.5 on page 651
0x4000 6800 - 0x4000 6BFF	bxCAN2		Section 23.9.5 on page 651
0x4000 6000[1] - 0x4000 63FF	Shared USB/CAN SRAM 512 bytes		
0x4000 5C00 - 0x4000 5FFF	USB device FS registers		Section 22.5.4 on page 608
0x4000 5800 - 0x4000 5BFF	I2C2		Section 25.6.10 on page 733
0x4000 5400 - 0x4000 57FF	I2C1		Section 25.6.10 on page 733
0x4000 5000 - 0x4000 53FF	UART5		Section 26.6.8 on page 776
0x4000 4C00 - 0x4000 4FFF	UART4		Section 26.6.8 on page 776
0x4000 4800 - 0x4000 4BFF	USART3		Section 26.6.8 on page 776
0x4000 4400 - 0x4000 47FF	USART2		Section 26.6.8 on page 776
0x4000 4000 - 0x4000 3FFF	Reserved		
0x4000 3C00 - 0x4000 3FFF	SPI3/I2S		Section 24.5 on page 692
0x4000 3800 - 0x4000 3BFF	SPI2/I2S		Section 24.5 on page 692
0x4000 3400 - 0x4000 37FF	Reserved		
0x4000 3000 - 0x4000 33FF	Independent watchdog (IWDG)		Section 18.4.5 on page 463
0x4000 2C00 - 0x4000 2FFF	Window watchdog (WWDG)		Section 19.6.4 on page 468
0x4000 2800 - 0x4000 2BFF	RTC		Section 17.4.7 on page 457
0x4000 2400 - 0x4000 27FF	Reserved		
0x4000 2000 - 0x4000 23FF	TIM14 timer		Section 15.6.12 on page 433
0x4000 1C00 - 0x4000 1FFF	TIM13 timer		Section 15.6.12 on page 433
0x4000 1800 - 0x4000 1BFF	TIM12 timer		Section 15.6.12 on page 433
0x4000 1400 - 0x4000 17FF	TIM7 timer		Section 16.4.9 on page 446
0x4000 1000 - 0x4000 13FF	TIM6 timer		Section 16.4.9 on page 446
0x4000 0C00 - 0x4000 0FFF	TIM5 timer		Section 14.4.19 on page 389
0x4000 0800 - 0x4000 0BFF	TIM4 timer		Section 14.4.19 on page 389
0x4000 0400 - 0x4000 07FF	TIM3 timer		Section 14.4.19 on page 389
0x4000 0000 - 0x4000 03FF	TIM2 timer		Section 14.4.19 on page 389

ARM Bus

- Introduced by ARM Ltd in 1996
- Widely used as the on-chip bus

ARM BUS

- AMBA = ARM Memory Bus Architecture
- AHB = ARM High performance Bus
- APB = ARM Peripheral Bus
- AMBA-AHB connects ARM core with memory, external DRAM
- AMBA-APB interfaces ARM core with external low-speed I/O devices using AMBA-APB bridge

AMBA

- AMBA-AHB connects to 32-bit data and 32-bit address at high speed
- AHB maximum bps bandwidth is 16 times ARM processor clock
- AMBA-APB bridge is used to communicate AHB bus to APB bus
- The bridge communicates to memory through AMBA-AHB

AHB Bus

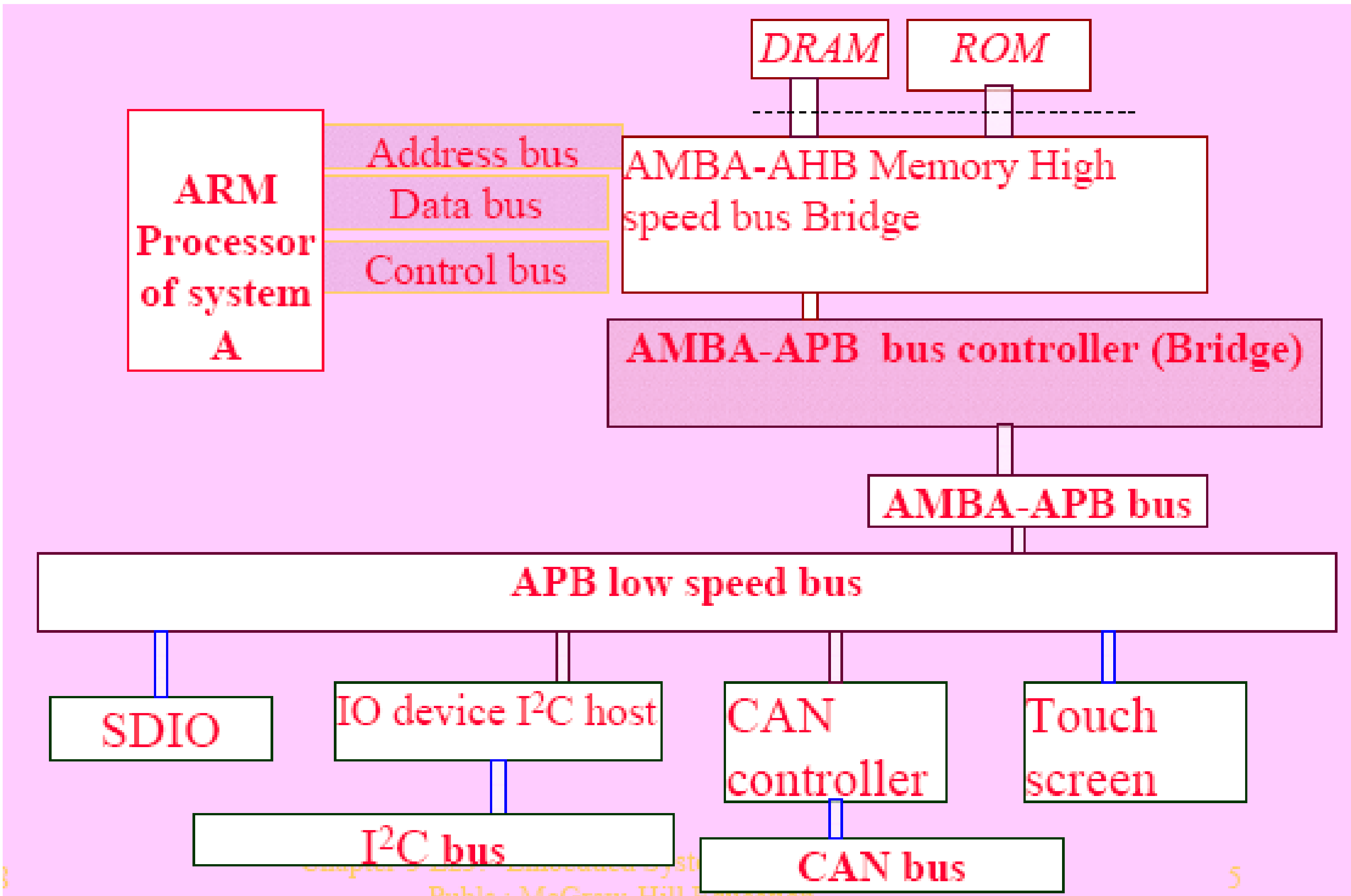
- AHB Bus are interconnected with:
  - Cortex core
  - Internal SRAM
  - Internal Flash memory
  - FSMC
  - AHB to APB
  - DMA
  - Ethernet DMA

APB bus

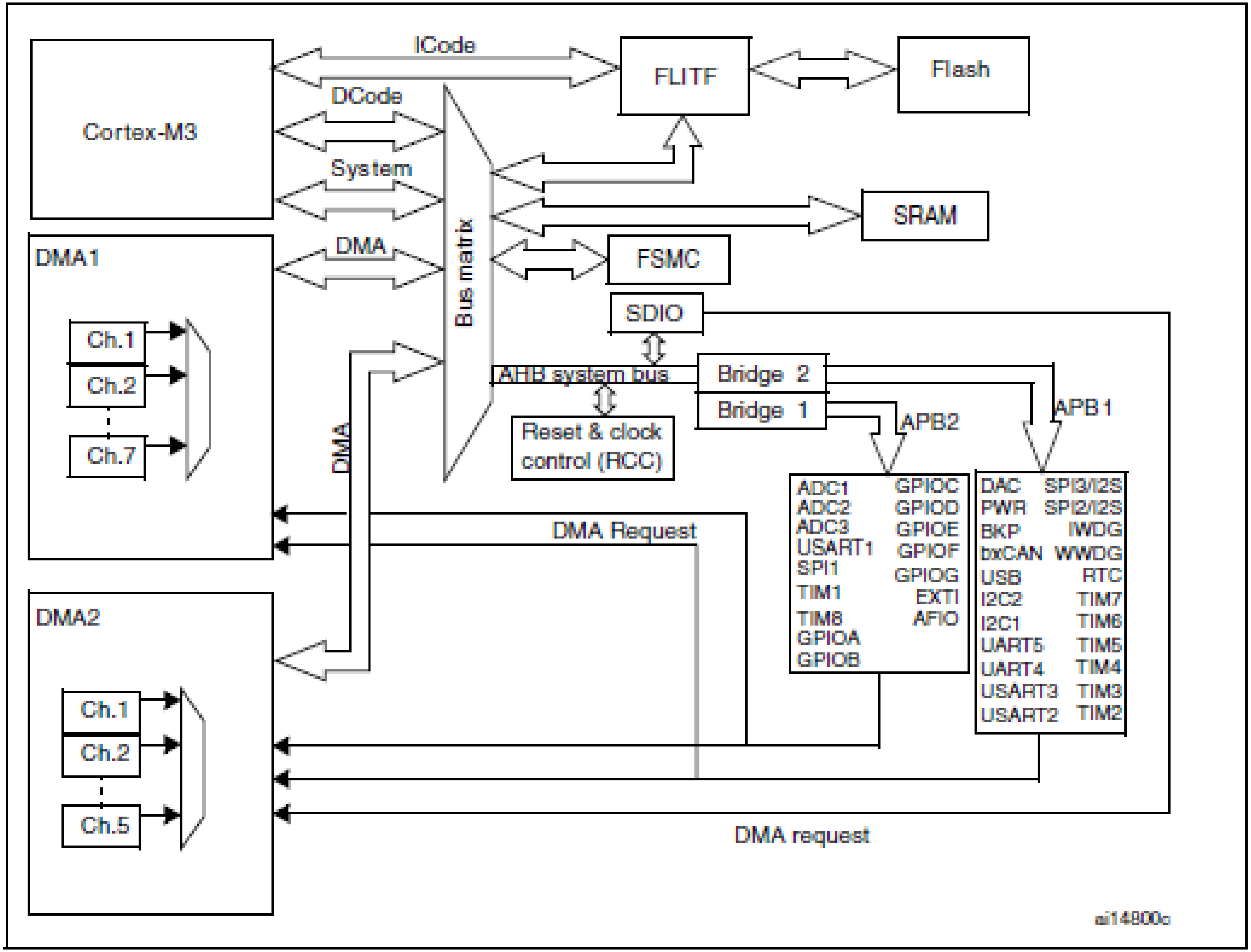
Connect:

- I^2C
- Touch screen
- SDIO
- MMC (multimedia-bus)
- USB
- CAN bus

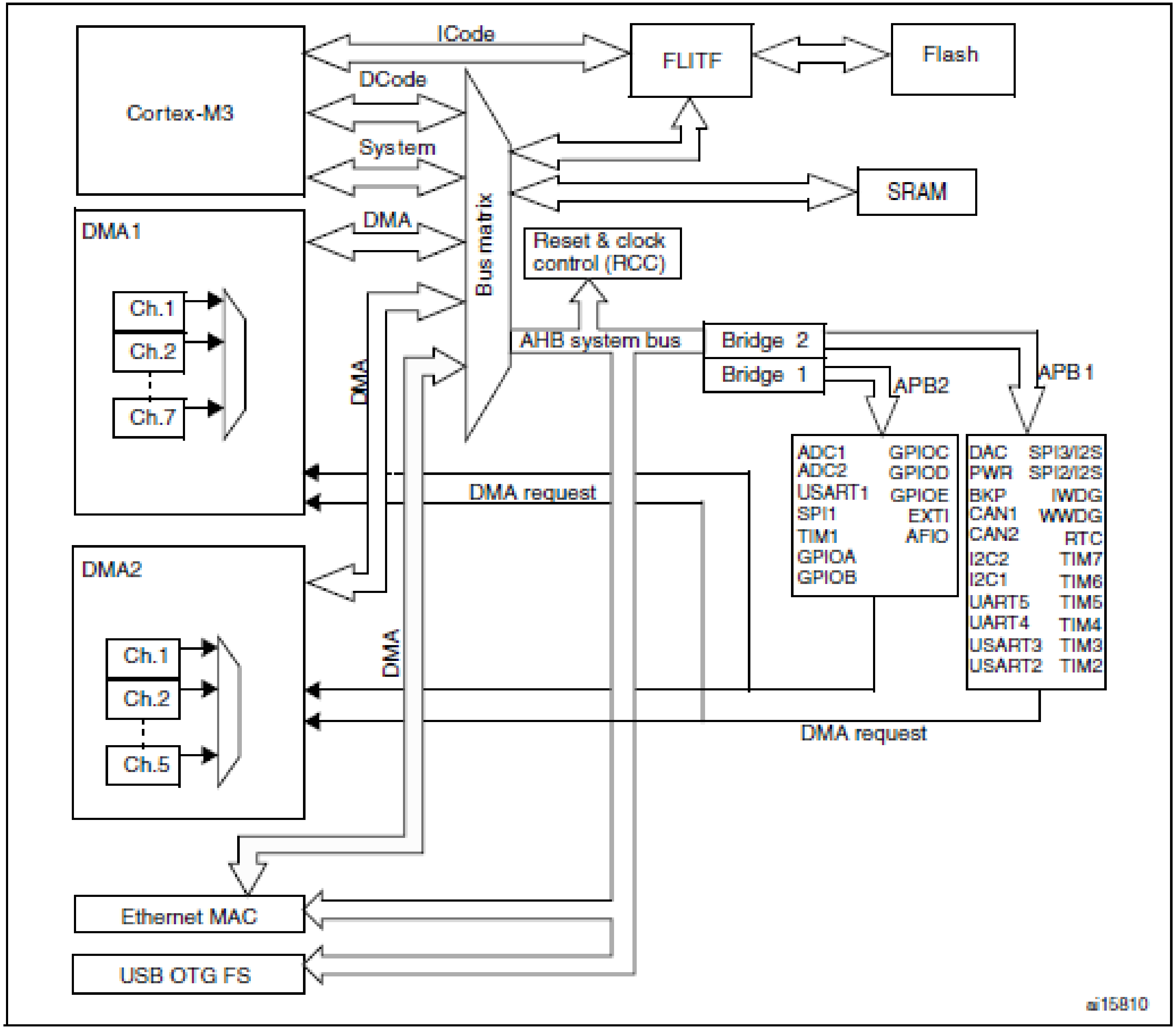
ARM BUS


DRAM
ROM
ARM Processor of system A
Address bus
Data bus
Control bus
AMBA-AHB Memory High speed bus Bridge
AMBA-APB bus controller (Bridge)
AMBA-APB bus
APB low speed bus
SDIO
IO device I2C host
CAN controller
Touch screen
I2C bus
CAN bus

System architecture



System architecture connecting devices



Bus system

- Icode bus : connects the instruction bus to the flash memory instruction interface
- Dcode bus : connects to flash memory data interface and debug access
- System bus : connects to bus matrix that manages the arbitration between core and DMA
- DMA bus: AHB master interface of the DMA to bus matrix
- Bus matrix: arbitrations between different buses
- AHB/APB bridge: bridge between AHB and APB

APB1 is limited at 36 MHz and APB2 is at 72 MHz

AHB and APB

- AHB = Advance High Performance Bus with max speed = 72 Mhz

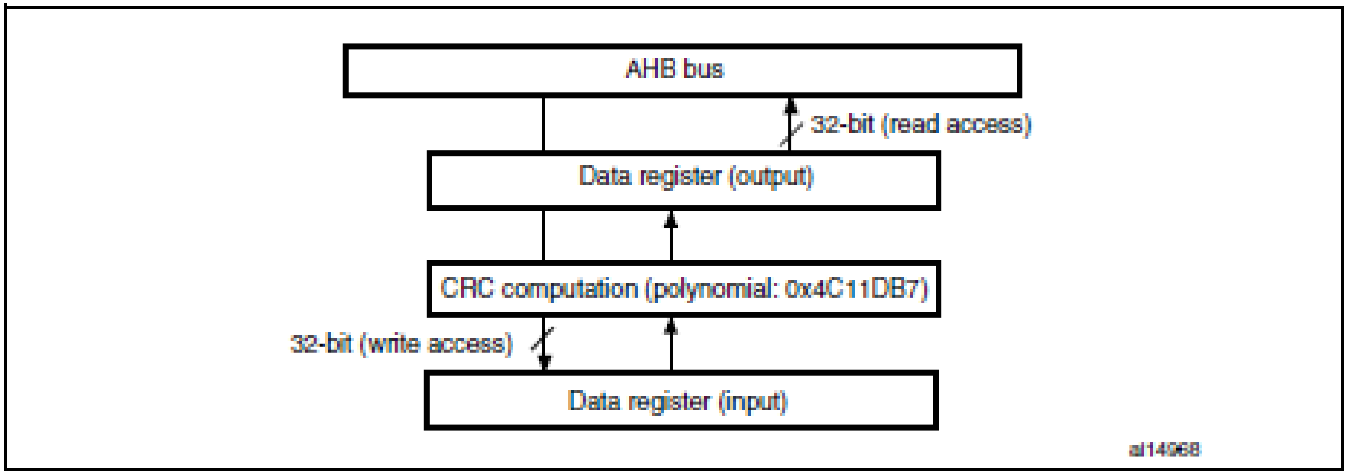
- APB = Advance Peripheral Bus with APB1 max speed = 36 Mhz and APB2 max speed = 72 MHz

- FSMC = Flexible static memory controller (able to interface with synchronous and asynchronous)

Cyclic redundancy checking (CRC)

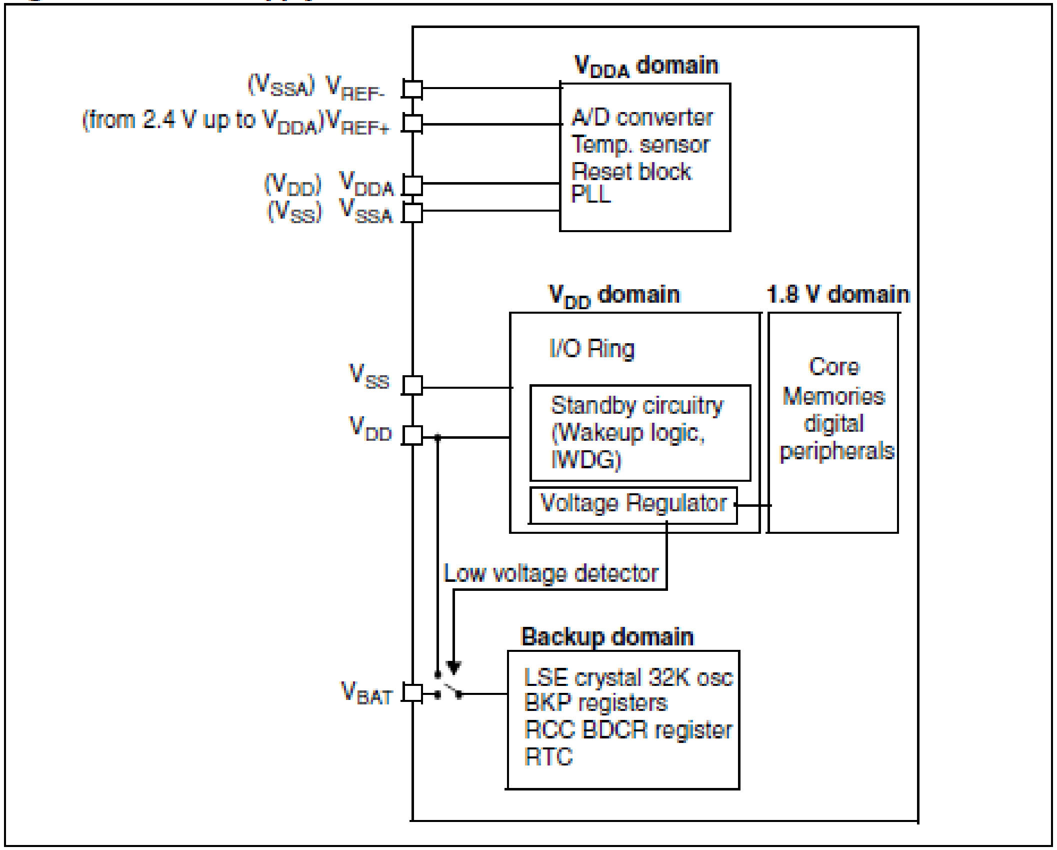
- Use to get a CRC code from a 32 bit data word
- Use to verify data transmission and storage integrity
- Example of CRC calculation block diagram
- Example of CRC polynomial:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$



Power supply overview

- The device requires 2-3.6V voltage supply.



Power supply

- Voltage regulator to provide 1.8V for core and memory

- Reset and Clock control, real time clock, and LSE (Low speed external crystal) oscillator are supported by backup domain power source

Low power mode

- Normal condition: run mode
- Low power mode:
  - Sleep mode

    CPU clocks are off, all peripherals are running
  - Stop mode

    all clocks are off
  - Standby mode

    1.8 V power domain are off (enter backup domain)

Low power mode

Mode name	Entry	wakeup	Effect on 1.8V domain clocks	Effect on V_{DD} domain clocks	Voltage regulator
Sleep (Sleep now or Sleep-on - exit)	WFI	Any interrupt	CPU clock OFF no effect on other clocks or analog clock sources	None	ON
	WFE	Wakeup event			
Stop	PDDS and LPDS bits + SLEEPDEEP bit + WFI or WFE	Any EXTI line (configured in the EXTI registers)	All 1.8V domain clocks OFF	HSI and HSE oscillators OFF	ON or in low-power mode (depends on *Power control register (PWR_CR)*)
Standby	PDDS bit + SLEEPDEEP bit + WFI or WFE	WKUP pin rising edge, RTC alarm, external reset in NRST pin, IWDG reset			OFF

WFI (wait for interrupt) and WFE (wait for event) are special ARM instructions

Reset

There are 3 types of reset:

- System reset
- Power reset
- Backup domain reset

System reset

Set all registers to their reset values except reset flags and registers in the backup domain

It is generated when:

- External reset pin is triggered
- Window watchdog ends of count condition (WWDG reset)
- Independent watchdog ends of count condition (IWDG reset)
- Software reset (SW reset)
- Low power management reset

Power reset

Set all registers to their reset values except registers in the backup domain

It is generated when:

- Power on/ Power down reset
- When exiting standby mode

Backup domain reset

Effects only the backup domain

It is generated when:

- Software reset
- V_{DD} or V_{BAT}, if both supplies have previously been off

STM32F10xxx board

Clocks

Three different clock sources can be used to drive SYSCLK

- HSI (High speed Internal) oscillator clock
- HSE (High speed External) oscillator clock
- PLL clock

The devices have two secondary clock sources

- 40 kHz low speed internal (LSI) RC for the independent watchdog and optionally drive RTC
- 32.768 KHz low speed external crystal (LSE) which optionally drive RTC

Each clock source can be switched on/off independently

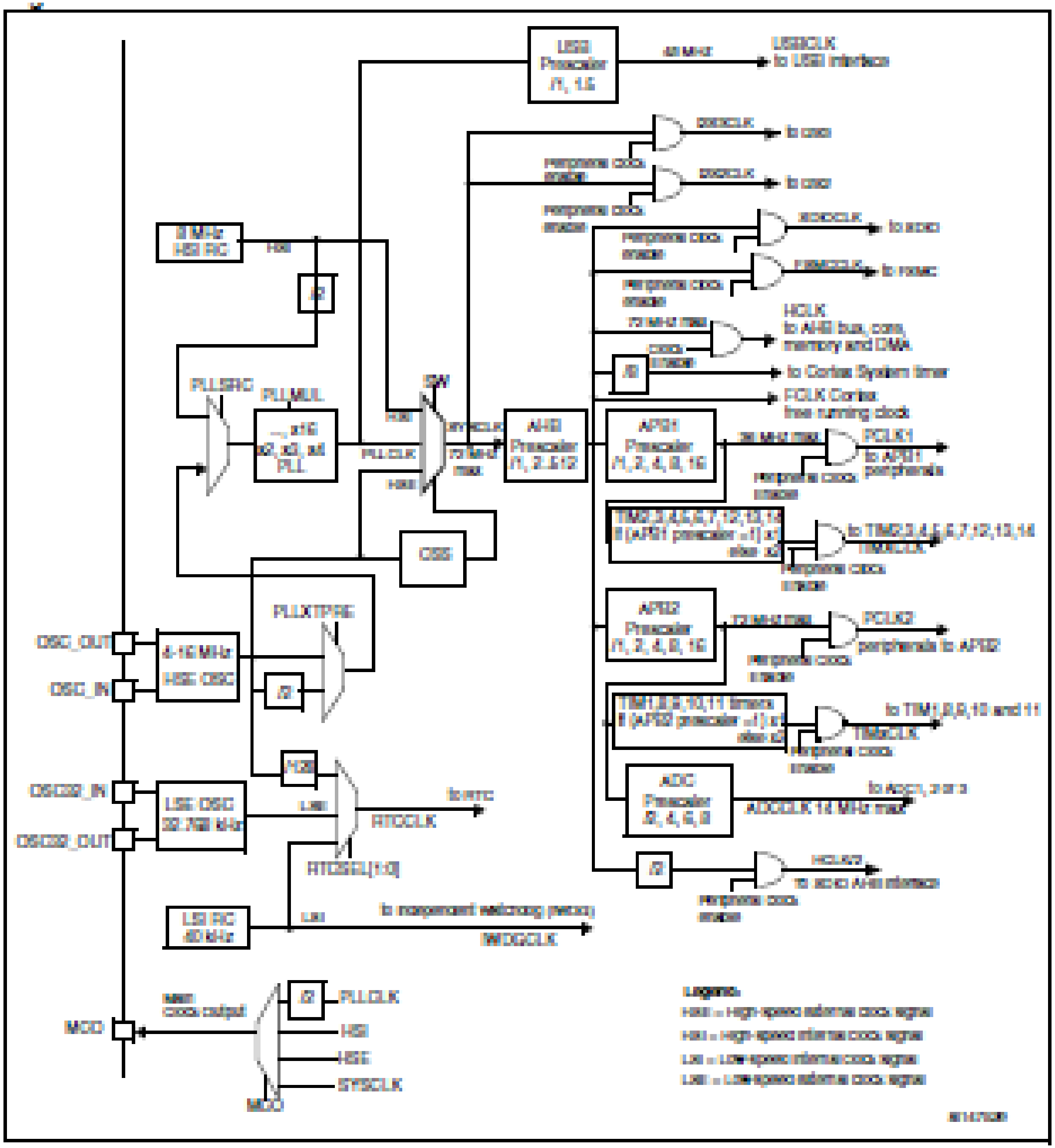
CLOCK

- HSI clock: provides a clock source at low cost (no external component)
- HSE clock: provides higher accuracy
- LSE clock: low power, but high accuracy
- LSI clock: low power for stop and standby mode

SYSCLOCK

- After a system reset, HSI is selected as system clock
- A switch from one clock source to another occurs only if target clock source is ready
- If a target is not yet ready, the switch will occur after the target is ready

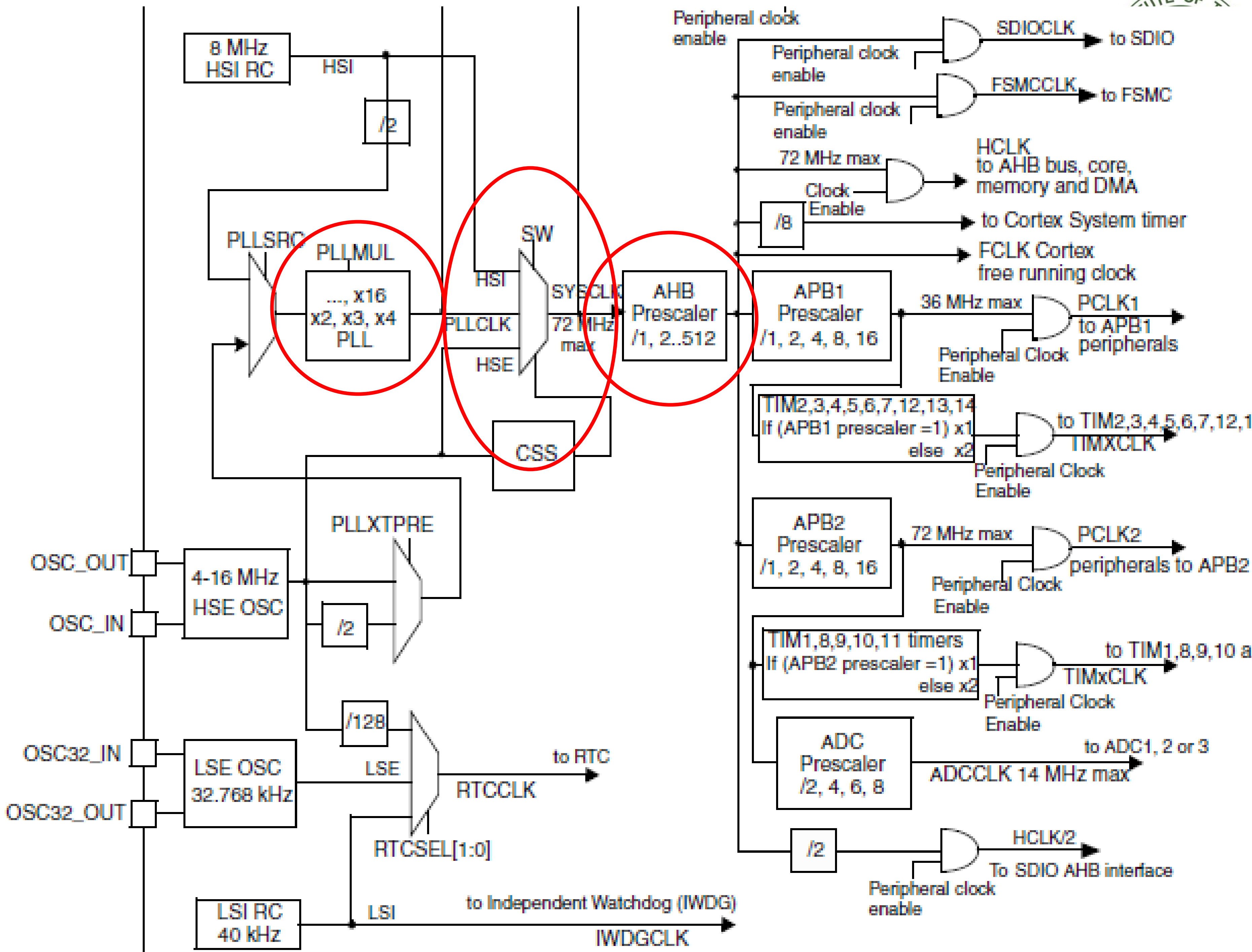
Clock tree



USB Prescaler
USBCLK
to USB interface
8 MHz HSI RC
HSI
/2
PLLSRC
PLLMUL
PLL
SW
PLLCLK
HSE
SYSCLK
AHB Prescaler
APB1 Prescaler /1, 2, 4, 8, 16
HCLK
to AHB bus, core, memory and DMA
to Cortex System timer
FCLK Cortex free running clock
PCLK1
CSS
PLLXTPRE
OSC_OUT
OSC_IN
4-16 MHz HSE OSC
APB2 Prescaler /1, 2, 4, 8, 16
PCLK2
ADC Prescaler /2, 4, 6, 8
OSC32_IN
OSC32_OUT
LSE OSC 32.768 kHz
/128
LSE
to RTC
RTCCLK
RTCSEL[1:0]
LSI RC 40 kHz
LSI
IWDGCLK
MCO
/2
PLLCLK
HSI
HSE
SYSCLK
Legend:

Our board

- HSE is 8 MHz crystal
- HSI is 8 MHz
- PLL = HSE x 9 = 72 MHz (Max. freq.)
- SYSCLK can be selected from PLL, HSE, or HSI



8 MHz
HSI RC
HSI
/2
PLLSRC
PLLMUL
..., x16
x2, x3, x4
PLL
PLLCLK
SW
HSI
HSE
SYSCLK
72 MHz
max
AHB
Prescaler
/1, 2..512
CSS
PLLXTPRE
OSC_OUT
OSC_IN
4-16 MHz
HSE OSC
/2
/128
OSC32_IN
OSC32_OUT
LSE OSC
32.768 kHz
LSE
to RTC
RTCCLK
RTCSEL[1:0]
LSI RC
40 kHz
LSI
to Independent Watchdog (IWDG)
IWDGCLK
Peripheral clock
enable
SDIOCLK
to SDIO
Peripheral clock
enable
FSMCCLK
to FSMC
Peripheral clock
enable
72 MHz max
HCLK
to AHB bus, core,
memory and DMA
Clock
Enable
/8
to Cortex System timer
FCLK Cortex
free running clock
APB1
Prescaler
/1, 2, 4, 8, 16
36 MHz max
PCLK1
to APB1
peripherals
Peripheral Clock
Enable
TIM2,3,4,5,6,7,12,13,14
If (APB1 prescaler =1) x1
else x2
to TIM2,3,4,5,6,7,12,1
TIMXCLK
Peripheral Clock
Enable
APB2
Prescaler
/1, 2, 4, 8, 16
72 MHz max
PCLK2
peripherals to APB2
Peripheral Clock
Enable
TIM1,8,9,10,11 timers
If (APB2 prescaler =1) x1
else x2
to TIM1,8,9,10 a
TIMxCLK
Peripheral Clock
Enable
ADC
Prescaler
/2, 4, 6, 8
to ADC1, 2 or 3
ADCCLK 14 MHz max
/2
HCLK/2
To SDIO AHB interface
Peripheral clock
enable

Real Time Clock (RTC)

- RTC can be either HSE/128, LSE or LSI clock which is set by the backup domain control register

- If LSE is selected, RTC continues to work even if V_{DD} is powered off, provided that V_{BAT} is maintained

Watchdog and clockout

- If independent watchdog is started either by hardware or software, LSI oscillator is forced ON and will supply the watchdog

- Microcontroller clock output (MCO pin) allows clock to be outputted by selected from: SYSCLOCK, HSI, HSE, PLL/2

Output clock

- HCLK : for AHB bus, core, memory and DMA
- FCLK: free running clock
- PCLK1: to APB1
- PCLK2: to APB2
- TIM1CLK: Timer 1
- TIMxCLK: Timer 2,3,4
- ADCCLK: to ADC module

Reset and Clock Control (RCC)

- RCC is a set of register to setup for reset and clock control

```
void RCC_Configuration(void)
{
    ErrorStatus HSEStartUpStatus;        // for debug info
    RCC_DeInit();                        //  initiialize RCC to default
    RCC_HSEConfig(RCC_HSE_ON);   // enable HSE
    HSEStartUpStatus = RCC_WaitForHSEStartUp();/* Wait till HSE is ready */
    if(HSEStartUpStatus == SUCCESS){
        FLASH_SetLatency(FLASH_Latency_2); // wait state 2 for 72MHz
        RCC_HCLKConfig(RCC_SYSCLK_Div1);  // HCLK = SYSCLK
        RCC_PCLK2Config(RCC_HCLK_Div1);  // PCLK2 = HCLK
        RCC_PCLK1Config(RCC_HCLK_Div2); /* PCLK1 = HCLK/2 */
        RCC_PLLConfig(RCC_PLLSource_HSE_Div1, RCC_PLLMul_9);        /* PLLCLK =
    8MHz * 9 = 72 MHz */
        RCC_PLLCmd(ENABLE); /* Enable PLL */
        while(RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET){}/*Wait till PLL is ready
        RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK); /* Select PLL as system clock
    source */
        while(RCC_GetSYSCLKSource() != 0x08){} /* Wait till PLL is used as system clock
    source */
    }
}
```

Boot configuration

- In STM32F10xxx, 3 different boot modes are provided and can be selected from the pin.

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as boot space
0	1	System memory	System memory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space

Boot mode

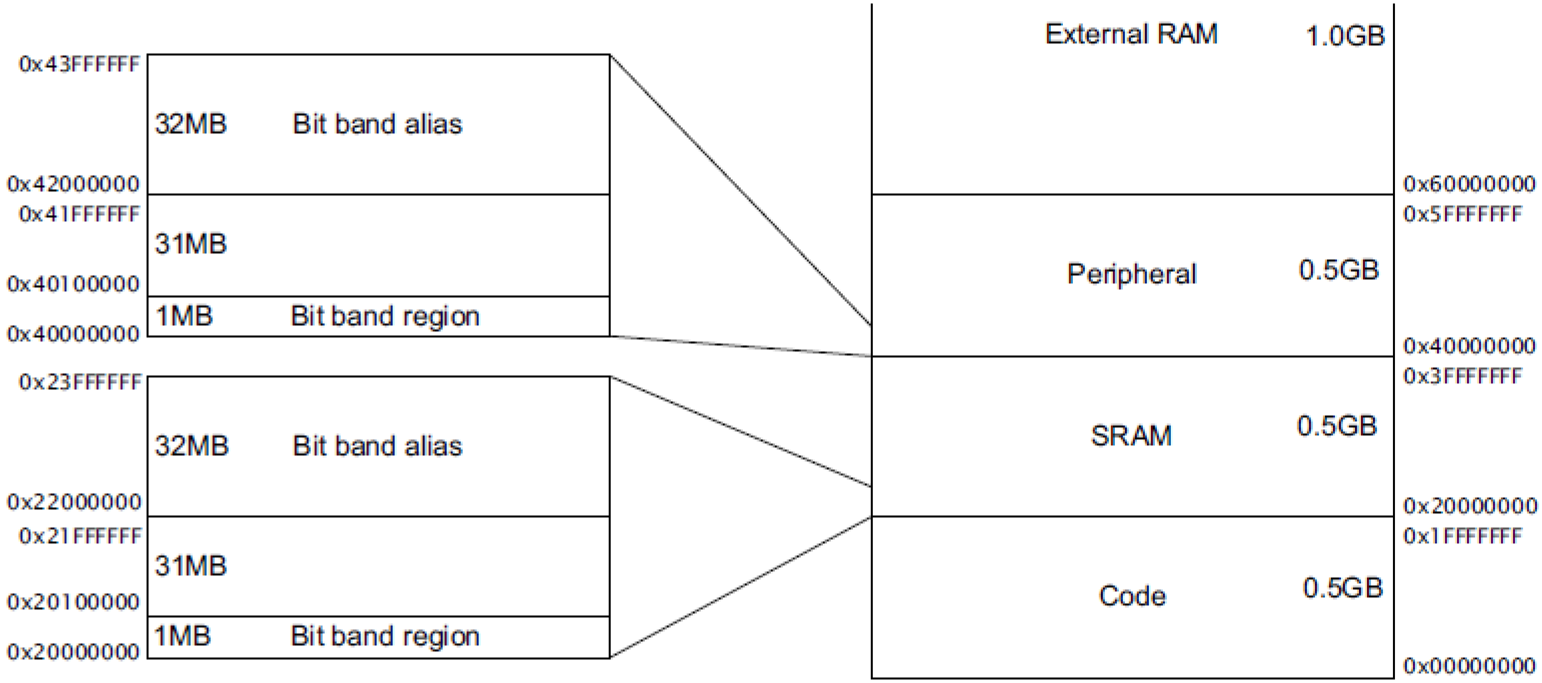
- Boot from main flash memory: flash memory is aliased to address 0x0000 0000, but can also accessed from address 0x0800 0000
- Boot from system memory: it can be acces from aliasing address: 0x0000 0000 or 0x1FFF B000
- Boot from embedded SRAM: only at address 0x2000 0000

Note: when booting from SRAM, NVIC exception table has to be relocated

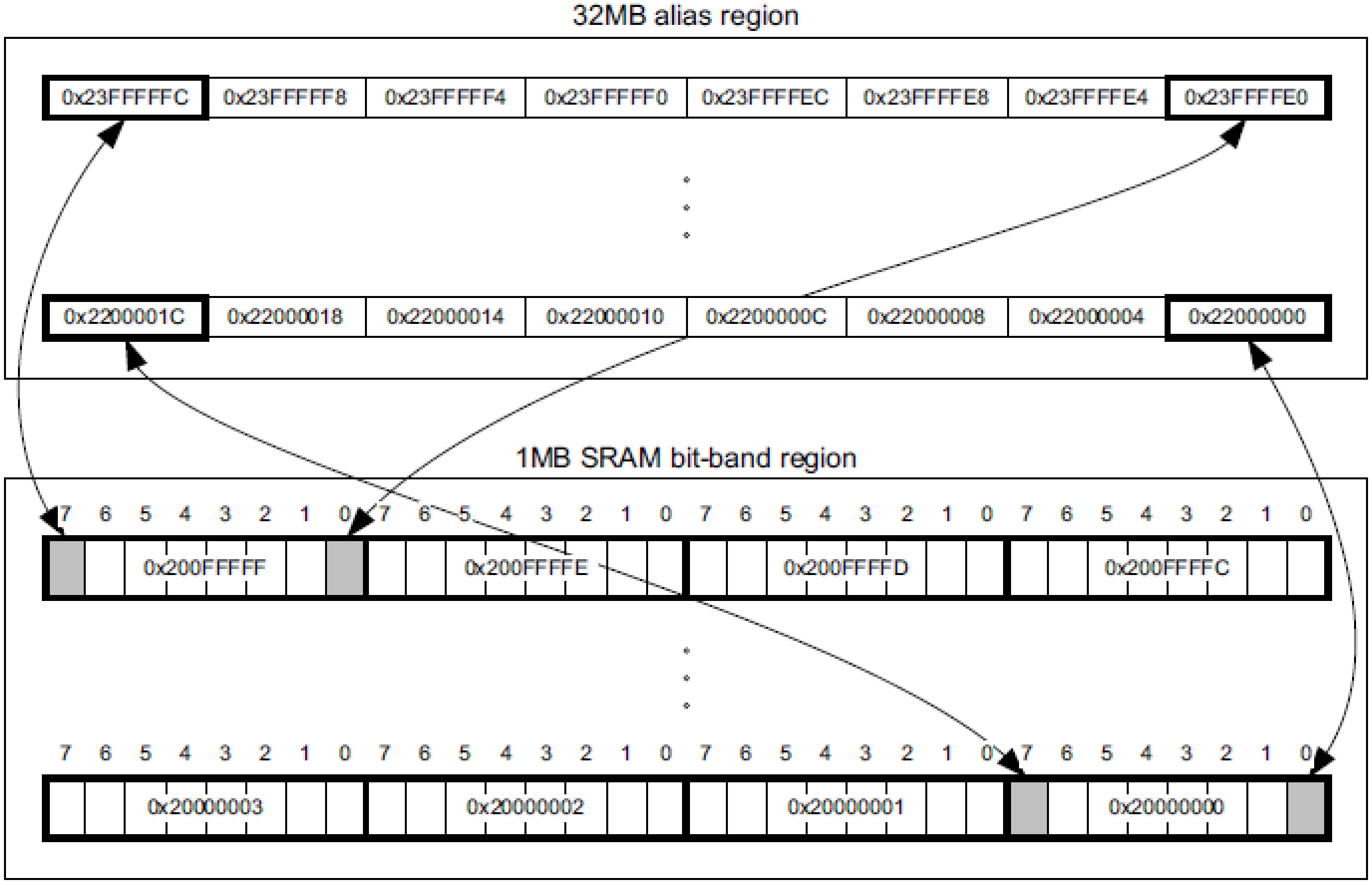
Bit banding

- Bit band operation support allows a single load/store operation to access a single data bit
- In ARM Cortex M3, this is supported in two predefined memory regions call bit-band regions
- One of them is located in the first 1 MB of the SRAM and the other is located in the first 1 MB of the peripheral region

Bit-band region



Example of bit-band access



Data mapping

Bit_word_address = bit_band_base + (byte_offset*32) + (bit_number x 4)

Example: To access bit 2 of the byte located in SRAM address 0x20000300

= 0x22000000 + (0x300*32) + 2*4

= 0x22008008

Questions?