

STM32H7 技术培训

- 探讨STM32H7xx设备新的架构,外设和系统的块
- 结束的时候你将能够
 - 理解STM32H7新产品特性
 - 安装开发工具, 运行demo,和外设的example
 - 能够介绍STM32H7xx

STM32H7xx - 系统概述

系统概述

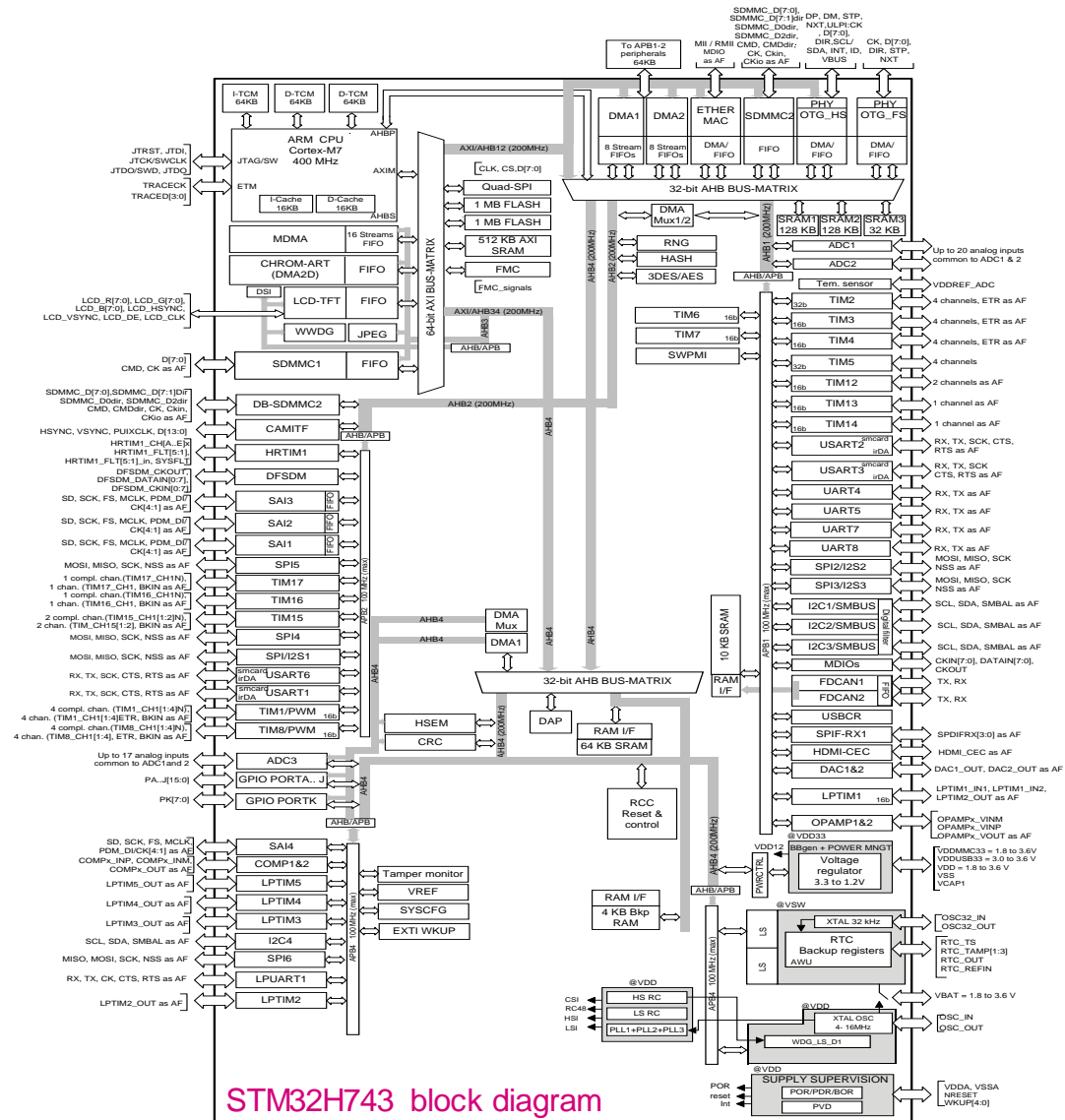
版次 1



框图

• 概述

- Cortex®-M7 core 频率 最高达 400 MHz
 - 16 Kbytes I-\$&D-\$
 - 16 MPU 区域
 - 单精度和双精度FPU
- 3x 互连矩阵
- 4 DMA 控制器用来减轻CPU负载
- 最多35通信外围设备
- 11 模拟外设
- 最多22 个定时器和看门狗
- 增强的时钟管理



STM32H743 block diagram

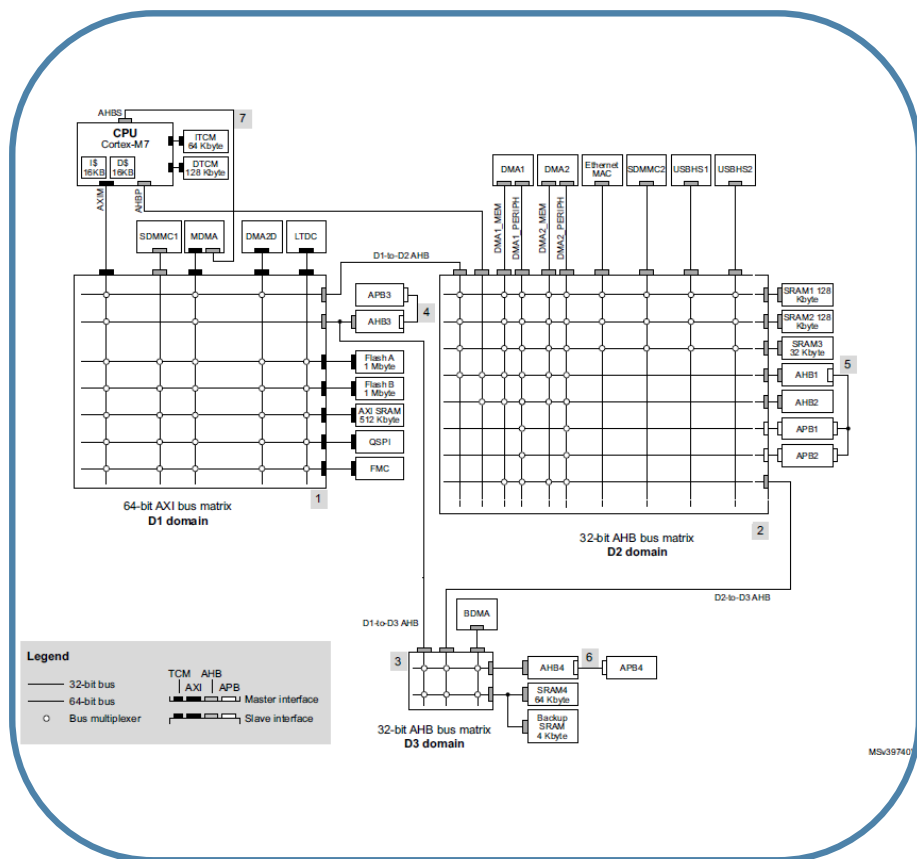
- 40 纳米制程
- 在400 MHz有出色的性能:低动态功率
 - Coremark程序在flash运行, 频率 400 MHz (外围设备关闭), 只有 87mA
 - 综合了高性能和低动态功耗
 - 停止模式223 uA
 - 供电 1.62 v(最小)3.6 v(最大) 。
 - 软件可配置的 Regulator Bypass 选项
- 高效的系统操作,和高速外围设备并行工作对系统性能影响不大

概述

- 总线主和总线从之间的内部连接是通过：
 - 一个AXI总线矩阵，
 - 两个AHB总线矩阵，
 - 和总线之间的桥

应用好处

- 高效的系统和高速外围设备可以同时操作
- 公平仲裁与循环算法
- 使用QoS能力进行AXI仲裁



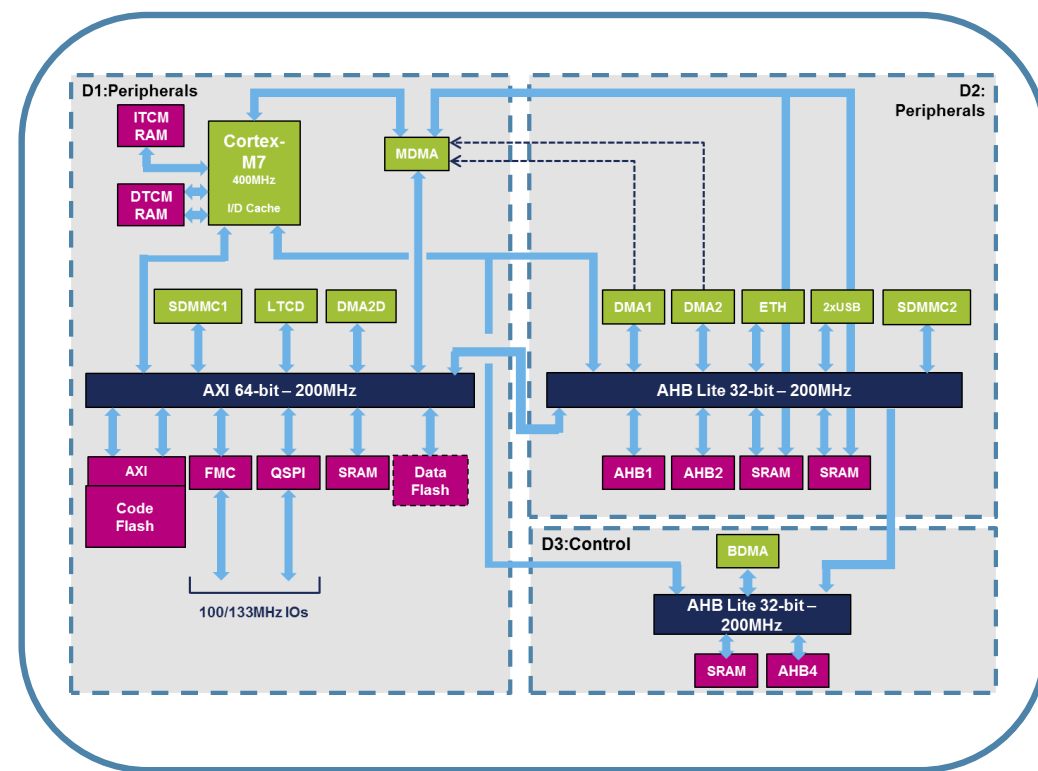
- 三个总线矩阵允许高效的系统和高速外围设备同时操作
 - 当几个主设备同时是激活的,能移除总线的堵塞
 - **AHB**总线矩阵使用公平仲裁的循环算法,
 - **AXI**总线矩阵使用优先级驱动仲裁 (**QoS**能力),
- 用于子系统自主操作的专用**DMA**控制器 (**BDMA**)
- 在子系统中, 当外围设备的不激活时, 内部时钟可以关闭。

源和目的地

8

大量的互连提供可能性

- 总线矩阵
 - AXI总线矩阵在D1域
 - AHB总线矩阵在D2域和D3域
- Bus-to-bus桥梁
 - 5 x AHB-to-APB, 2 x AXI-to-AHB
- 内部域总线
 - D2-to-D1 AHB
 - D1-to-D2 AHB
 - D1-to-D3 AHB
 - D2-to-D3 AHB



10多个总线主控

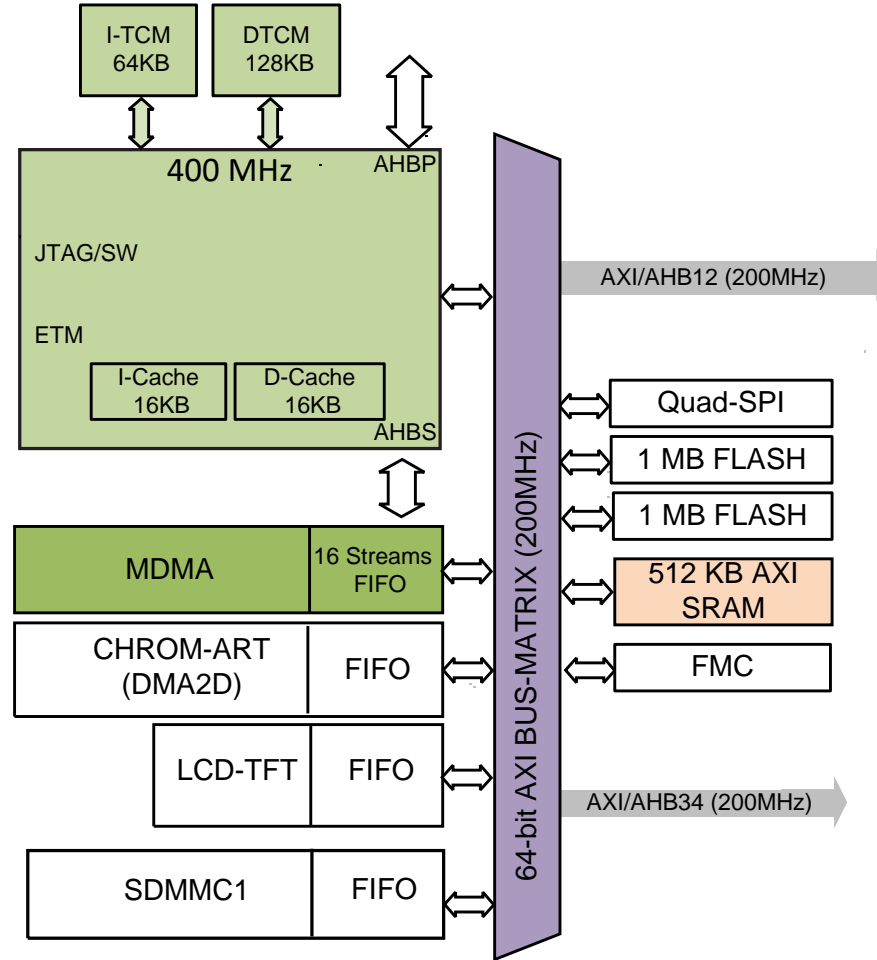
- CPU总线
- 高性能外设
 - SDMMC1, SDMMC2
 - MDMA
 - DMA1 和 DMA2
 - BDMA
 - Chrom-Art Accelerator™ (DMA2D)
 - LCD-TFT (LTDC)
 - USBHS1(hs ,fs) 和 USBHS2 (fs)

- 这一领域提供了处理和图形功能
- Cortex-M7 Core
 - 运行在400MHz
 - 16kB I-cache, 16kB D-cache
- AXI 互连：最大频率 200MHz
 - AHB 总线最大频率 200MHz
 - APB 总线最大频率 100MHz
- 图形组件
 - JPEG Codec
 - DMA2D
 - LTDC

- 嵌入式存储器
 - 2x 1MB independant Flash banks with ECC
 - ECC on Flash is applied on each word of 256-bit
 - SRAM with ECC (512kB + 128kB + 64kB)
 - ECC on RAM is applied on each word of 32-bit (4 bytes).
- 外部存储器
 - FMC (SRAM, NAND Flash, SDRAM, ...)
 - QuadSPI
 - SD/MMC

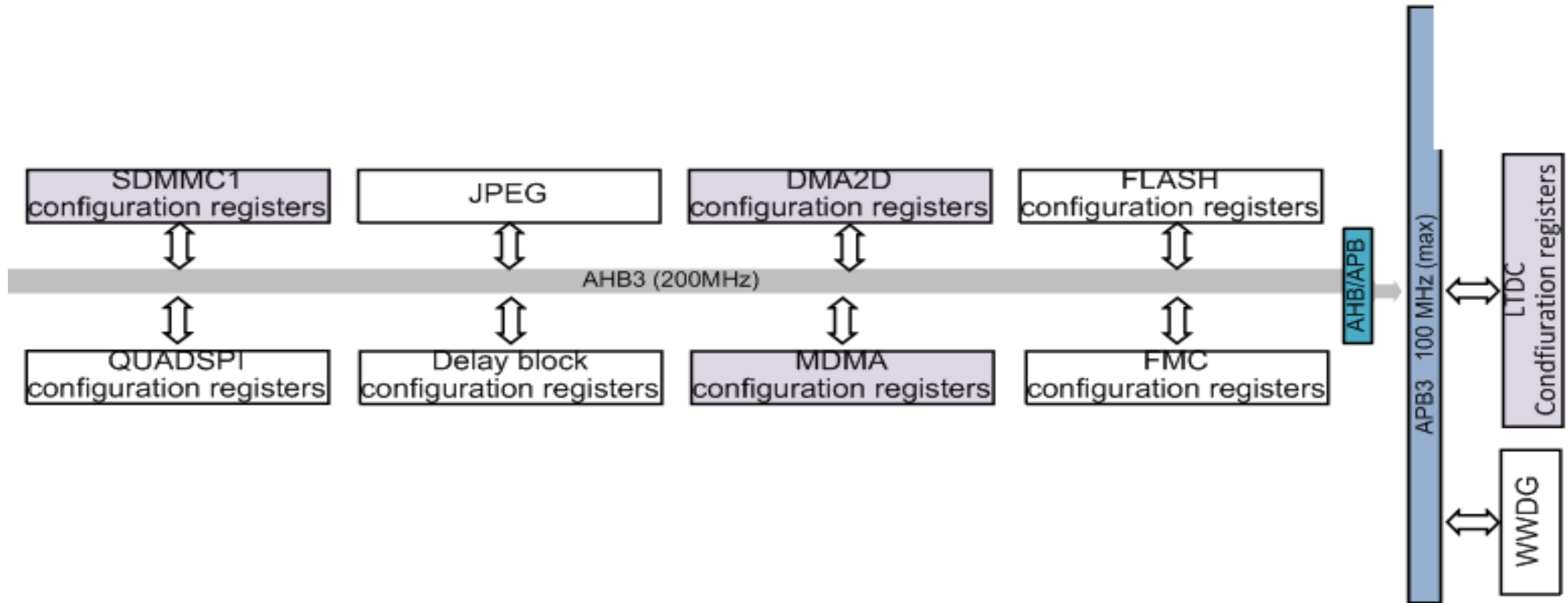
D1 : 系统互连

12



D1 : 总线和外设映射

13

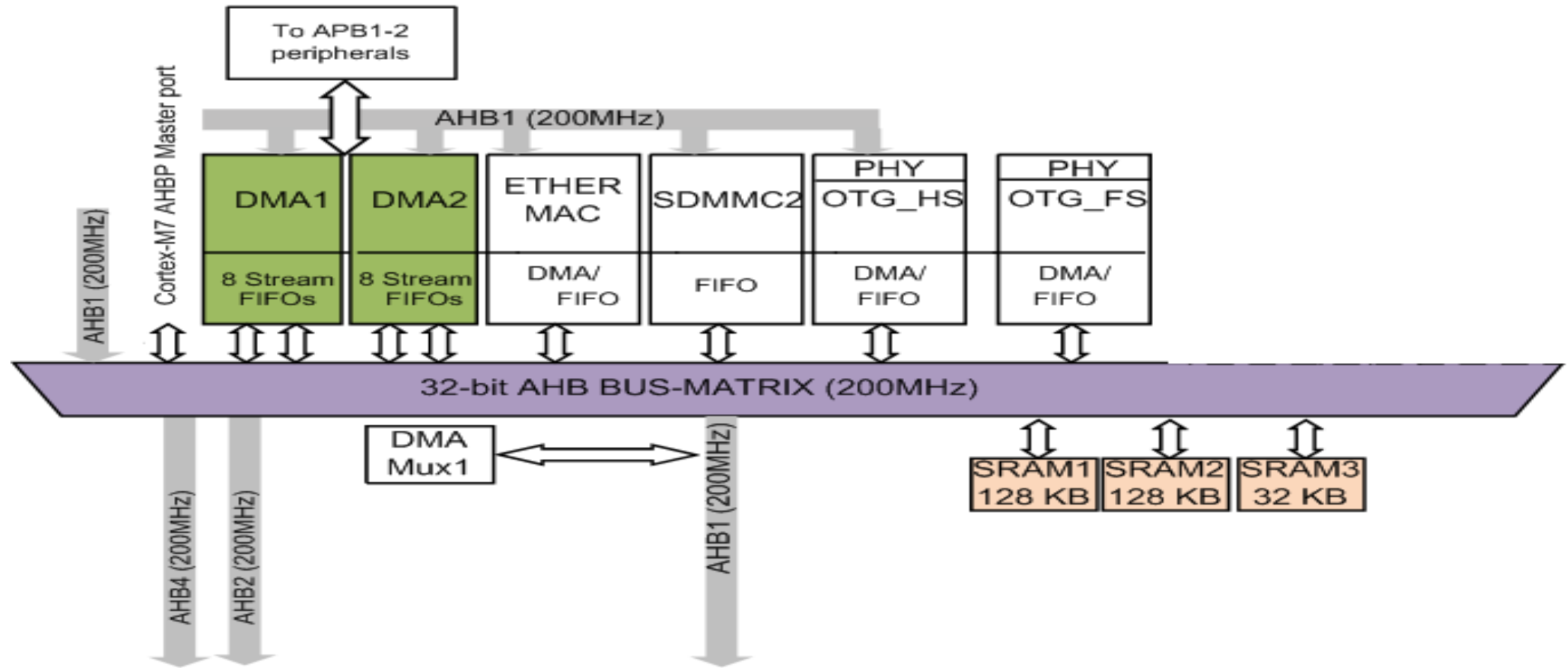


- 这个域提供了外设管理
- AHB 互连: 最大频率 200MHz
 - APB 总线 最大频率 100MHz
- 外设
 - DMA1/2
 - USB, ETH
 - CAN, Timer, SPI, UART, ...

- 嵌入式存储器
 - SRAM with ECC (128kB + 128kB + 32kB)
 - ECC on RAM is applied on each word of 32-bit (4 bytes).
- D2 SRAMs 可以作为外设在这个域的本地 DMAs 数据输入/输出的缓冲区。本地 DMAs 传输结束，数据可以通过MDMA 转移到D1处理域。

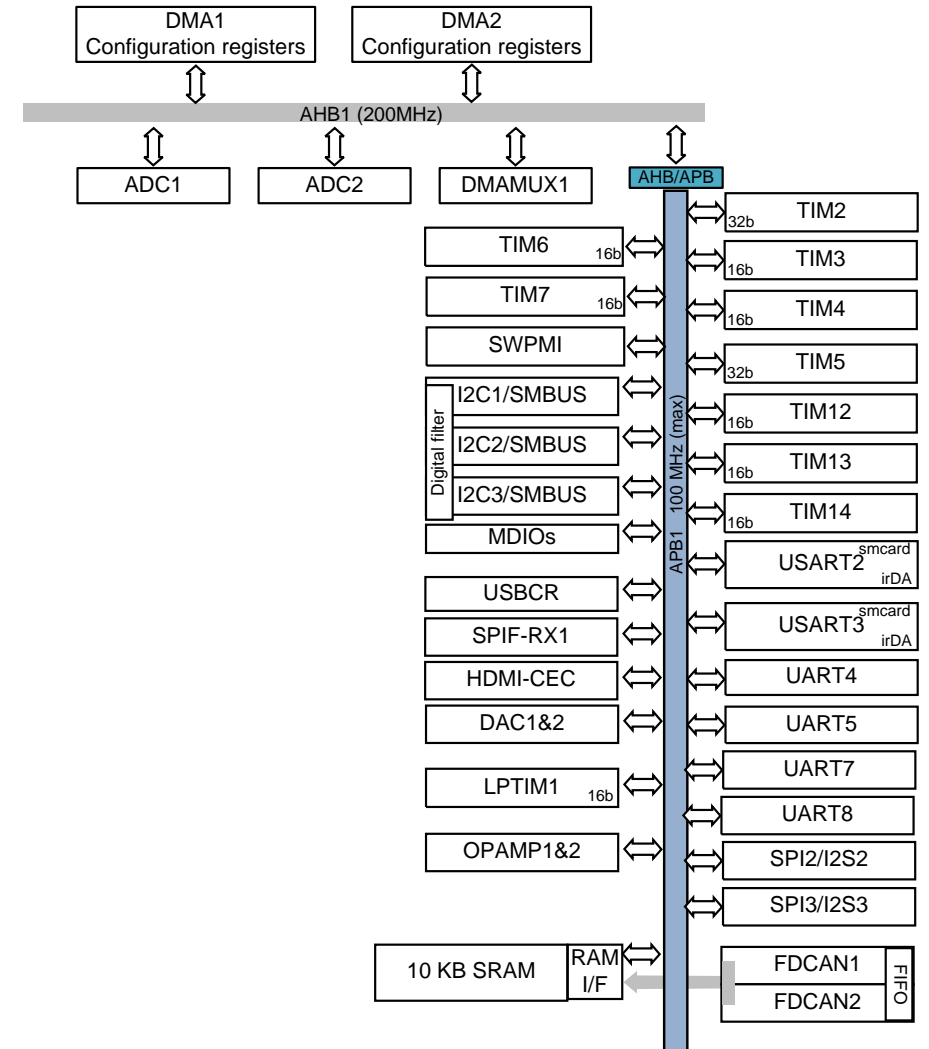
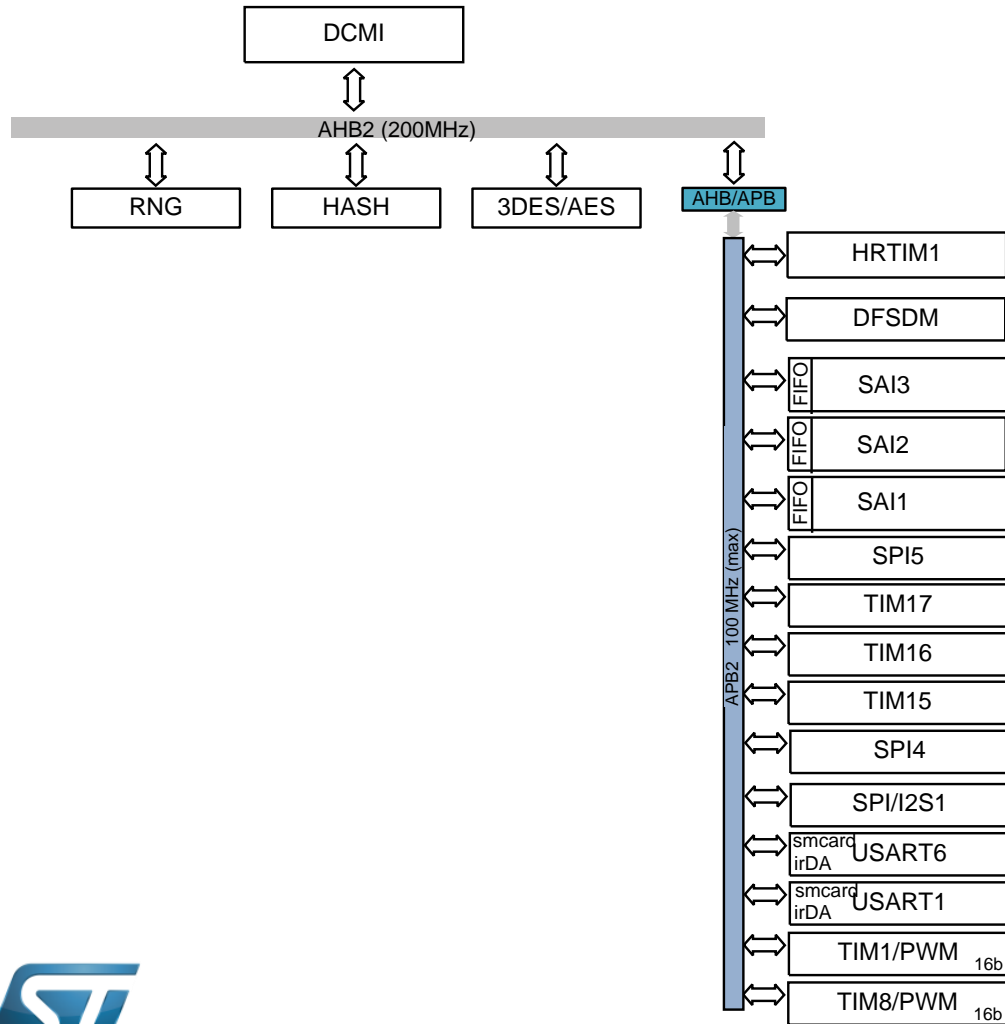
D2 : 互连

16



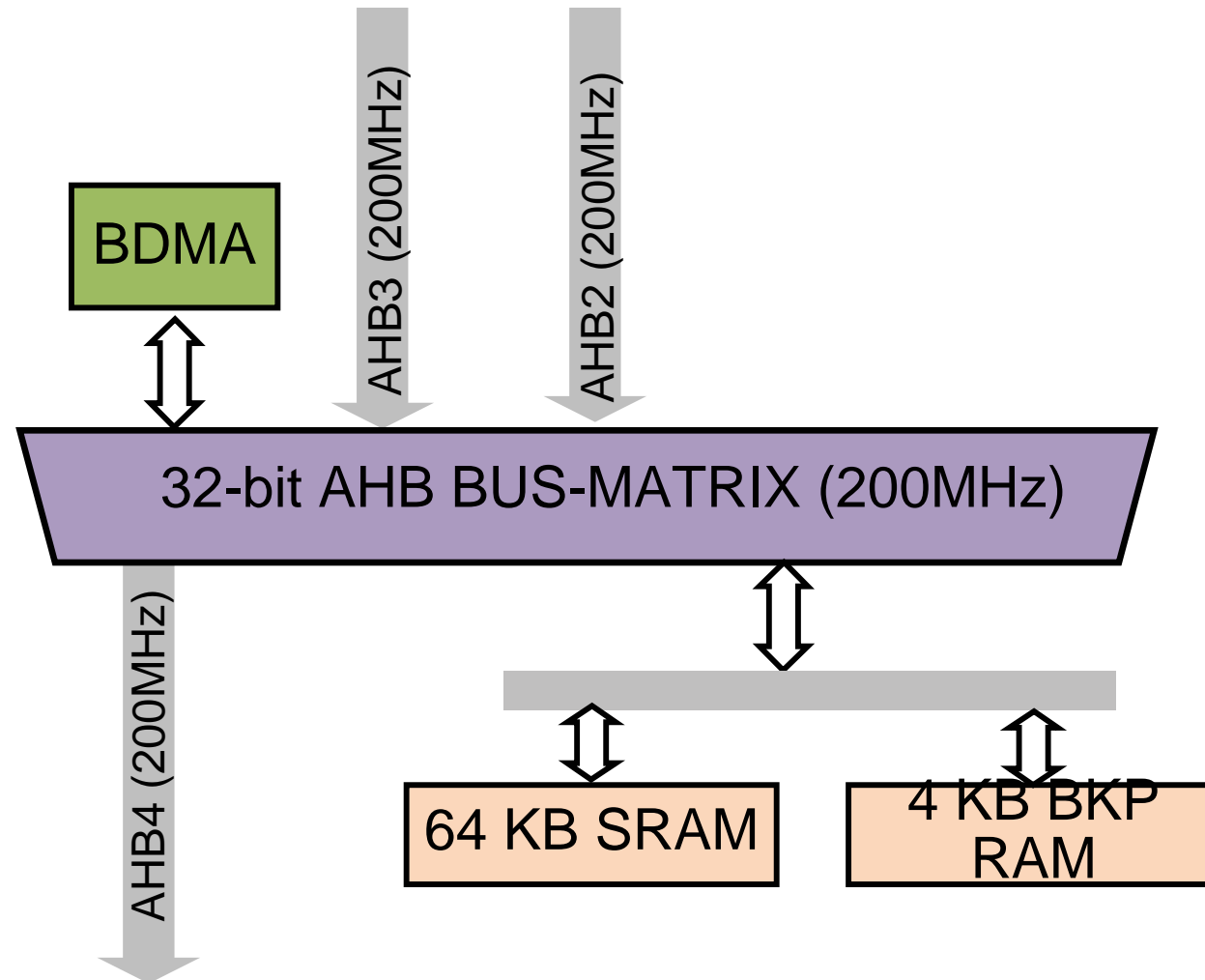
D2: 总线和外设映射

17



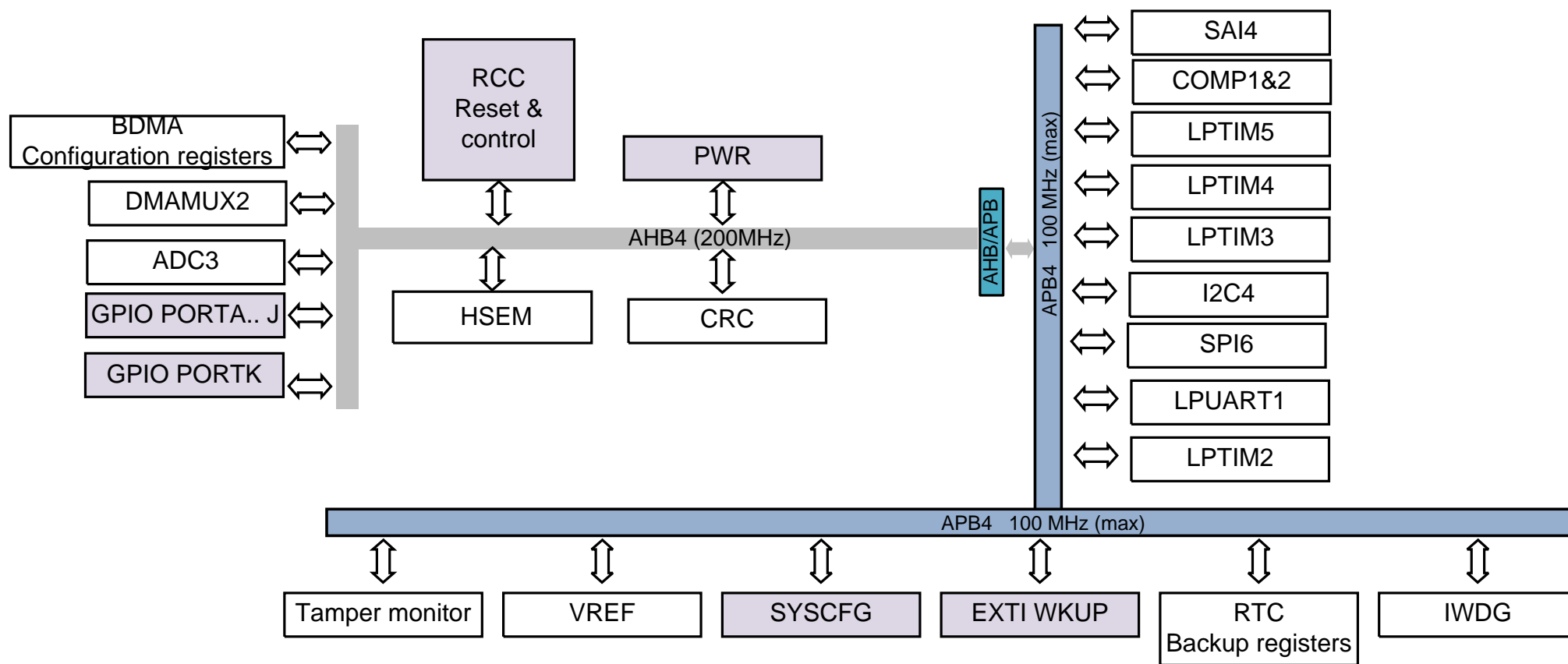
- D3提供系统管理和低功耗操作模式特性
- AHB 互连: 最大频率 200MHz
 - APB 总线 最大频率 100MHz
- 外设
 - BDMA
 - LPTIM, I2C, SPI, LPUART, SAI, ADC, ..
 - RCC, PWR, SYSCFG, HSEM, GPIOs, ..

- 嵌入式存储器
 - SRAM with ECC (64 kB)
- D3 SRAM 可以用来保留数据，当D1和D2域进入DStandby模式



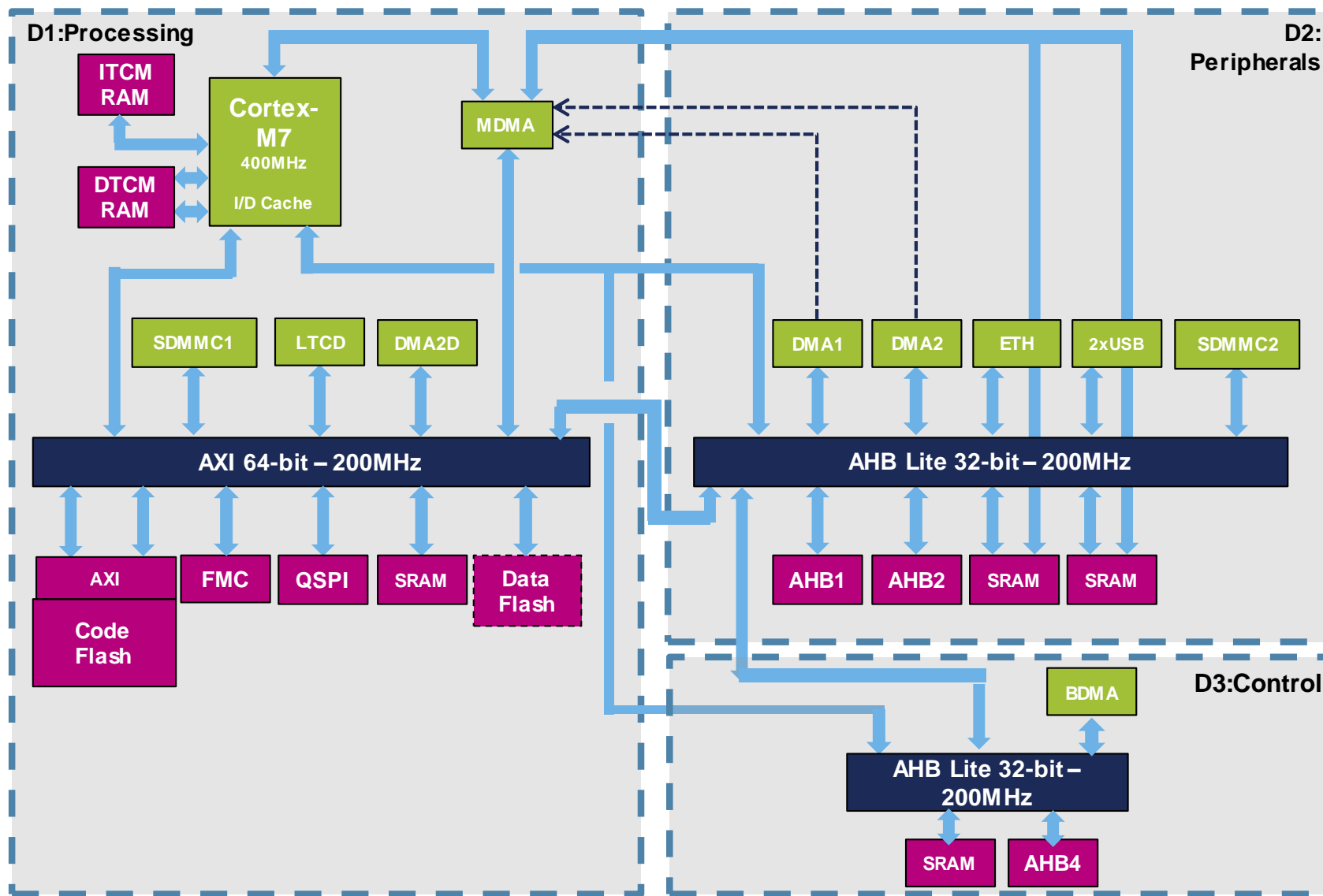
D3 :总线和外设映射

21



STM32H743系统体系结构的简化视图

22



- STM32H7xxx 设备功能:

- 嵌入式SRAM

- Up to 864 Kbytes of System SRAM
 - 128 Kbytes of data TCM RAM
 - 64 Kbytes of instruction TCM RAM
 - 4 Kbytes of backup SRAM

- 闪存

- 闪存接口管理CPU AXI访问闪存。闪存的组织如下所示:

- 两个1 MB的内存块，每个块分为8个扇区.
 - 一个信息块:

- » 当设备在系统内存模式启动时，从系统内存引导。

- » 选项字节配置读写保护、BOR 电平,当设备处于待机或停止模式，看门狗软件/硬件的重置,.

- 嵌入系统SRAM 分为5个块:
 - AXI SRAM (D1 域) 通过 D1 域 AXI 总线矩阵访问:
 - **AXI SRAM** (512 KByte) 映射在地址 0x2400 0000
 - 支持 bytes, half-words, full-words or double-words 访问
 - AHB SRAM (D2 域) 通过 D2 域 AHB 总线矩阵访问 :
 - **AHB SRAM1** (128 KByte) 映射在地址 0x3000 0000
 - **AHB SRAM2** (128 KByte) 映射在地址 0x3002 0000
 - **AHB SRAM3** (32 KByte) 映射在地址 0x3004 0000
 - 支持 bytes, half-words or full-words 访问
 - AHB SRAM (D3 域) 通过 D3域 AHB 总线矩阵访问 :
 - **AHB SRAM4** (128 KByte) 映射在地址 0x3800 0000 , 并且大多系统主控制器可以访问
 - 支持 bytes, half-words or full-words 访问

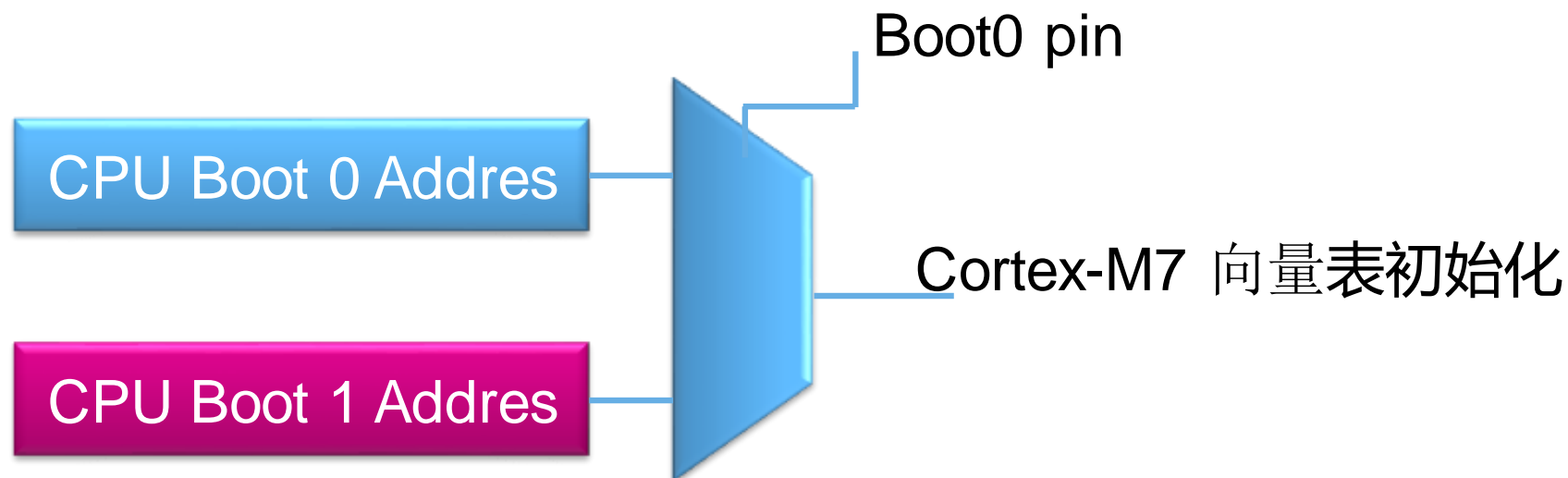
嵌入 SRAM : D2 AHB SRAMs

25

- D2 域 AHB SRAMs 也可以重新映射到：
 - 地址 0x1000 0000 for AHB SRAM1
 - 地址 0x1002 0000 for AHB SRAM2
 - 地址 0x1004 0000 for AHB SRAM3
- 全部系统主控制器都可以通过 D2 域 AHB 总线矩阵来访问 D2 域 AHB SRAMs

- TCM SRAMs 是给 Cortex®-M7 专用的:
 - DTCM-RAM on TCM 接口 映射在地址 0x2000 0000 , 并且只能通过下面两种方式访问:
 - Cortex®-M7,
 - MDMA 通过 Cortex®-M7 CPU AHBS 从总线.
 - ITCM-RAM on TCM 接口 映射在地址 0x0000 0000 , 并且只能通过 Cortex®-M7 访问.
- 这些内存可以运行在最大的CPU 时钟频率 (400MHz) 不需要等待

- 引导地址通过可选字节提供，缺省值如下：
 - 当Boot0=0 从闪存地址 0x0800 0000 引导
 - 当Boot0=1 从系统 bootloader 0x1FF0 0000 引导
- 复位后，在4个系统时间的上升沿锁定引导 pin 的值。



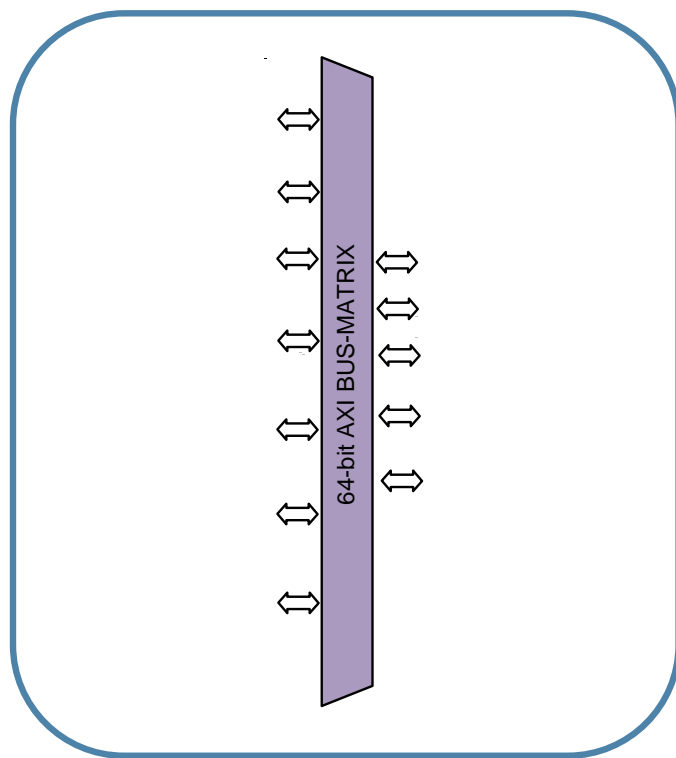
- 如果程序引导内存地址超出内存映射区域，或者是保留区域，缺省的引导地址是：
 - BOOT_ADD0: FLASH at 0x0800 0000
 - BOOT_ADD1: ITCM-RAM at 0x0000 0000
- 当Flash **level 2** 保护允许时：
 - 只有从Flash 引导是有效的。
 - 如果程序引导内存地址超出内存映射区域，或者是 RAM地址，缺省的引导地址是 Flash 0x0800 0000.

STM32H7xx - AXI 互连

AXI互连

版本 1



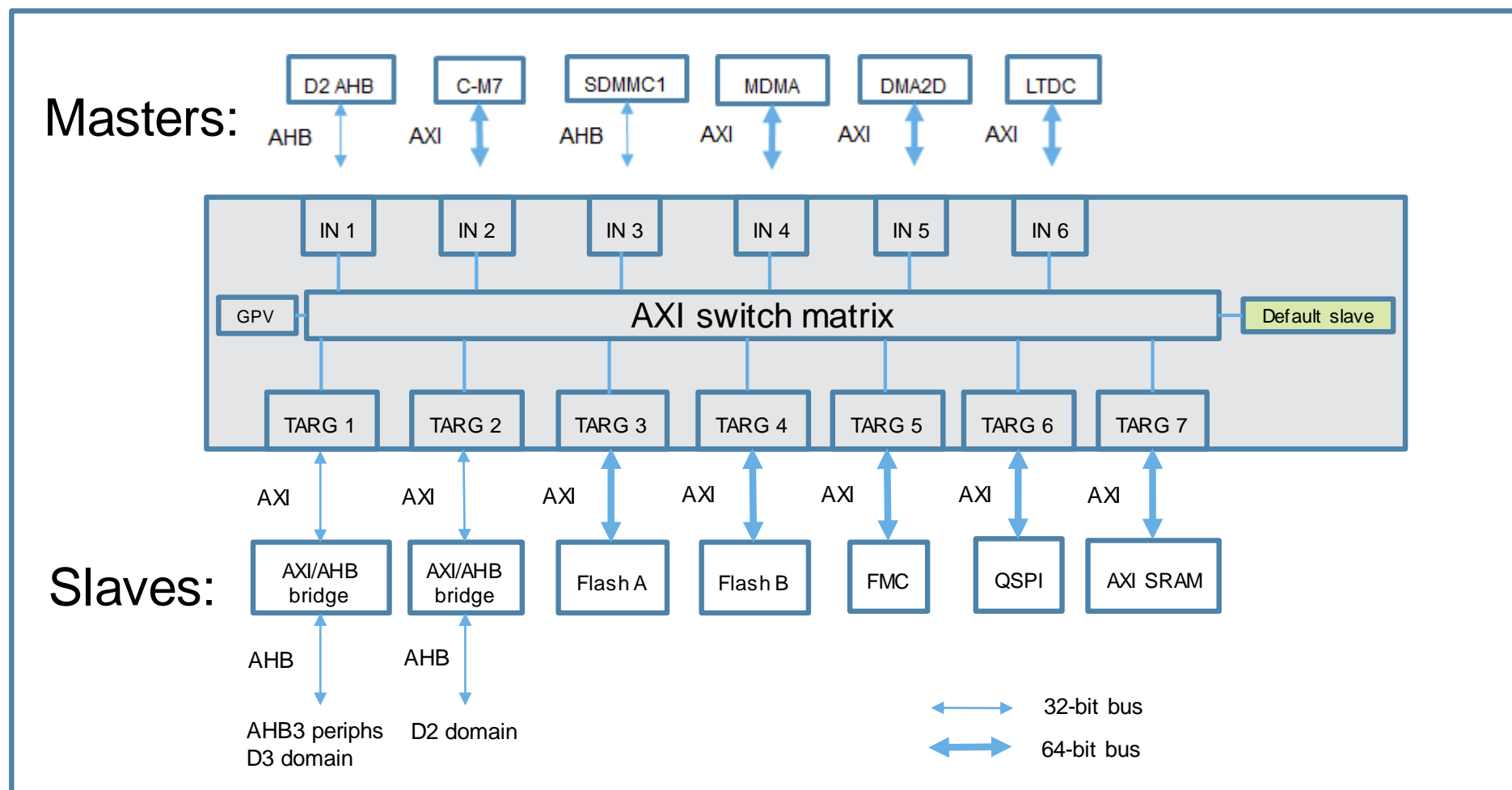


• STM32H7 AXI 互连特色

- 64-bit AXI 总线开关矩阵
- AHB/AXI 桥接功能
- 可编程流量优先级管理
 - QoS 可以通过GPV 进行软件配置

应用好处

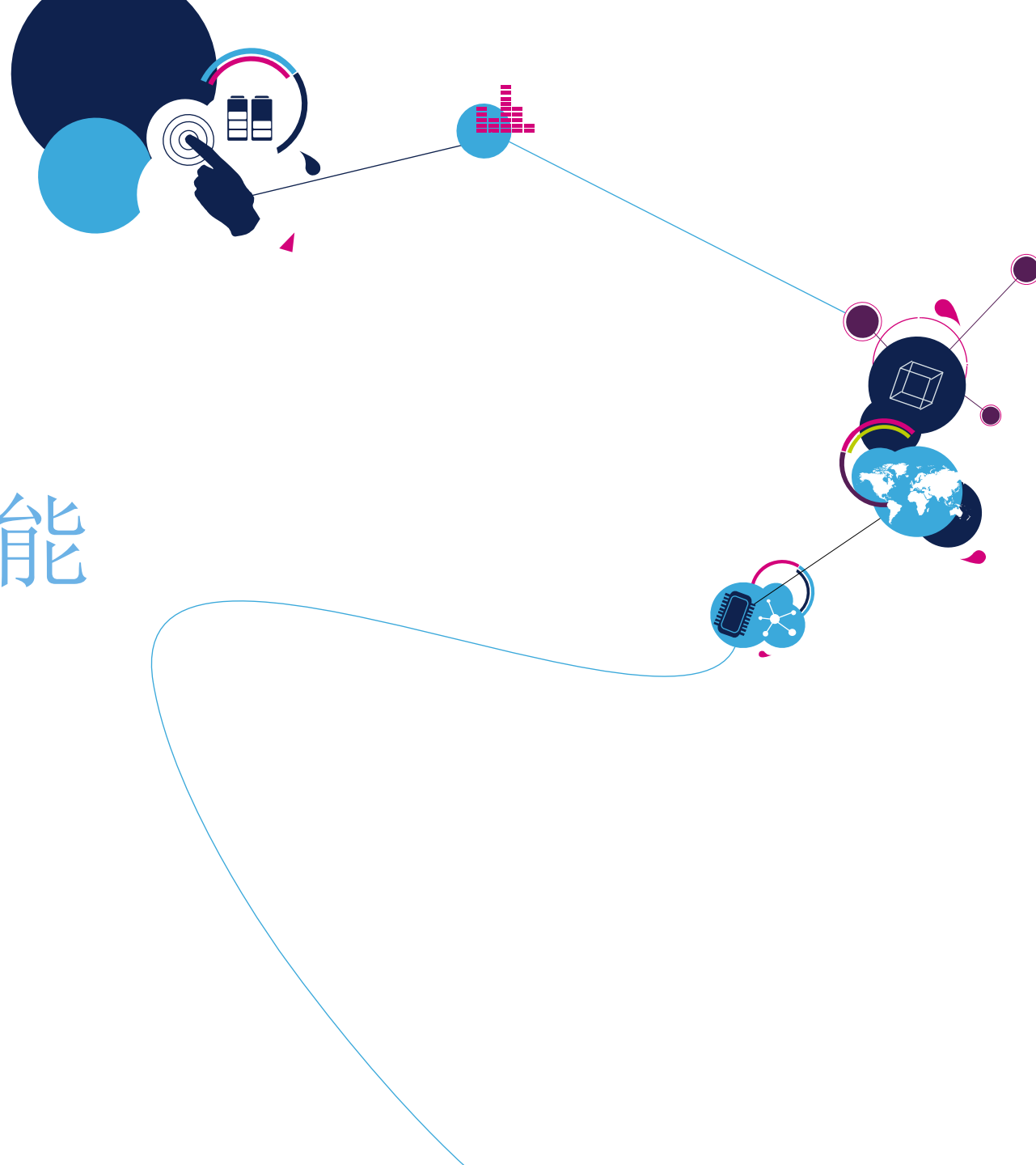
- 优化数据传输带宽
- 配置互连参数
- 提高实时任务的响应能力



- 当两个AXI主同时尝试访问相同的AXI从，基于优先级仲裁
- AXI 读通道和写通道优先级可以独立配置
 - 配置的优先级值从0到15,
 - 值越高,优先级越高。
 - 如果两个重合事务到达相同的AMIB,优先级高的事务比优先级低的事务先通过.
 - 如果两个事务有相同的QoS值,采用最近用过 (LRU)，它的优先级就更高的方案

- QoS是有用的，像图形处理等任务(LTDC, DMA2D)
- 优先级分配会影响整个应用程序，应该仔细研究
 - 将一个 **master** 分配给高优先级，并配置频繁访问相同的**slave**，会屏蔽其他低优先级的 **master**访问这个**slave**
- 任何访问未分配的地址空间，由缺省的 **slave** 处理,并生成返回信号
- 这将确保此类交易完成,不会在 **master** 和 **ASIB**产生阻塞。

STM32H7 性能



- 可以配置每个外围时钟开或关
 - 复位后,除了闪存接口时钟, 所有外围时钟关闭,
 - 在D1Run模式 TCM and AXISRAM 时钟总是开
 - 在D2Run模式 SRAM1/2/3 时钟总是开
- 当程序从AXISRAM, SRAM1/2/3, or SRAM4运行时:
 - Flash 可以放在省电模式
 - Flash 时钟可以关掉
 - 中断向量也必须重新映射到**SRAM!**

模式	说明
CRun	CPU Active → CPU, CPU-子系统总线矩阵, CPU 使能外围时钟活跃.
CSleep	CPU Sleep → CPU 时钟停止, CPU-子系统总线矩阵, CPU sleep 使能外围时钟活跃.
CStop	CPU Deepsleep → CPU, CPU-子系统总线矩阵), CPU外围时钟停止.

- CPU 操作模式由CPU直接控制

- 进入低功耗模式

- CSleep 通过执行 WFI/WFE进入或者 从 ISR 返回. (DEEPSLEEP = 0)
 - Cstop 通过执行 WFI/WFE进入或者 从 ISR 返回. (DEEPSLEEP = 1)

- 退出低功耗模式

- 通过执行 WFI 进入或者 从 ISR 返回
 - 任何有足够的优先级CPU中断
 - 通过执行 WFE 进入
 - CPU 事件
 - SVONPEND = 0 →任何有足够的优先级中断
 - SVONPEND = 1 → 任意中断

模式	说明
DRun	域总线矩阵，时钟活跃
DStop	域总线矩阵，域外围时钟停止
DStandby	域关闭.

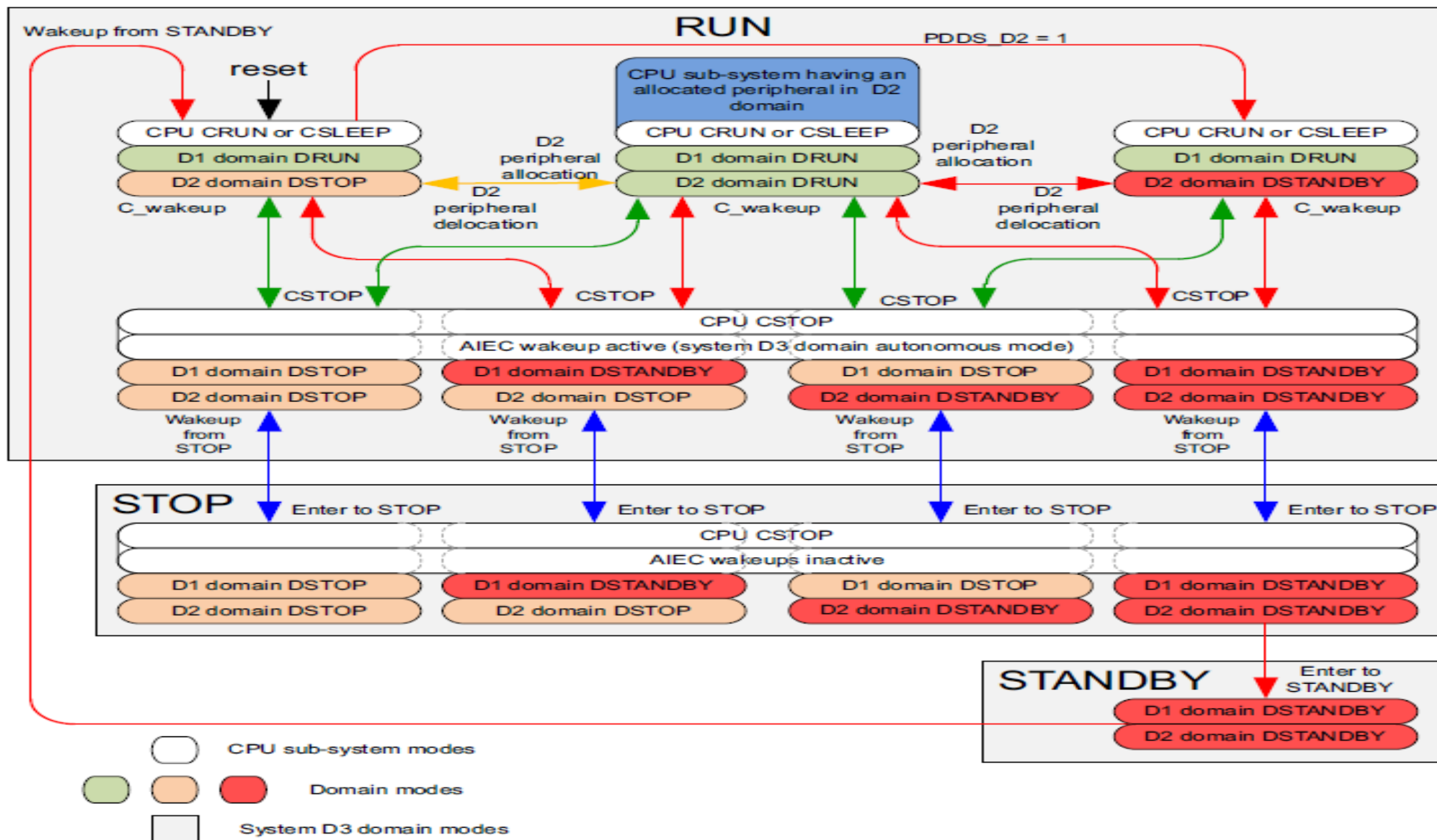
- D2 域操作模式由 CPUs 控制，CPUs 分配这个域的外围设备
 - 域的操作：
 - Drun 如果CPU分配外围设备在这个域，并且CPU是CRun 或 CSleep
 - Dstop如果CPU分配外围设备在这个域，并且CPU是CStop ， PDDS_D2 bit 选择 stop
 - Dstandby 如果CPU分配外围设备在这个域，并且CPU是 Cstop， PDDS_D2 bit选择 standby

模式	说明
Run	系统时钟是活跃，并且转发到系统
Stop	系统时钟停止
Standby	系统关闭。(备份域可能保持活跃)

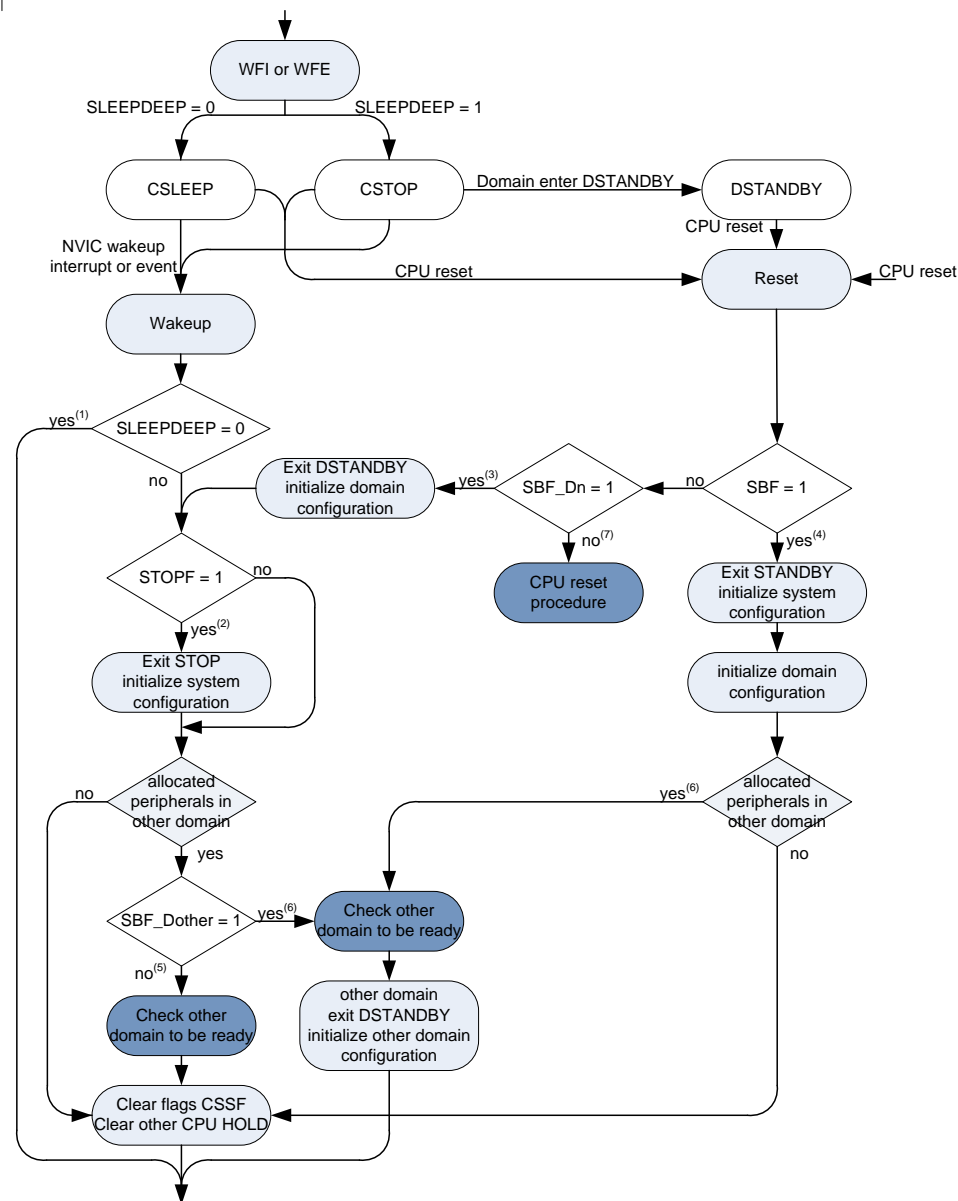
- 系统操作模式由CPU和唤醒源控制
 - Run 当CPU是 CRun 或 CSleep 或一个唤醒源是活跃的.
 - Stop 当CPU是 Cstop, 并且所有的唤醒源被清除, D3 域没有强制在 Run 模式, PDDS_Dx bits 选择 stop
 - Standby 当CPU是 Cstop, 并且所有的唤醒源被清除, D3 域没有强制在 Run 模式, PDDS_Dx bits 选择 standby
- 系统支持 D3 自主 RUN 模式:
 - D3域的外设需要使能自主模式。
 - CPU 强制 D3域留在运行模式通过RUN_D3 寄存器位

- 系统功率状态是由 CPU 和 D3 域控制.
- 状态转换可能由以下发起:
 - CPU 进入低功耗模式.
 - CPU, Domain 和 System 模式
 - 外设分配/解位
 - 域模式
 - 唤醒事件
 - CPU, Domain 和 System 模式
 - 改变 PDDS 设置为 STANDBY
 - 域模式





- Cortex-M SLEEPDEEP 位
 - Sleep
- PWR 控制
 - STOPF
 - 域 SBF_D1 和 SBF_D2
 - System SBF



CPU 唤醒状态依赖域和系统的状态

CPU 模式	域模式	系统模式	S B F_ D n	S B F	S T O P F	CPU 唤醒 RUN	系统 clock
CSleep	DRun	Run	0	0	0	By NVIC interrupt or event.	Unchanged
CStop						DStop	
	DStandby	Run	1	0	0		By EXTI wakeup via reset
		Stop	1	0	1	Default after STOP	
		Standby	0	1	0	By WKUP wakeup via reset	Reset after STANDBY
	any	any	any	0	0	0	By POR via reset

STM32H7

运行模式电流消耗

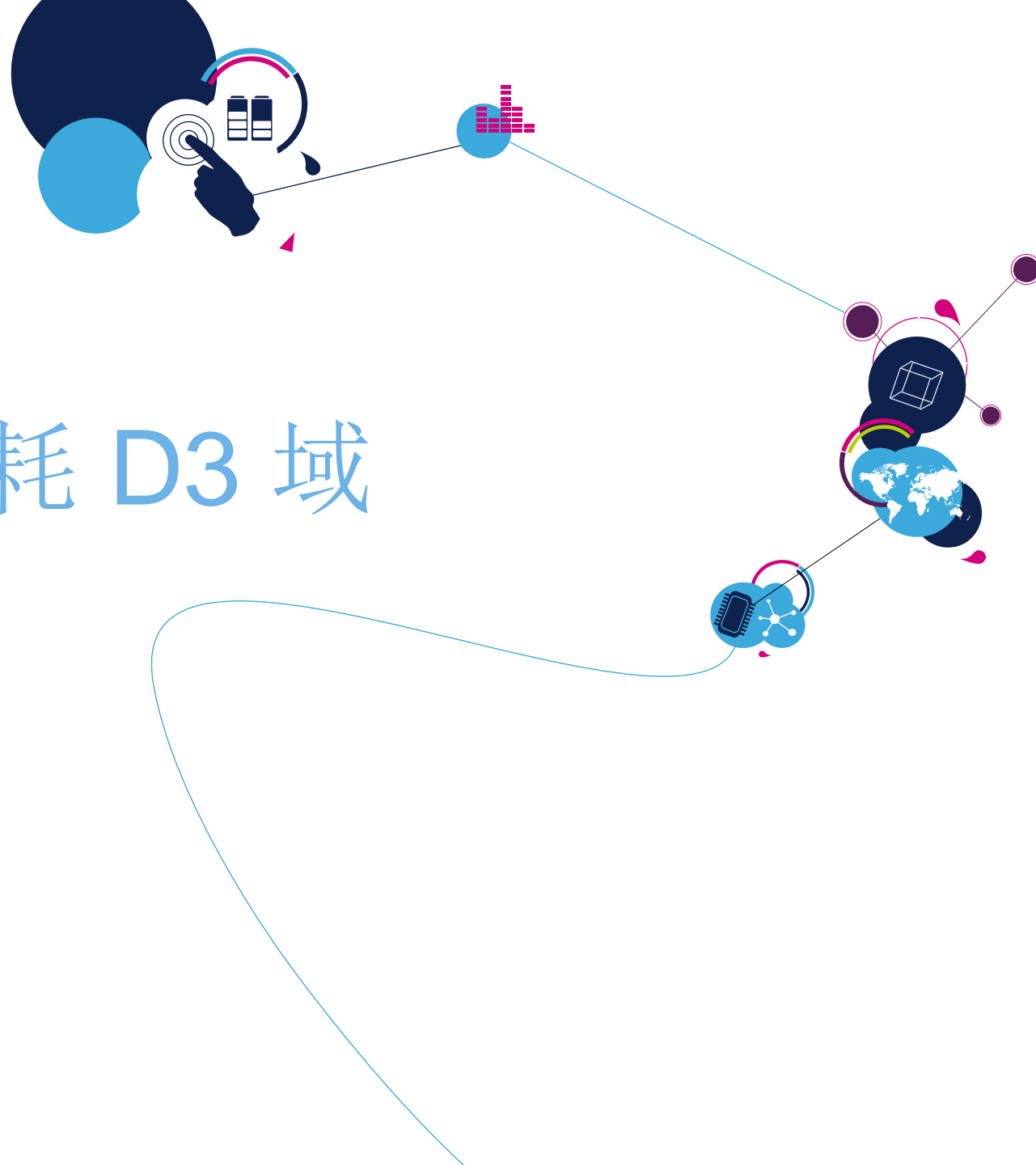
43

Scale	频率(MHz)	电流消耗(mA)
Scale1	400	87,3
Scale2	300	59,9
Scale3	200	37,6

- **Scale1** : 400MHz exec from FLASH bank 1 cache ON_ CM4 OFF All periph disabled
- **Sacle2** : 300MHz exec from FLASH bank 1 cache ON_ CM4 OFF All periph disabled
- **Scale3** : 200MHz exec from FLASH bank 1 cache ON_ CM4 OFF All periph disabled
- Current consumption measurements at 3.3V
- All peripherals are disabled

STM32H7- 低功耗 D3 域

版本 0.1

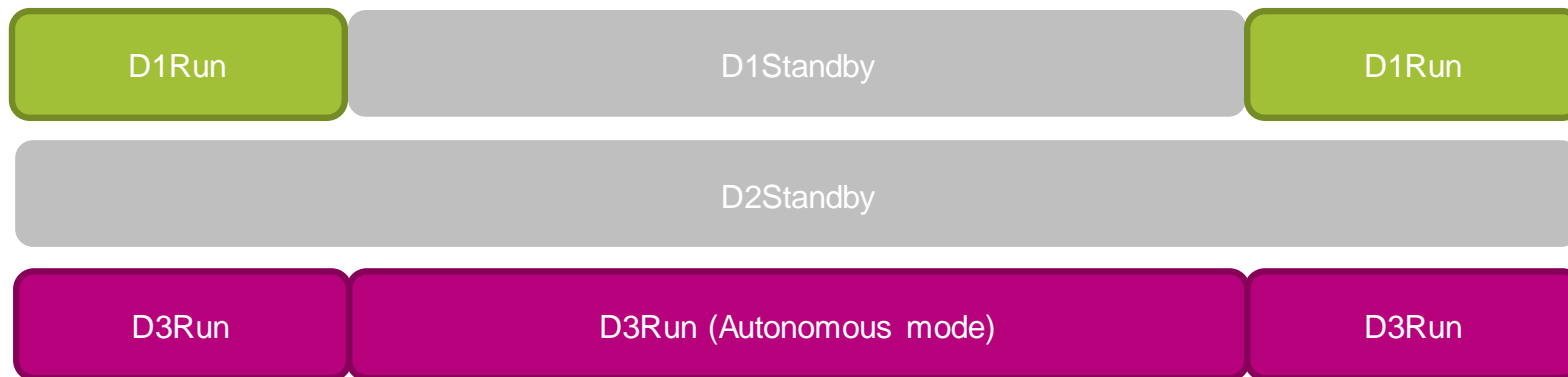


- 本节介绍降低功耗的新的概念 对比 现有的STM32
- 新的特色已经在 RCC, PWR, EXTI 项作了说明.这节提供概述 和 例子。

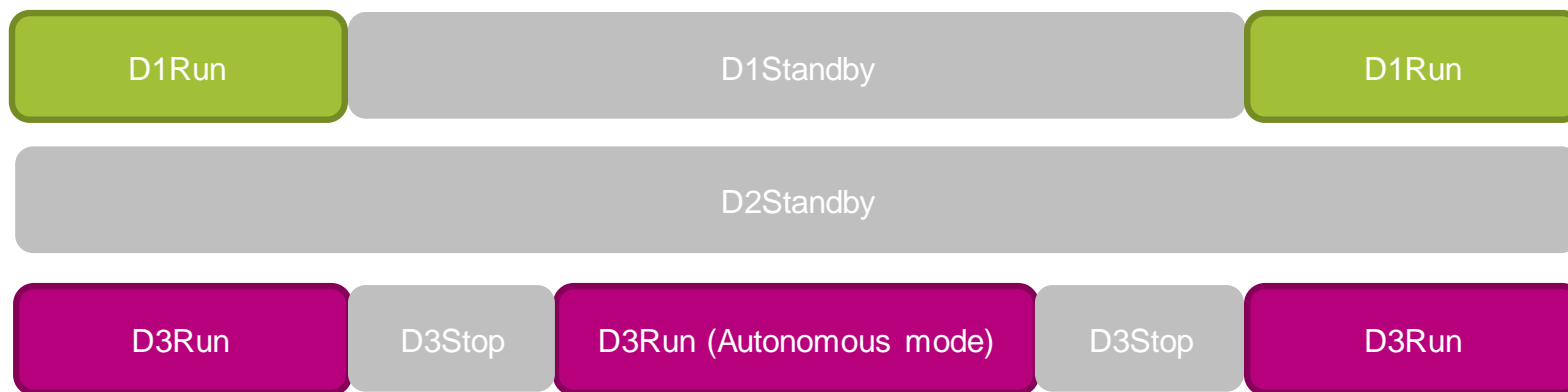
D3 自主Autonomous mode(实例 1)

46

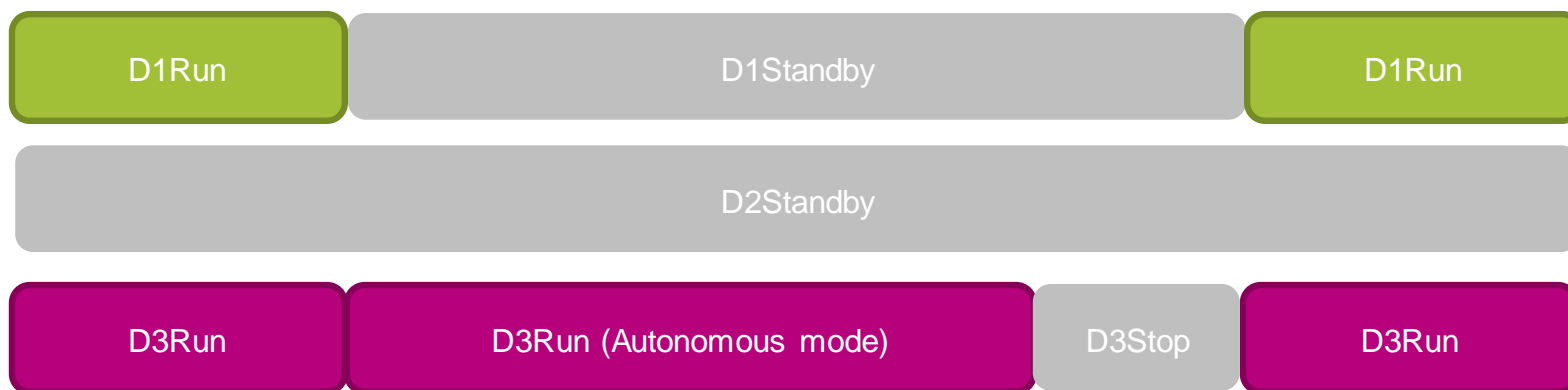
- PWR 的 RUN_D3 位，可以使D3域保持运行，在其他域进入Dstop 或 Dstandby，
- 当CPU进入Cstop（并且D1 Dstop 或 Dstandby），D3 域的外设要保持运行，必须通过软件在RCC上使能自主模式
- D3 域的外设没使能为自主模式，当CPU进入Cstop 时，会停止



- 在自主模式时， D3域可以通外设唤醒事件唤醒
 - GPIO, RTC wakeup, COMPx, I2C4, LPUART1, SPI6, LPTIM2, 3, 4, and 5
- D3域自主模式能重新进入低功耗模式通下列触发
 - BDMA ch6 and ch7, LPTIM4 output, LPTIM5 output
- 唤醒事件设置在EXTI， 并且通过DMA 或 LPTIM 清除（悬挂请求清除在EXTI）



- 这里，在进入Cstop时，唤醒事件能通过软件在EXTI（任何配置事件）激活。
- 唤醒: 和实例2一样



- 中断 Interrupts

- 只有使能了外设总线时钟 (pclk, hclk...), 中断才能由外设产生
- 中断必须清除通过CPU 或 其它的总线主控制器
 - 要么通过 外设寄存器清除对应的事件
 - 或者通过更新FIFO中断级别
- 例如: spi1_it, tim1_brk_it and tim1_upd_it.

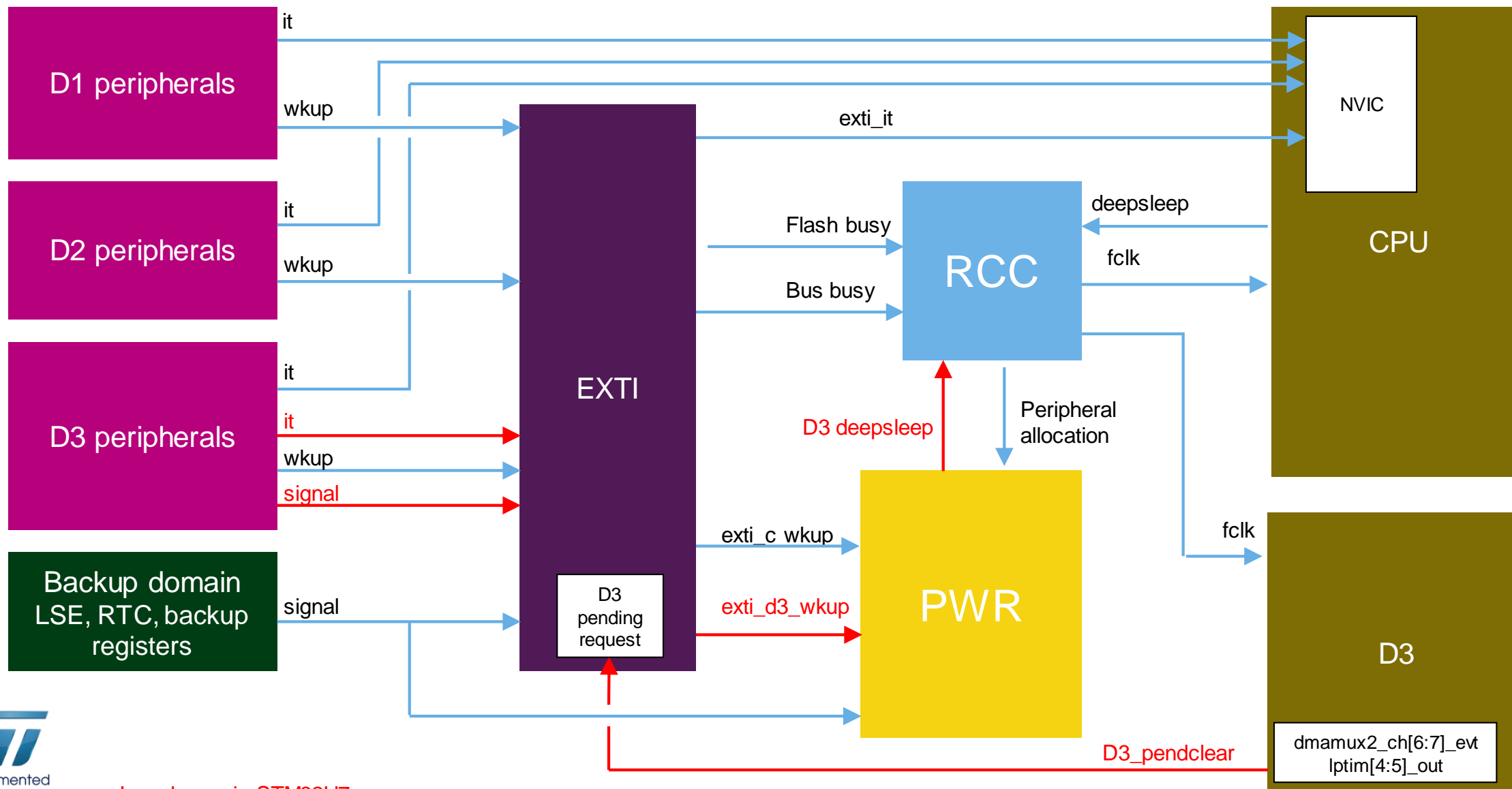
- 唤醒事件 Wakeup events

- 一些外设能产生中断事件，甚至它们的接口时钟不存在。这些中断事件叫做唤醒事件（或异步中断）
- 这些唤醒事件允许 STM32 从停止模式退出
- 例如: i2c1_wkup, usart1_wkup and lptim1_wkup.

- 信号 Signals

- 一些外设能产生脉冲代替中断信号，这些脉冲叫做信号。
- 例如: Examples: lptim2_out and lptim3_out

- 所有中断都连接到 NVIC
 - 例如: spi1_it, dma1_str1_it
- 外设有能力从STOP唤醒系统,或从Cstop唤醒 CPU，唤醒事件通过 EXTI 连接到 NVIC
 - 例如: i2c4_wkup
- STM32H7 新特色: D3域的外设，除了唤醒事件，中断和信号也通过 EXTI 连接到 NVIC ，使 CPU 从 Cstop退出
 - 例如: spi6_it, bdma_ch1_it, lptim2_out



- 程序通过STM32H7 LPUART每隔 M ms 发送 N 个数据, 来达到最小的功耗。
- 如前面«情况3» 所描述的一样

- 许多其他场景可以构建
 - UART reception and CPU is waken up once a defined number of data has been received through UART and stored in SRAM4 or if there is an error
 - UART transmission and CPU is waken up once a defined number of data has been transmitted through UART from the SRAM4
 - I2C slave mode reception in case of adress match. CPU is woken up in case of error or when a defined number of data has been received and stored in SRAM4
 - I2C master message transmission/ reception with or without restart condition
 - SPI slave reception
 - SPI master transmission/reception
 - SAI transmission
 - ADC conversion at regular time interval ...

系统状态	D1状态	D2状态	D3状态
RUN	DRUN/DSTOP/DSTANDBY	DRUN/DSTOP/DSTANDBY	DRUN
STOP	DSTOP/DSTANDBY	DSTOP/DSTANDBY	DSTOP
STANDBY	DSTANDBY	DSTANDBY	DSTANDBY

- 当一个域的总线矩阵时钟在运行，这个域处在 **DRUN** 状态。如果CPU 在 **CRUN / CSLEEP** 状态，并且分配了该域的外设，这个域的总线矩阵时钟要运行。
- 当一个域的总线矩阵时钟不再运行，这个域处在 **DSTOP**状态。这个域的 CPU 处在**CSTOP**状态，其它的CPU也没有分配外设在这个域，或者 CPU 处在 **CSTOP**状态。
- 当VCORE关掉，域 处在 **DSTANDBY** 状态。

系统重新开始：

- HSI 作为系统和外设的时钟。
- 所有的 PLLs 关闭。

系统从**STOP**重新开始：

- 应用可以选择 HSI 或 CSI 做为系统时钟。
- 应用可以选择 HSI 或 CSI 做为外设时钟。

从 **DSTOP** 或 **CSTOP** 重新开始:

- 时钟设置使用，进入**DSTOP/STOP** 之前的设置。

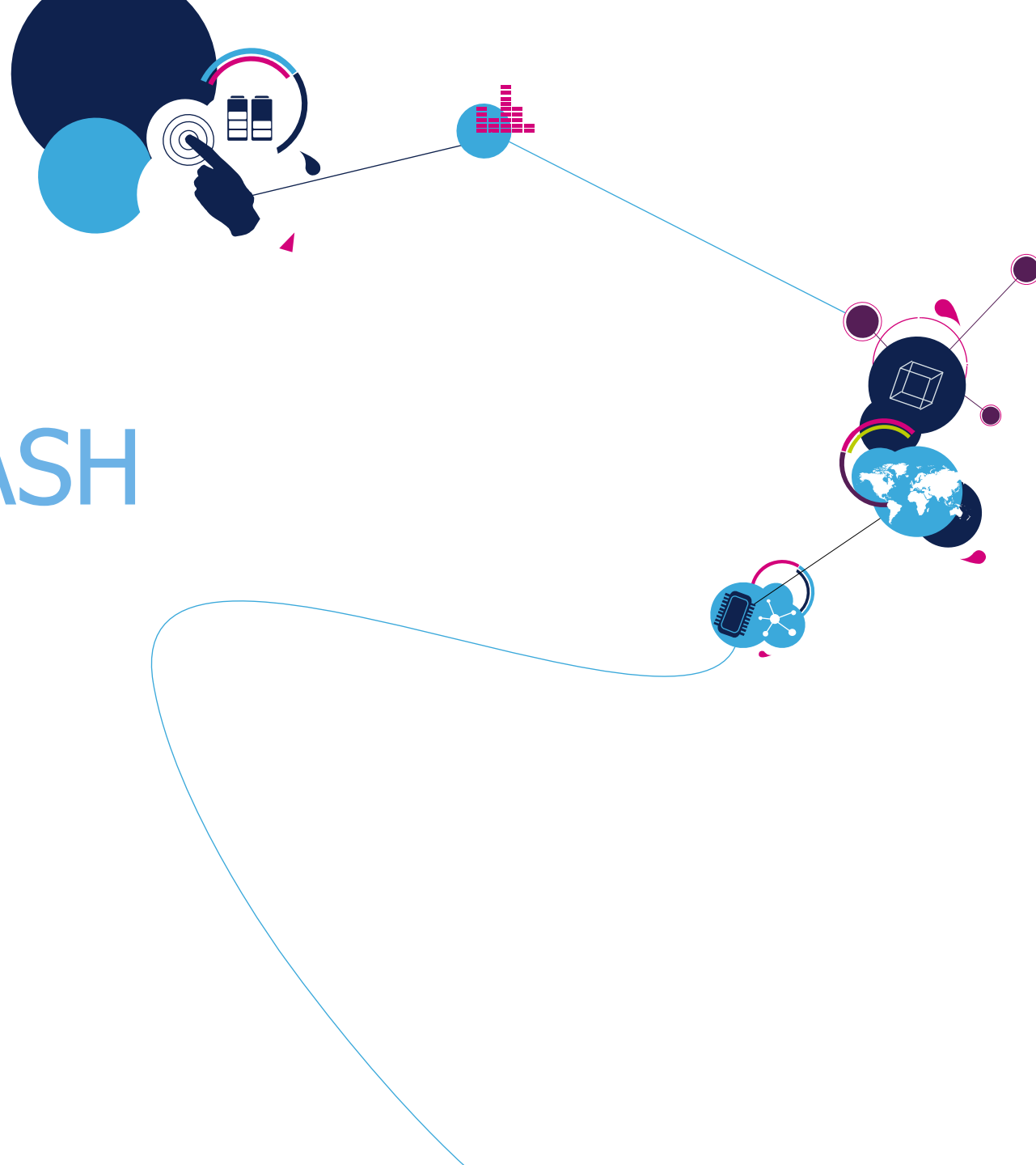
在 D3 域的外设可以工作在自治模式下

- 在自治模式下，位于D3域的外设，甚至 D1和D2域 处在 DSTOP 或 DSTANDBY状态下，还可以接收外设的时钟，。
- 自治模式下外设的时钟的门控，取决于D3域的模式。
- 同时需要注意 D3 域，它自己可以依靠外设的活动，从DRUN切换到 DSTOP，从 DSTOP切换到 DRUN 根据周围的活动。

STM32H7- FLASH

嵌入式 FLASH

版本 0.1

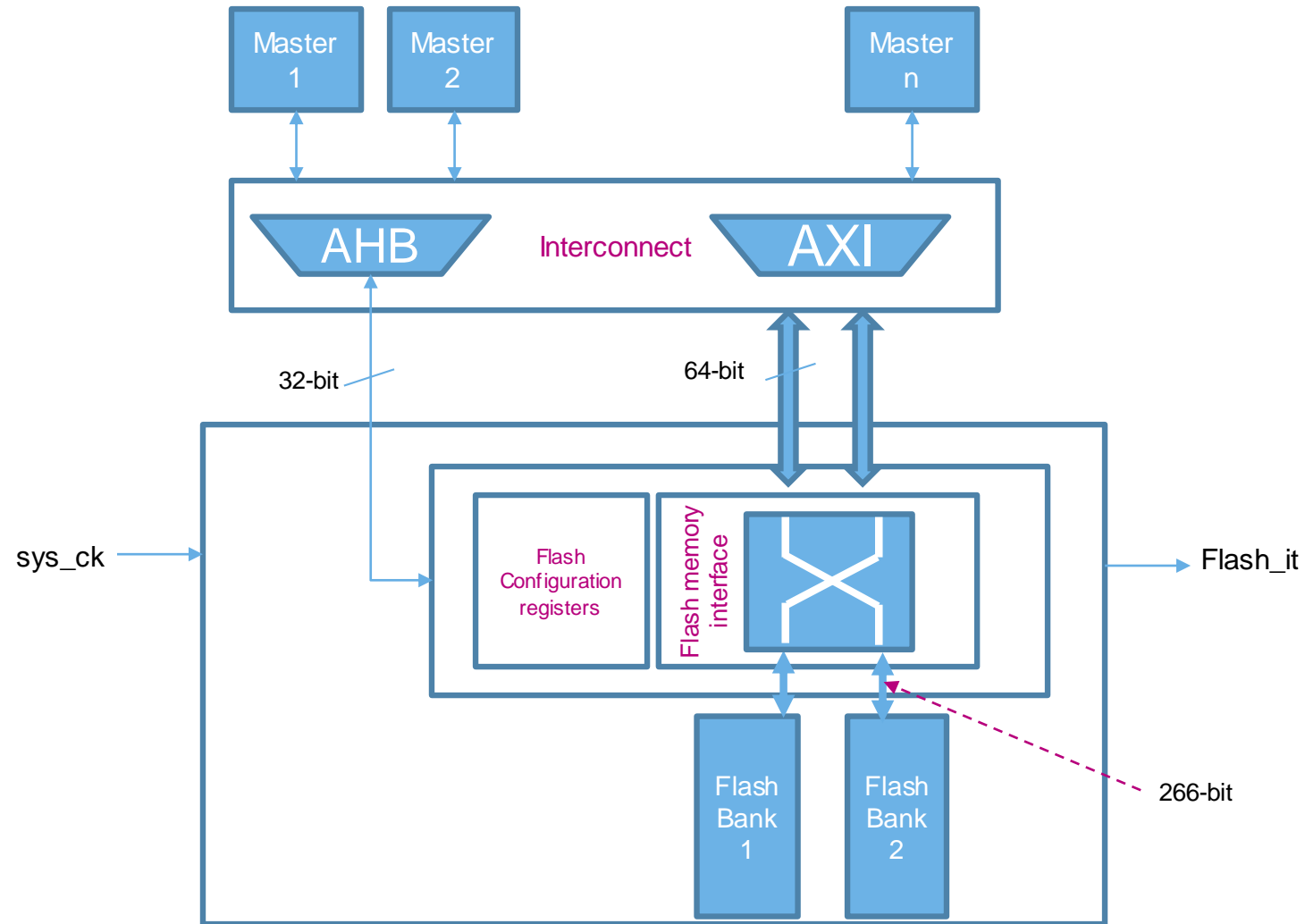


- 多达 2 Mbytes Flash
- 256 位宽的 Flash 字
- 错码校验Error Code correction (ECC) : 10 ECC 位 每个 Flash 字
- 编程 Program:
 - Flash 编程由 256 位
 - Double-word, word, half-word and byte 写操作
- 双库组织支持同时操作
 - 能在两个库并行进行两个 read/program/erase 操作

- 增强的保护特色
 - 读保护
 - 扇区写保护
 - 2 个 PCROP 保护区 (每个库 1 个) (只执行内存)
- 2 安全区在用户 Flash 内存(每个库 1 个)
- 库交换: 每个Flash 库的 用户内存的地址映射可交换
- 可选字节

Flash 内存块图

60



- Flash存储器组织 266 位作为 Flash字:
 - 256-位 数据 Flash 字 (8 words or 32 bytes)
 - 和 10 ECC 位
- Flash 存储器分为两个独立的库，每个库有：
 - 1 Mbyte 用户 Flash存储器块分为 128 Kbytes 扇区(8x扇区)
 - 系统 Flash存储器 128 Kbytes
 - 2 Kbytes (64 Flash words) 用户可选字节给用户配置用
 - 只在 bank 1 有效
 - 只能通过 Flash register 接口访问 (它不是存储器映射)
- 在同一时间在每个Flash库， Flash接口能驱动不同的操作

Flash 存储器组织

62

Block		Name	Block base address	Size
Bank 1	Main Memory	Sector 0	0x0800 0000 - 0x0801 FFFF	128 Kbyte
		Sector1	0x0802 0000 - 0x0803 FFFF	128 Kbyte
		
		Sector7	0x080E 0000 - 0x080F FFFF	128 Kbyte
	System Flash	System memory	0x1FF0 0000-0x1FF1 FFFF	128 Kbyte
	Option byte Sector		Not memory mapped	2 Kbyte
Bank 2	Main Memory	Sector 0	0x0810 0000 - 0x0811 FFFF	128 Kbyte
		Sector1	0x0812 0000 - 0x0813 FFFF	128 Kbyte
		
		Sector7	0x081E 0000 - 0x081F FFFF	128 Kbyte
	System Flash	System memory	0x1FF4 0000-0x1FF5 FFFF	128 Kbyte

- Flash 接口支持下列访问类型：
 - Double-word (64 bits)
 - Single-word (32 bits)
 - Half-word (16 bits)
 - Byte (8 bits)
- Flash 接口实现：
 - 一个双 AXI 总线接口给 code/data 访问
 - 一个 AHB 接口给 Flash 接口配置
- 根据闪存接口的频率，为了确保正确的闪存接口的读操作，等待状态的次数（时间）必须正确配置在 **FLASH_ACR register**

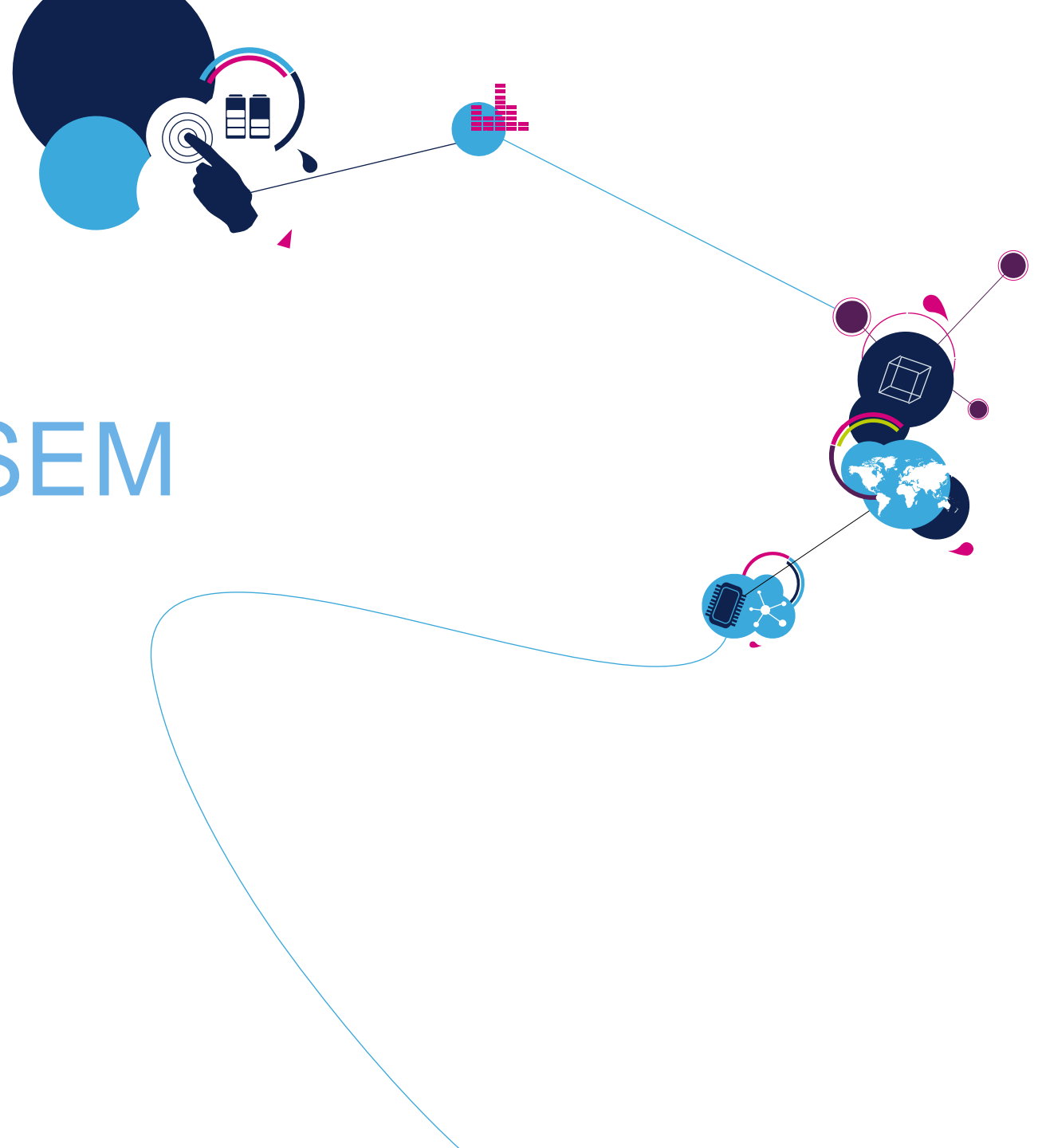
- 上电后，时钟是采用**HSI（64 MHz）**和**7ws**默认的配置在寄存器**FLASH_ACR**
- AXI**时钟频率和闪存读取时间的关系

V_{CORE}		Wait States(WS) (LATENCY)	Maximum AXI Frequency
VOS 1	1.15V – 1.26V	0WS	70 MHz
		1WS	140 MHz
		2WS	210 MHz
VOS2	1.05V – 1.15V	0WS	55 MHz
		1WS	110 MHz
		2WS	165 MHz
		3WS	220 MHz
VOS3	0.95V – 1.05V	0WS	45 MHz
		1WS	90 MHz
		2WS	135 MHz
		3WS	180 MHz
		4WS	225 MHz

- Flash接口有 read command buffer
 - Buffer 深度固定为 3 请求
 - 当它满了 (3 读请求队列在 buffer), 任何新的请求将中断总线接口和终于主控制
- Flash接口有一个 read data buffer
 - 任何系统读取请求, 属于相同的闪存数据字 (256位) 的数据, 数据直接读取当前数据读取缓冲区, 不触发额外的闪存读取操作
 - 读取缓冲区中没有数据的, 系统读取请求将触发闪存读取操作.
- 读取操作按Flash接口接收到的顺序执行.
 - 当当前读取事务的最后一个数据从闪速存储器转移到闪存接口内的读取数据缓冲器时, 读取缓冲区条目将释放。

STM32H7 - HSEM

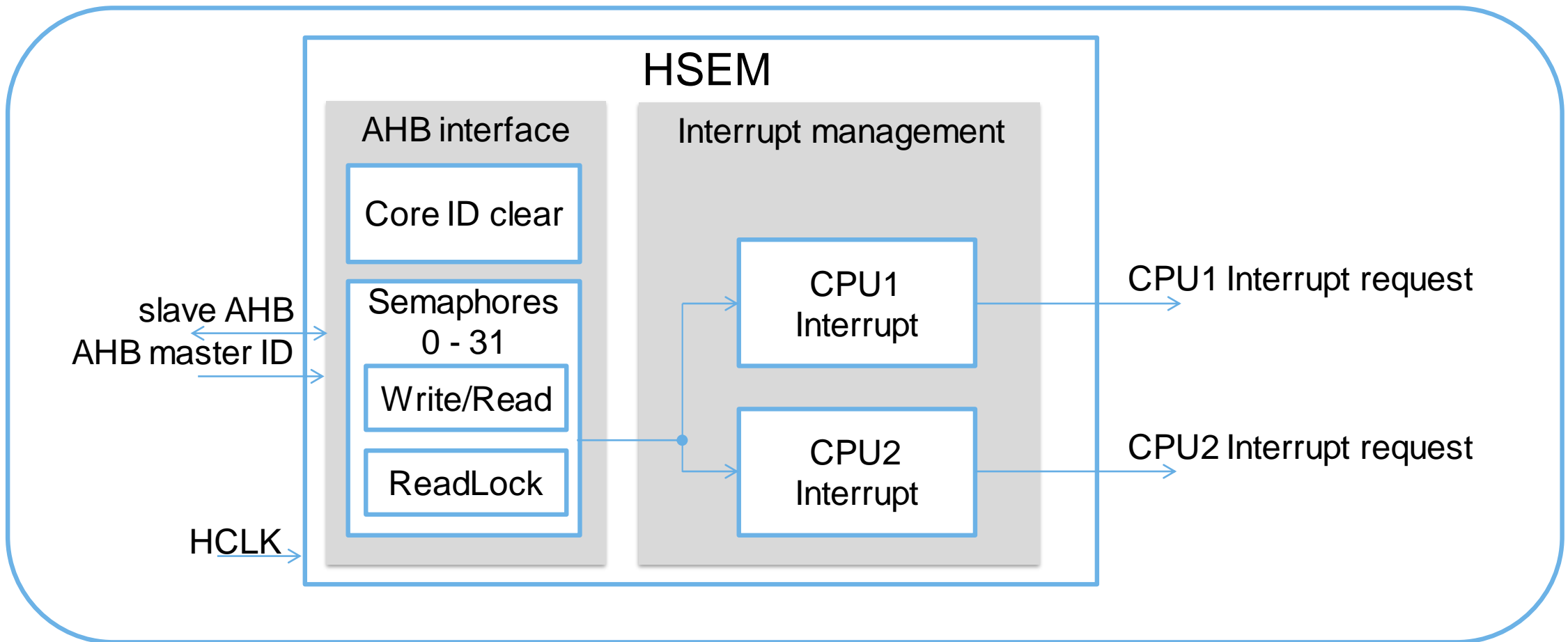
硬件信号量



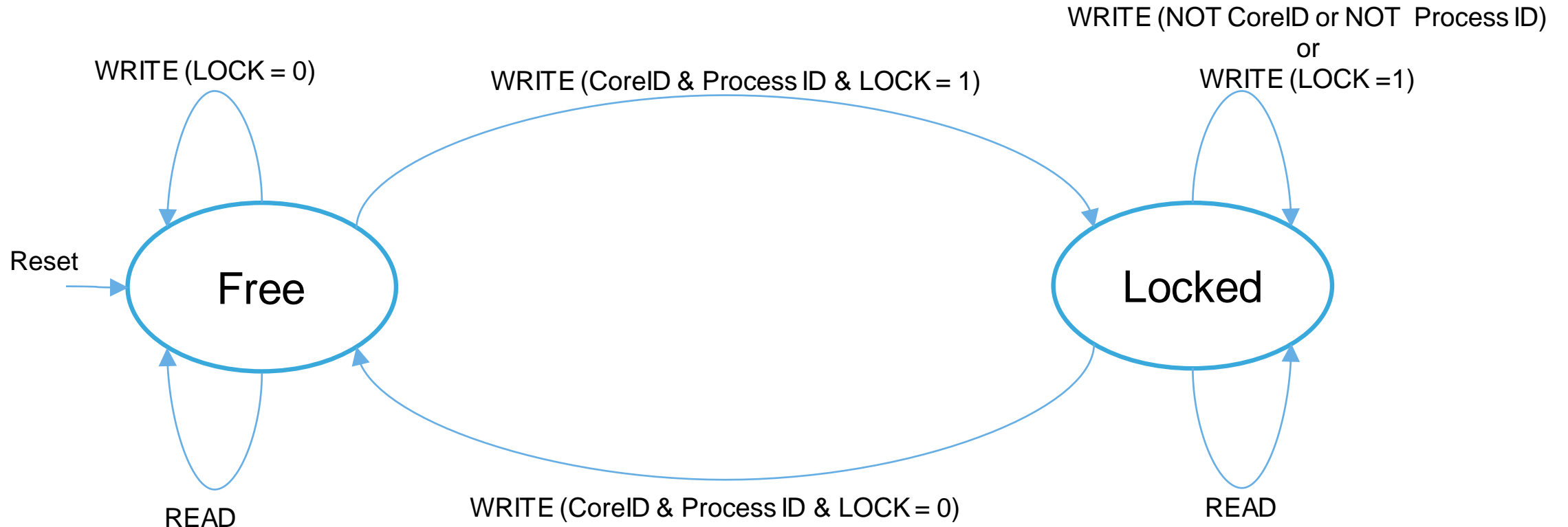
- 管理访问权限和同步
 - 不同的进程运行在相同的CPU上
 - 不同的CPU
- 32信号量
- 两种锁机制
 - 2-step write, read back lock
 - 1-step read lock
- 信号量释放中断生成

应用好处

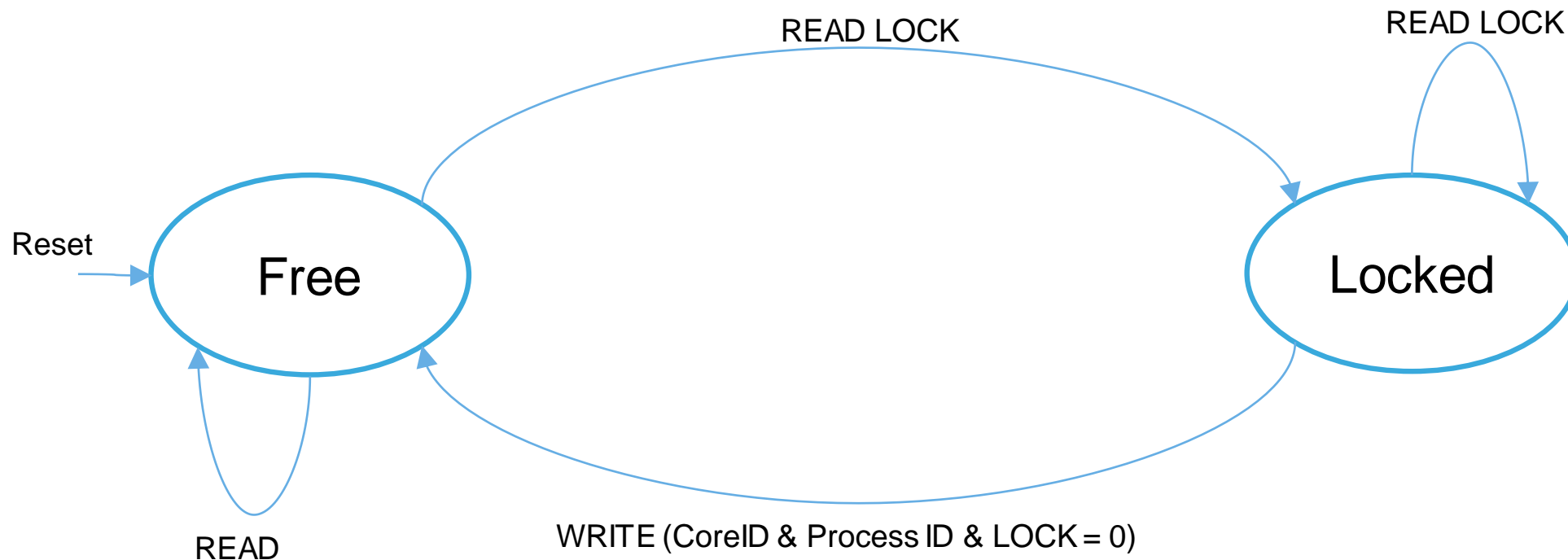
- 防止共享资源访问冲突
- 确保在进程之间进行同步
- 没有阻塞信号处理



2-step write lock



1-step read lock



一般 Core ID 清除

- 所有通过CPU锁的信号量，可以通过 Semaphore clear register 清除
 - 当 CPU 不再正常时，允许清除锁定信号量
- 写 Core ID 和 Key value.

中断事件	说明
INT0_SEM	Semaphore getting free interrupt to CPU1
INT1_SEM	Semaphore getting free interrupt to CPU2

模式	说明
Run	Active. Peripheral interrupts cause a CPU to exit CSleep and CStop modes.
Sleep	Frozen.
(D)Stop	Frozen.
(D)Standby	Powered-down. The peripheral must be reinitialized after exiting domain and system Standby mode.

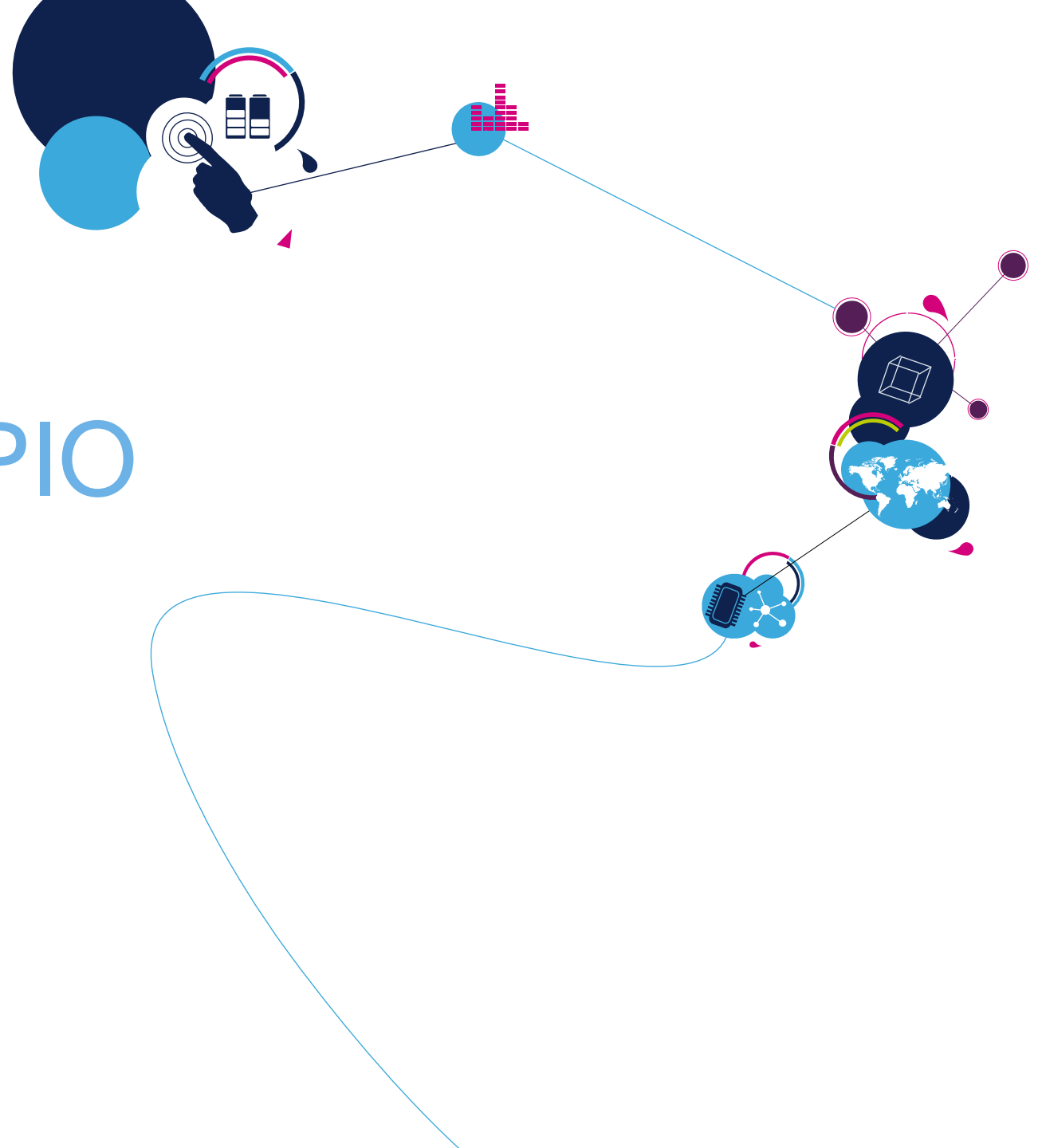
- 管理共享资源.
 - RCC, PWR, AES, RNG,
 - Lock semaphore before peripheral access.
 - Free semaphore once done.
- 同步处理
 - Process A lock semaphore when waiting for input, and got to sleep.
 - Process B when input available locks and frees an other semaphore, which will wakeup process A.
 - Processor A free it's semaphore.

- 这是一个列表**HSEM**相关外设控制器。如果需要更多的信息，请参考这些外设培训
 - Reset and clock control (RCC)
 - Interrupts (NVIC)
 - Asynchronous Event and Interrupt Control (EXTI)

STM32H7 - GPIO

通用输入/输出接口

版本 1



- 双向操作高达 **168 * I / O** 引脚
 - 共享访问 **11 GPIOx** 端口命名从 **GPIOA** 到 **GPIOK**, 每个端口最多有 **16 I/O** 脚
 - 所有的引脚都有外部中断和唤醒的能力
 - 原子比特置位和复位操作, 使用 **BSRR** 和 **BRR** 寄存器
 - 独立配置每个 I/O 引脚
 - 复位后 I / O 引脚 处于模拟模式
- **GPIO** 寄存器接口直接连接到 **D3 AHB4** 总线
- 当 **VDD** 高于 **1.8 V**, 大多数的 I / O 引脚 能承受 **5 V** 的电压。

- No compromises on communication interfaces at low-voltage
 - 高速配置位在 SYSCFG_CCCSR
- 当激活 HSLV bit 。当电源电压低于2.5 V，它可以提升端口的性能。
- 只有当产品供应电压低于2.5 V时，必须使用
- 当VDD高于2.5 V，设置此位可能是破坏性的
 - 只有 IO_HSLV 用户选项位设置了，这位才能激活

直接路径到 ADC

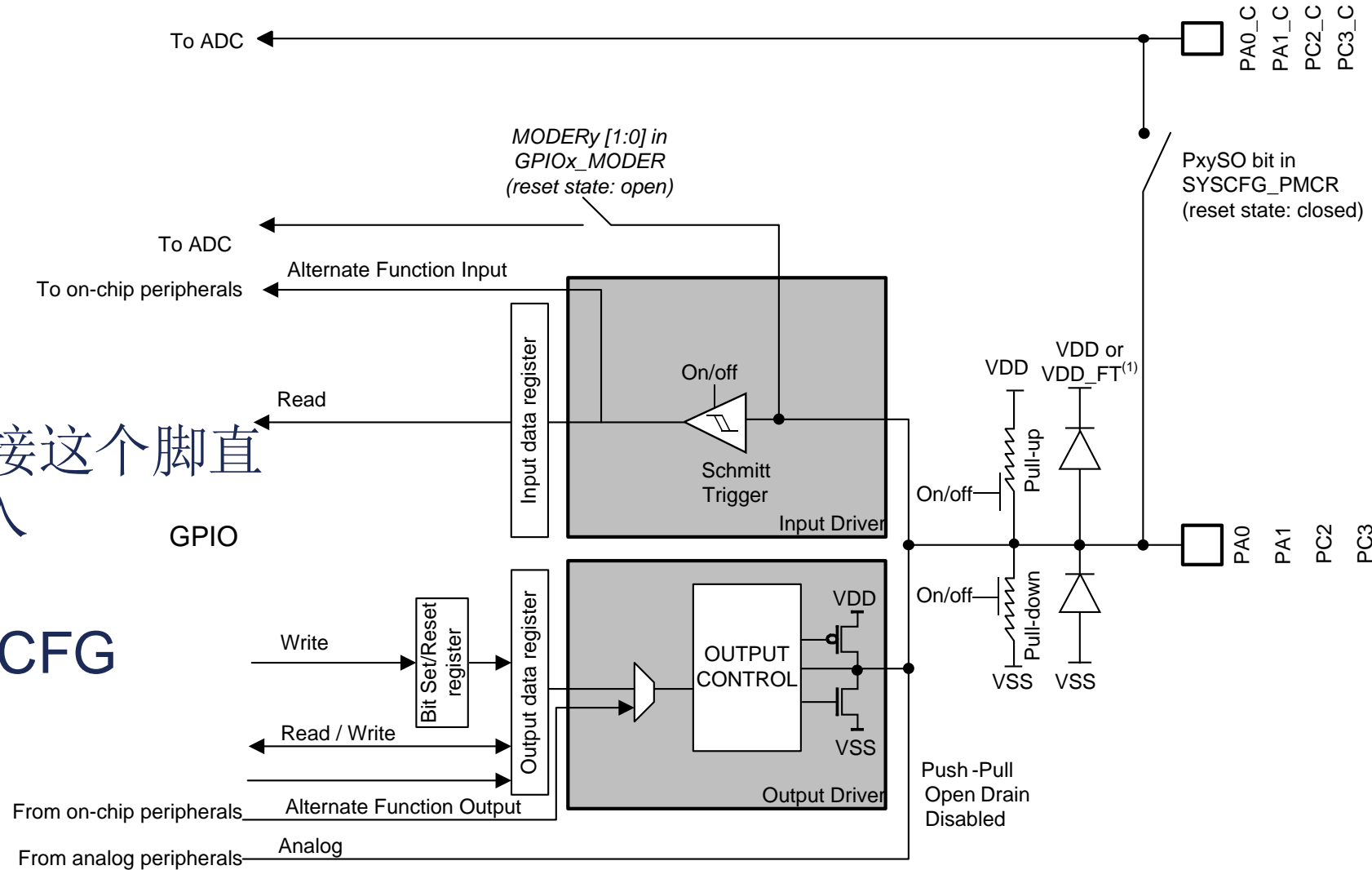
79

- 一些引脚直接连接到 ADC:

- PA0_C,
- PA1_C,
- PC2_C
- and PC3_C

- 一个模拟开关允许连接这个脚直接到 ADC analog 输入

- 连接的处理通过 SYSCFG

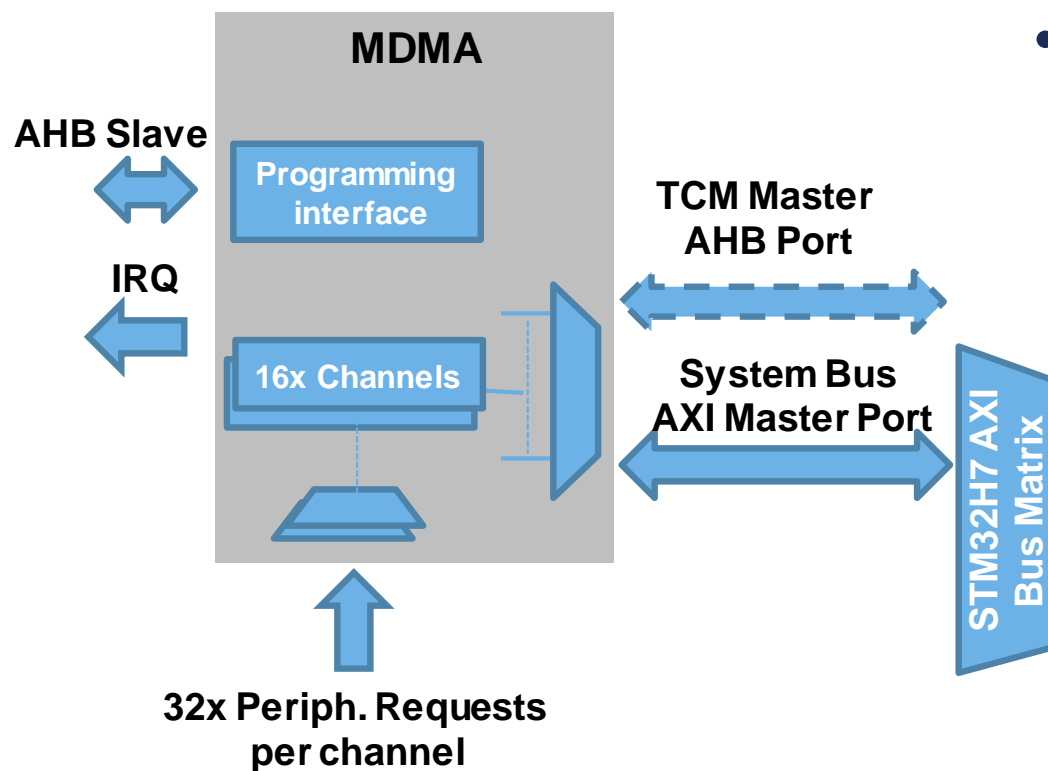




STM32H7 - MDMA

主直接内存存取控制器 Master direct memory access controller (MDMA)

版本 1



• STM32H7 MDMA 特征

- 双 master 总线
 - 64-bit AXI and 32-bit AHB master port
- 灵活的和独立的配置
- 硬件和软件优先级管理
- 可配置的数据传输模式
 - Peripheral-to-Memory, Memory-to-Peripheral, and Memory-to-Memory modes

应用益处

- DMA 支持块传输和链表
- 减少CPU 在数据传输管理中的负载
- 简单的集成

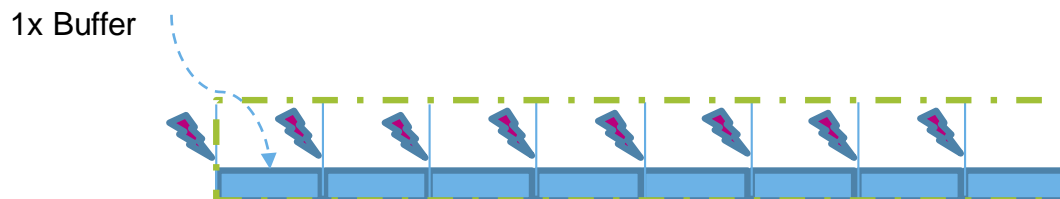
- 每个MDMA通道可以执行
 - **单块传输**: 一个块传输（高达64Kbyte），传输结束，DMA通道禁止.
 - **重复块传输**: 在禁止通道前，一定数目的块传输（高达 4096 块)
 - **链表传输**: 当前数据传输完成后，新的控制块结构从内存装入，并且一个新的块传输启动。
- 单个通道的灵活性
 - **源和目的**，独立的加，减，或不变。
 - **源和目的**，独立的传输大小和增减大小
 - **Endianness eXchange**: Byte, Half-word and Word

- MDMA 支持单一或增量突发传输,
- 软件可配置的突发大小, 多达128 bytes
- 最大突发数据大小128 bytes. 对于数据量较大的突发长度限制在128级FIFO, 将用于存储临时需要传送的数据
 - e.g. 16x64-bit or 32x32-bit
- 对于TCM内存访问, 仅当增量和数据大小相同且小于或等于32位时才允许突发访问.

- 单个请求，数据的大小组成传输，为下列中的一种

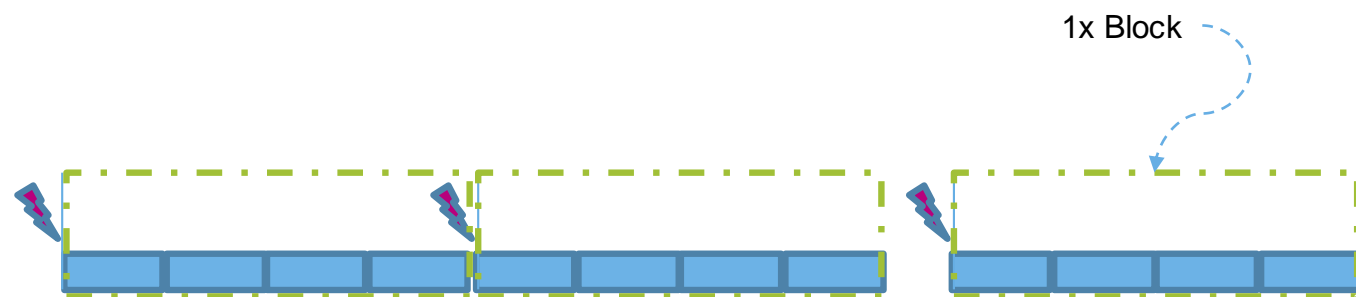
1. 缓冲传输大小

- TRGM='00'



每个请求一个缓冲被传输

- 单个请求，数据的大小组成传输，为下列中的一种：
 1. 缓冲传输大小
 - TRGM='00'
 2. 块的大小
 - TRGM='01'



每次请求一个块被传输

- 单个请求，数据的大小组成传输，为下列中的一种：

1. 缓冲传输大小

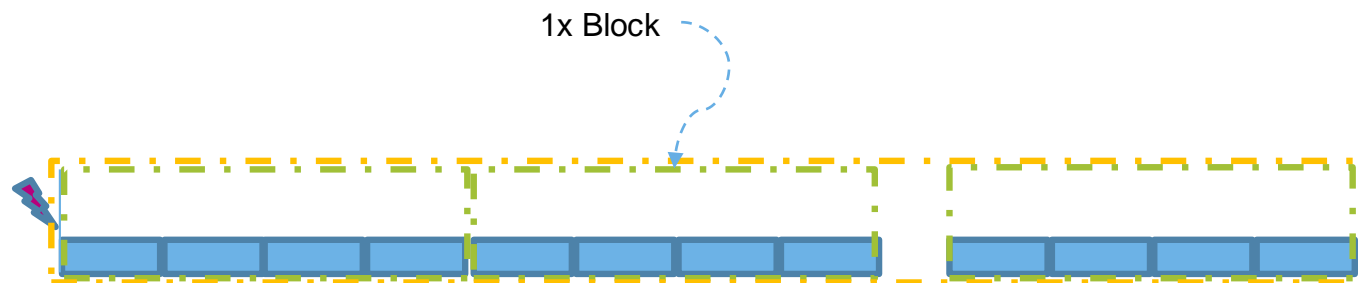
- TRGM='00'

2. 块的大小

- TRGM='01'

3. 重复块

- TRGM='10'



每次请求几个块被传输

- 单个请求，数据的大小组成传输，为下列中的一种：

1. Buffer 传输大小

- TRGM='00'

2. 块的大小

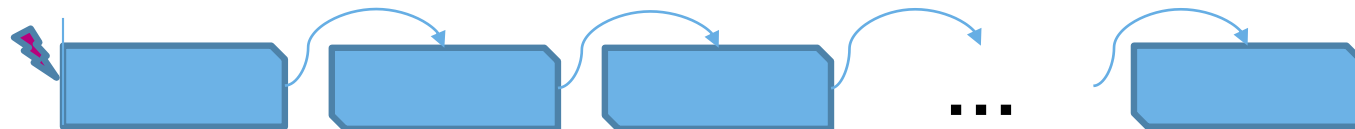
- TRGM='01'

3. 重复块

- TRGM='10'

4. 完成通道数据

- TRGM='11'



一个请求触发开始传输，直到通道链表指针为NULL

- **buffer transfer** 是通过MDMA进行的最小的数据大小传输,传输过程不会有新的仲裁, 如果MDMA通道之间的有请求。
- 块传输的情况,在传输完一个单独的 **buffer**, MDMA将进入一个新的仲裁阶段, 在新的外部请求和内部已有记忆中选择一个。
 - 如果没有其他更高的优先级,通道请求被激活时,一个新的 **buffer** 传输将启动同样的通道
 - 在一个块有多个**buffer** 传输段的情况, 如果没有更高级的MDMA请求, 每个 **buffer** 传输之后完成的通道仲裁会被阻塞。

- 两种数据的数组大小对 MDMA 和 应用 会有影响
 1. **Buffer transfer size**:它定义了MDMA这个级别，不可中断的最大的数据传输长度
 - 一个通道，这个长度的数据传输完成，需要检查之前其它通道的请求
 2. **Burst size**: 它定义了总线仲裁级别上，不可中断的最大的数据传输长度
 - 这是在 **Burst** 模式下，可能的数据传输的长度。
 - 它可以阻塞其它的 masters 获得访问总线
- 正确的选择 burst and buffer transfer size ，在实时的应用中，对其它的 MDMA channels and masters 是非常重要的。

- **Buffer transfer** 是在一个通道上，一个MDMA请求事件，传输的最小逻辑上传输的数据量。
 - 在当前通道上，需要传输的数据总量,是在 MDMA请求后由触发模式决定。
- 要传输的数据项的数目，它们的宽度（8位，16位，32位或64位）和用于数据传输的 **burst** 长度是软件可编程。
- 通过在给定掩码地址写掩码数据值，DMA/外设请求事件之后的服务请求被响应。
 - 主要用于清除请求事件源标志。

- 一个块是一个连续的数组,多达 64 Kbytes,由连续的 buffer transfers 传输完成
- 每个块的数据由 “start address” and “block length” 定义。
- 根据触发模型配置,一个整块传输可以由一个外设/ DMA请求触发。
 - 如果没有其他高优先级通道请求激活, 将用相同的通道启动一个新的 buffer transfer

- 当一个块传输完成后,可以执行三个操作:
 - 块是一个重复的块传输的一部分: **block length** 重载和 **new block start address** 的计算基于块重复地址更新寄存器中的信息。
 - 它是一个单一块或是重复的块传输的最后一块:下一个块信息从内存加载
 - 当前MDMA通道, 最后一个块传输完成: 这个 通道是禁用的 并且 没有进一步的 MDMA请求将被接受。

- 块重复模式允许 **repeat a block transfer**,用不同的源和目标的“**start addresses**”。
- 当重复块模式处于活动状态(重复计数非0),在当前的块传输完成，块的参数将被更新。
 - 重新装入数据字节的块数目到传输值 (BNDT)
 - 更新块的源和目的地址值，根据块的重复源 / 目的地址更新配置(BRSUM/BRDUM)
 - 重复计数器减 1
- 当重复块计数器达到0,最后一块将被视为一个单独的块传输。

MDMA Linked 链表模式

94

- 链接列表模式允许从 **Channel Link Address register (CLAR)** 所给的地址，加载新的 **MDMA configuration**，
- 此操作后,通道准备接受新的请求,如前面定义的块/重复块模式,或继续传输，如果触发模式设置为 **Complete channel data (TRGM=11)**，
- 当加载触发和总线选择寄存器(**TBR**)值时，自动触发源可能会改变。
- 如果触发模式设置为**TRGM=11**:触发模型和选用**SW**请求不能被改变，

MDMA Linked链表模式

95

- 通道配置(channel link address LAR)必须在AXI地址空间
- LAR value 必须用 Double Word address 对齐, i.e. $LAR[2:0] = 0x0$

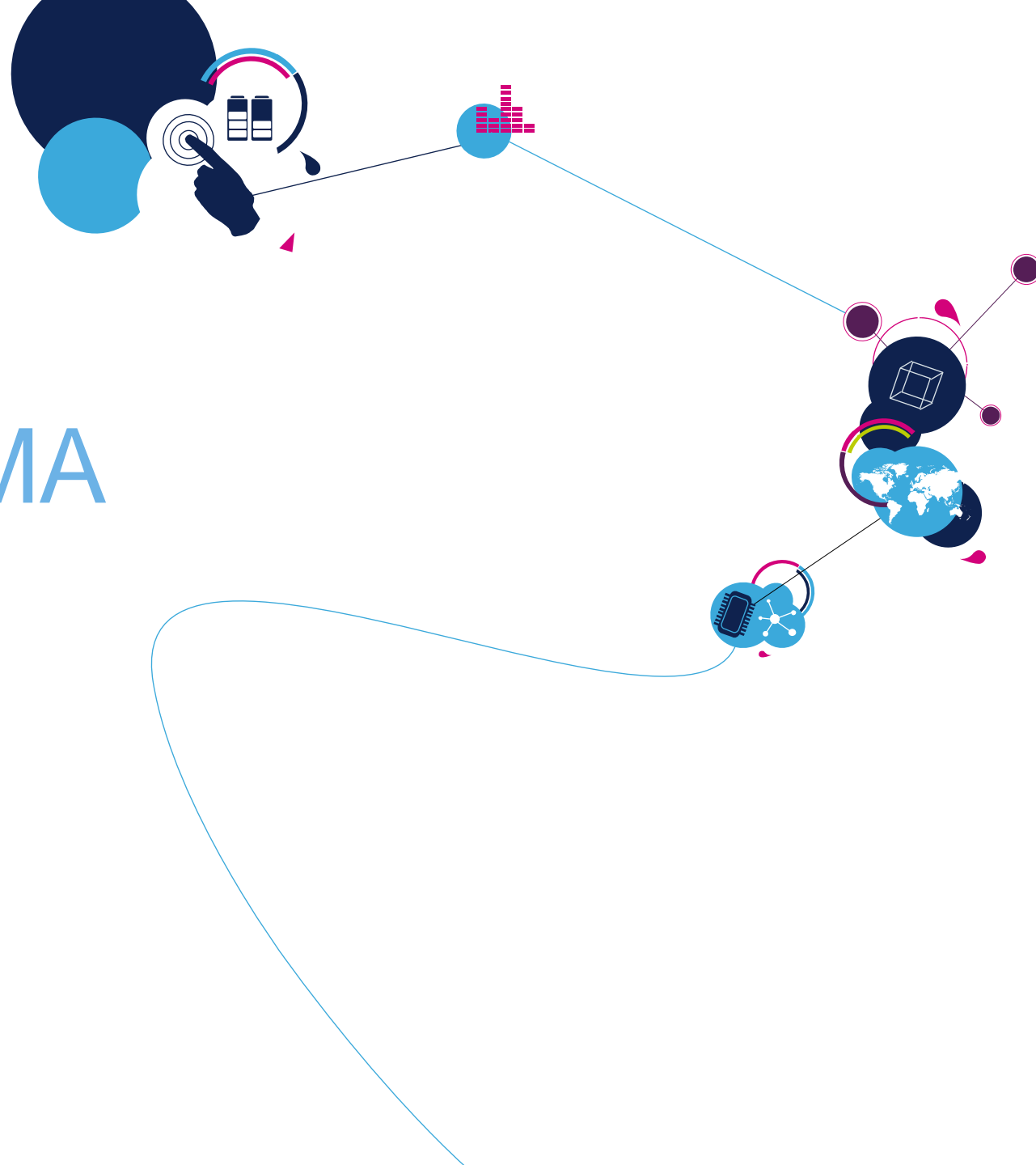
Register (32-bit word)	Offset from Link Address register	Description
CTCR	0x00	Transfer Configuration register
CBNDTR	0x04	Block number of data register
CSAR	0x08	Source address register
CDAR	0x0C	Destination address register
CBRUR	0x10	Block Repeat address Update register
CLAR	0x14	Link Address register: Next descriptor
CTBR	0x18	Trigger and Bus selection Register
CMAR	0x1C	Mask address register
CMDR	0x20	Mask Data register

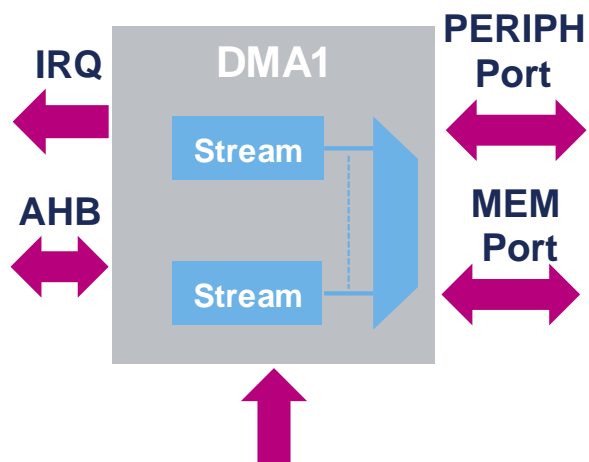
- 从其他 (DMA1 / DMA2和BDMA)收集数据和使数据在D1域的CPU可用 (DTCM或AXI-SRAM)， MDMA 是有用的。
- DMA链表， 执行一组DMA传输， 且不需要CPU干预
 - 可以用来准备数据给其它的 DMAs ， 然后设置DMA配置开始传输。
 - 它是用来支持分散/聚合。这意味着源和目标区域不需要占用连续的内存区域。源和目标数据区域是由一系列的链表描述符控制数据块的转移。

STM32H7 - DMA

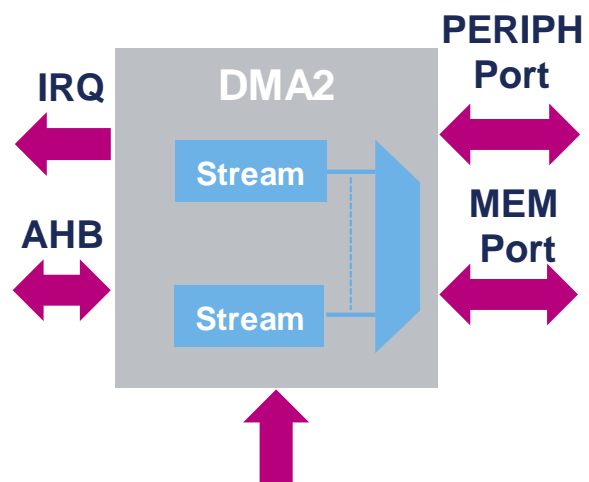
直接内存存取控制器(DMA)

版本 1





Up to 105 request per stream



Up to 105 request per stream

• DMA1/2 特征

- 双 AHB master 总线
- 灵活配置
- 硬件软件优先级管理
- 配置数据传输模式
 - Peripheral-to-Memory, Memory-to-Peripheral, and Memory-to-Memory modes

应用好处

- DMA 支持D2 域的 timers, ADC, 和 通讯外设
- 把CPU从数据传输管理中卸载
- 简单的集成

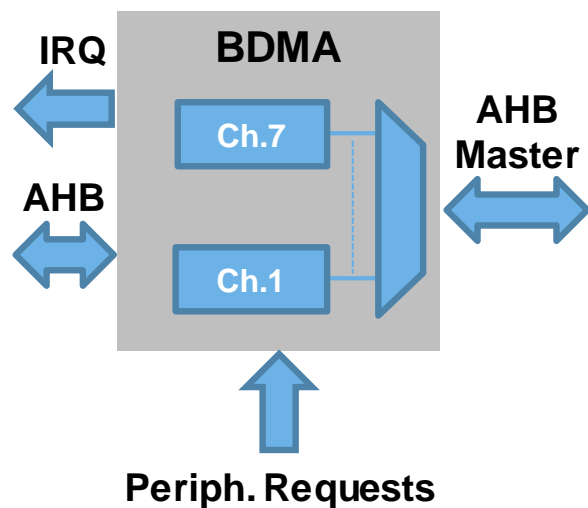
- 参考和这个外设有关的这些外设培训:
 - DMAMUX (DMA request router)
 - MDMA (Master direct memory access controller)

STM32H7 - BDMA

基本的直接内存访问控制器 (BDMA)

版本 1





• STM32H7 BDMA 特征

- 灵活的配置
- 硬件软件优先级管理
- 可配置的数据传输模式
 - Peripheral-to-Memory, Memory-to-Peripheral, Peripheral-to-Peripheral, and Memory-to-Memory modes

应用的好处

- DMA 支持 D3 域的外设
- 当 CPU 进入 CStop 并且 D1 或 D2 域处在 Dstandby 时，执行数据传输

- 7 个独立的可配置通道
 - 每个通道都可以 Hardware request or software trigger
 - 软件可编程的优先级与硬件优先级的平等
- 独立的和灵活的通道配置
 - 完全可编程的通道 (data format, increment type, address)
 - 独立的通道中断标志 (half transfer, transfer complete, transfer error, global flags)
 - 支持 circular buffer 管理.
- 在出现总线访问错误的情况下，错误通道自动禁止

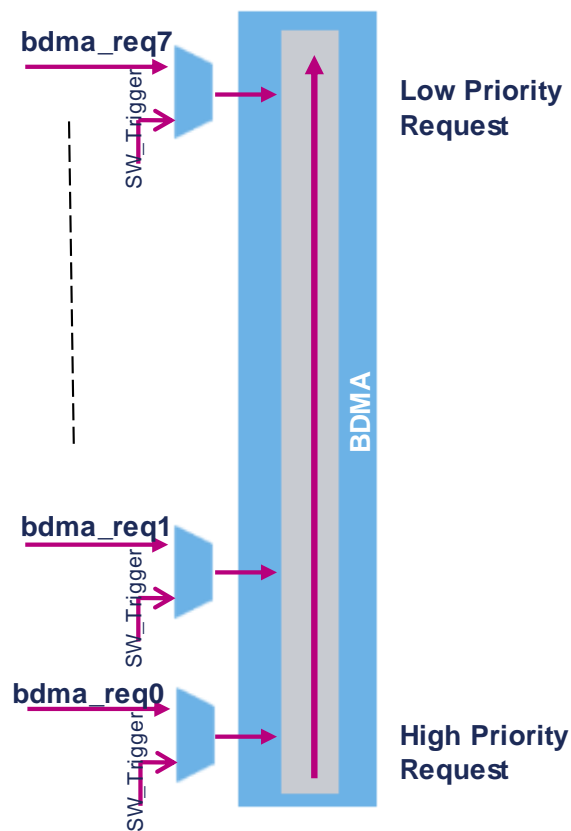
- 可编程特征
 - 独立的源和目的数据大小 (8-bit/16-bit/32-bit)
 - 独立的源和目的地址
 - 独立的源和目的地址指针增量
 - 可编程数字传输的数据多达65535个请求
- 循环模式
 - 处理连续数据流的循环缓冲区
 - 源和目的地址自动重载
 - 数据传输大小自动重载

- Memory-to-Memory mode
 - 一旦启用通道，传输就开始 (没有硬件请求)
- Peripheral-to-Memory, Memory-to-Peripheral, Peripheral-to-Peripheral
 - 传输发生在每个硬件的请求时
 - 一旦传输完成后,请求被响应

BDMA requests 映射

105

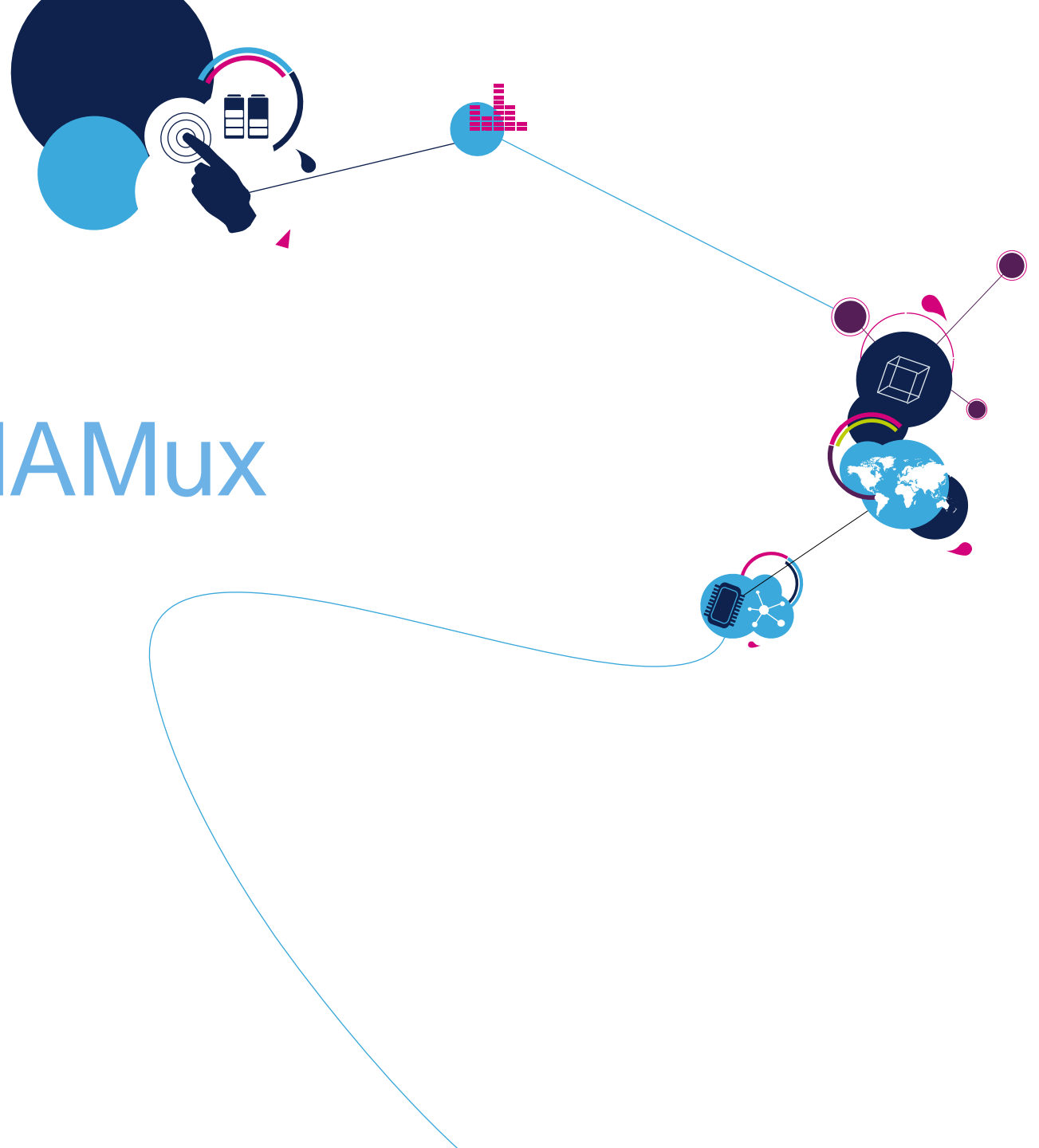
- BDMA1 控制器提供 7 channels 的访问
 - **New:** 通过一个多路复用器来映射外设请求(Not OR gate)
 - 每个通道有独立软件触发

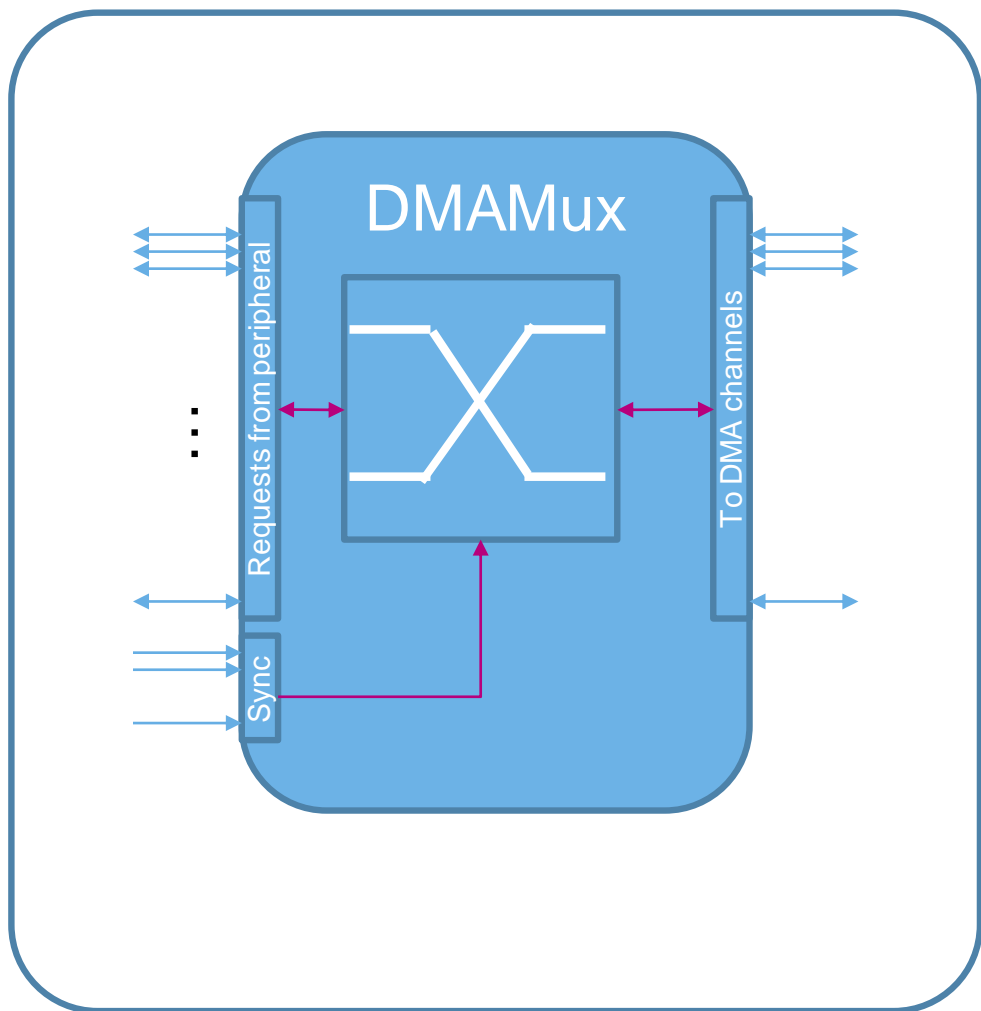


STM32H7- DMAMux

DMA 请求路由

版本 0.1





- DMA请求路由 (DMAMux) 管理:
 - 分配DMA请求线到外设
 - 在同步输入时，请求转发与事件同步
 - 请求链接使用DMA请求计数器，并且事件生成给DMA

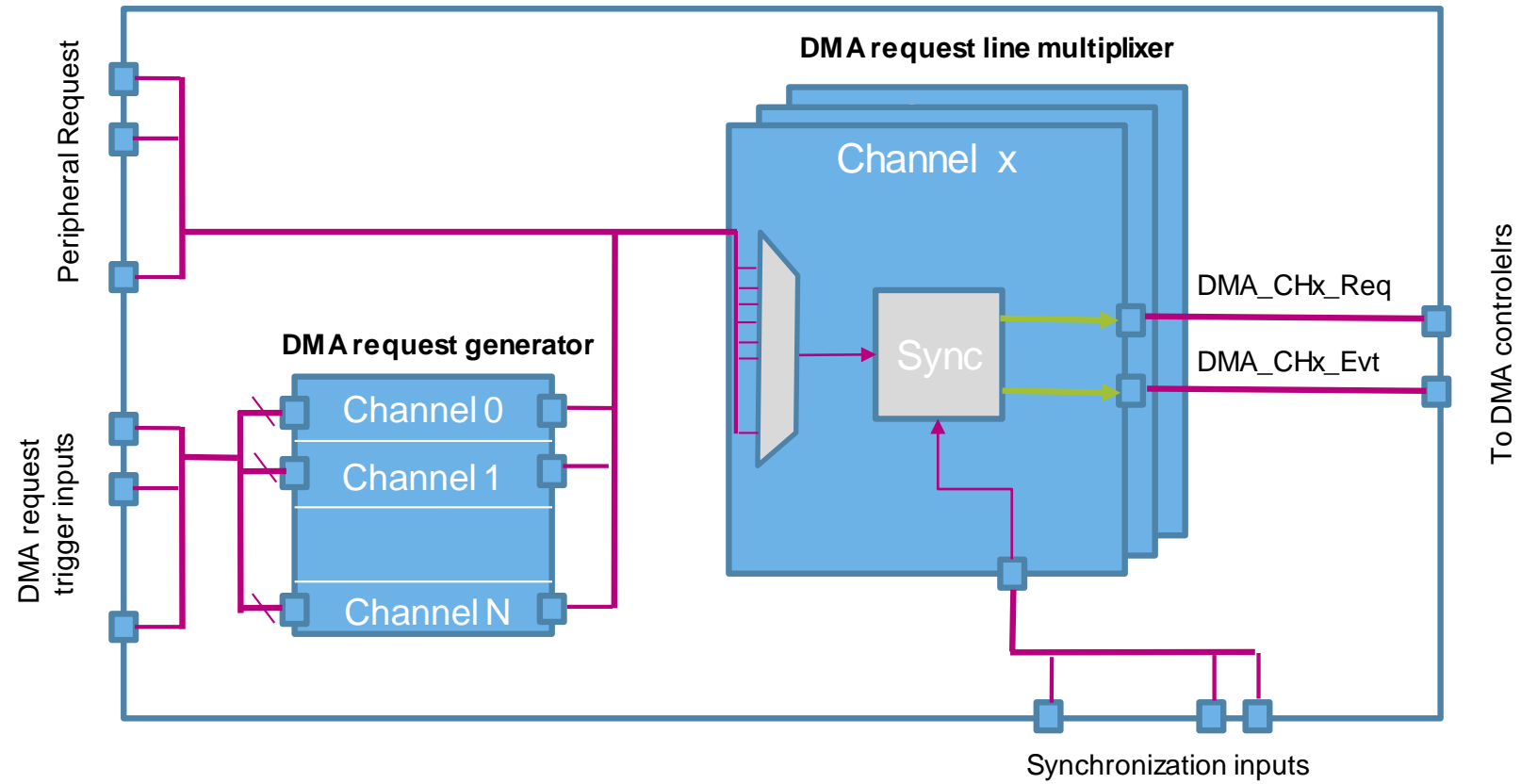
应用好处

- DMA请求映射的选择有高灵活性
- 外部和内部的DMA请求管理

- DMA 请求线多路复用器
 - 灵活的映射外设请求到 DMA controller channel
 - 多达 107 个输入 DMA request lines
 - 同步操作模式,有单独的 enable bit
 - 多达16 个同步输入选择器
 - 选定同步输入时, 有事件 overrun flag
- DMA 请求生成通道:
 - DMA 请求生成, 在异步事件时
 - 多达 32 个 DMA 请求触发输入选择
 - Event overrun flag, 给选定的DMA请求触发输入

DMAMux 框图

109



无条件的操作模式

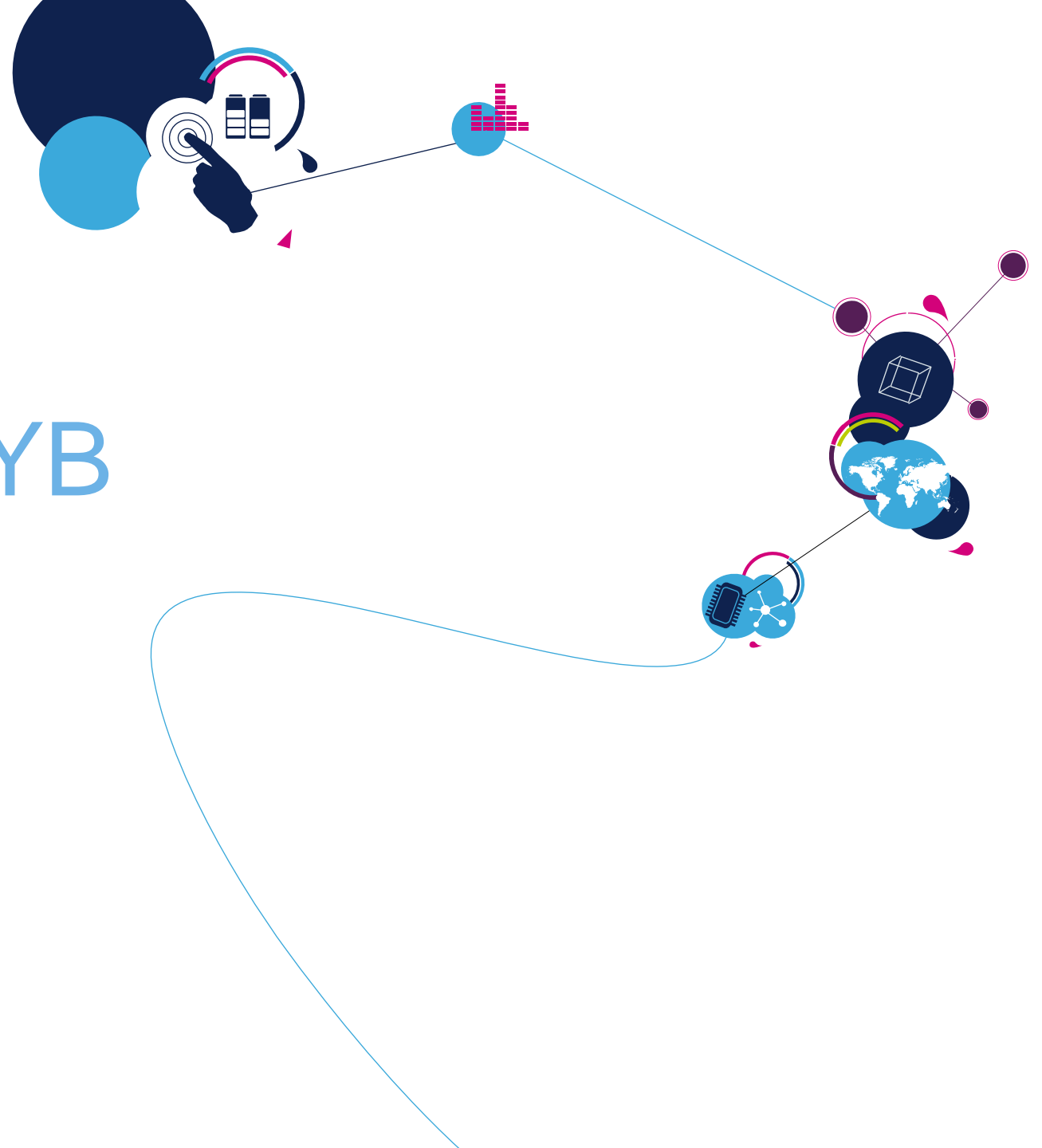
- 在无条件的操作模式,连接一个输入的DMA请求到多路通道输出, 是通过下列方式选择
 - 通道控制寄存器(DMAMUX_CxCR)的 程序号 DMAREQ_ID
 - 对于每个外设请求行,ID是受影响的。
 - DMAREQ_ID=0x00 对应于没有DMA请求行选择
- DMAMUX通道的配置后, 相应的DMA控制器通道上可以配置到路由。
 - 不允许配置两个不同的DMAMUX通道, 选择相同的DMA请求来源

每个通道单独可以设置为同步操作模式

- 在同步操作模式下,输入的DMA请求连接到多路复用器通道输出的条件是
 - 同步输入事件通过控制寄存器的SYNC_ID 字段选择
 - 同步事件可以是选择输入的上升沿,下降沿或上下沿
 - 和内置的 DMA请求计数器
- 同步事件后,每个服务DMA请求, DMA请求的计数器递减。直到下溢:
 - DMA request counter 自动重载控制寄存器 NBREQ field 的值
 - 并且 DMA请求行禁止与多路通道的输出连接

STM32H7- DLYB

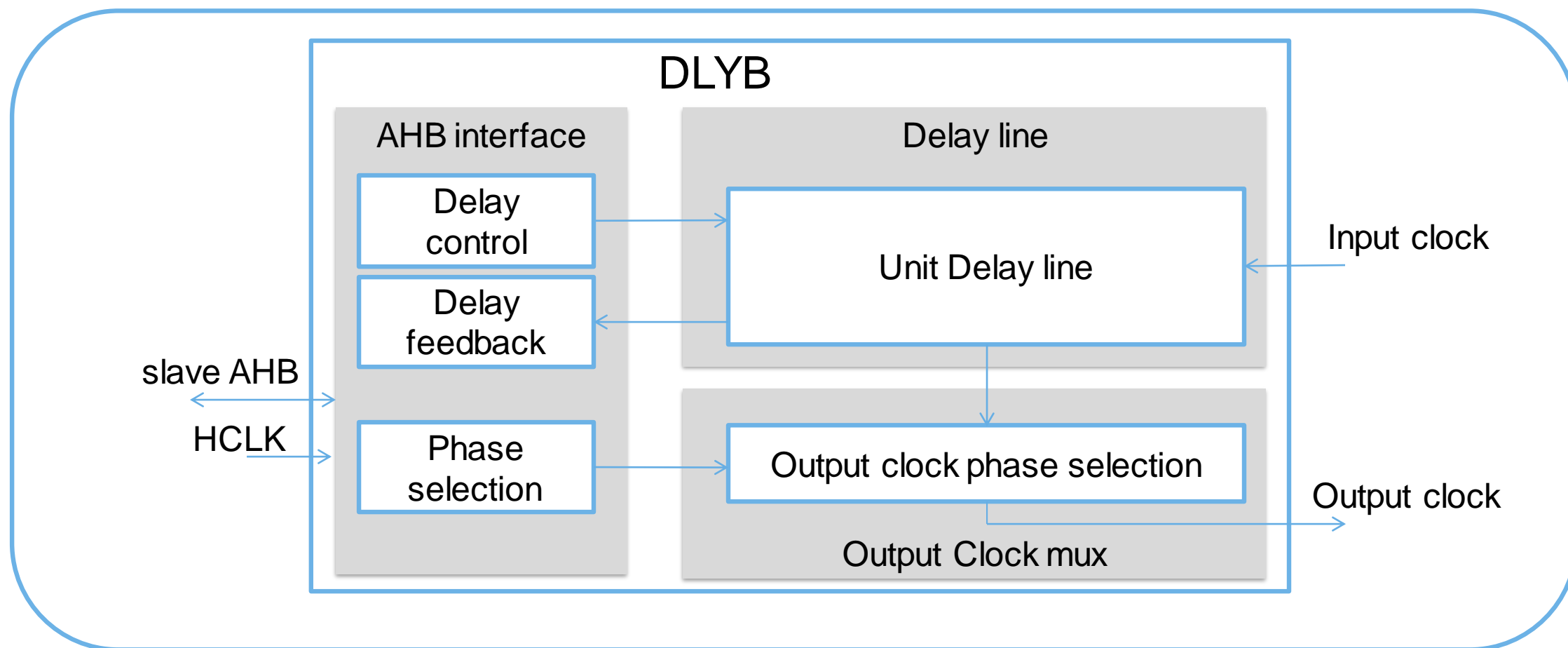
延迟块



- 为SDMMC和 QSPI 数据通信接口，产生一个延迟采样时钟
- 输入时钟频率范围 25 MHz to 208 MHz.
- 固件控制
- 没有电压和温度漂移补偿.

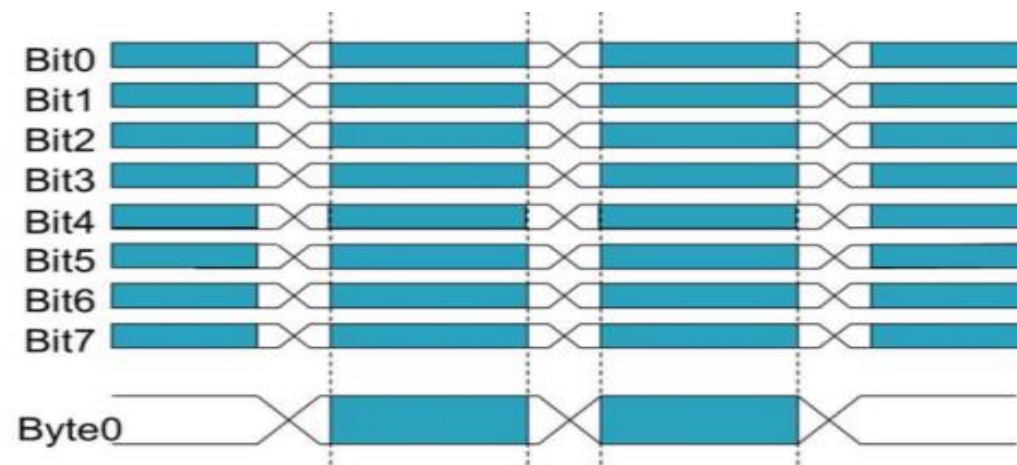
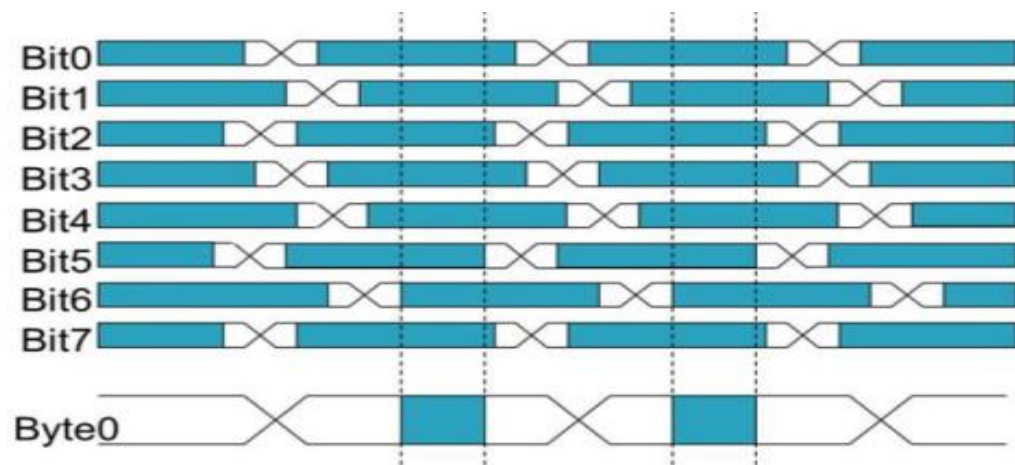
应用好处

- 支持SDMMC卡可变延迟
- 提高SDMMC和QSPI的时序裕量

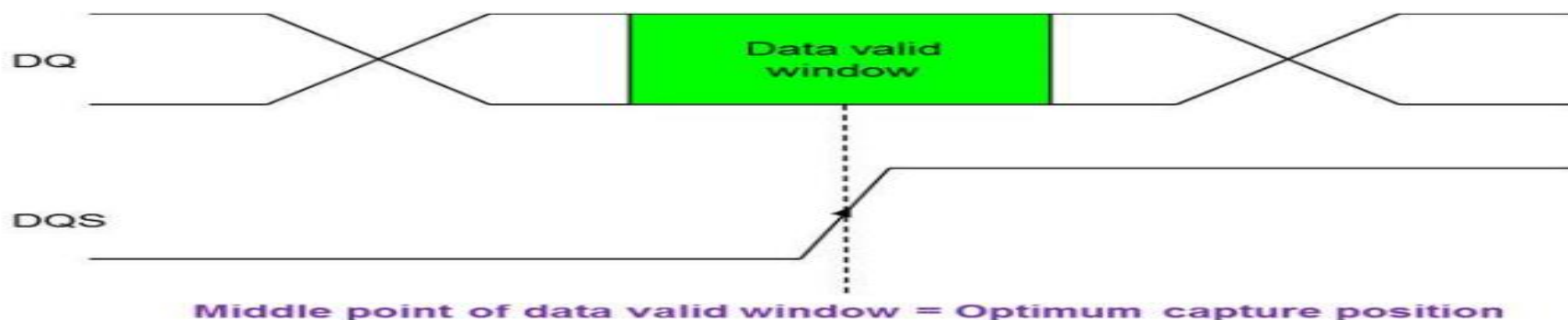


长度均衡

115



- 最终，同一组的信号必须匹配设置和持有时间

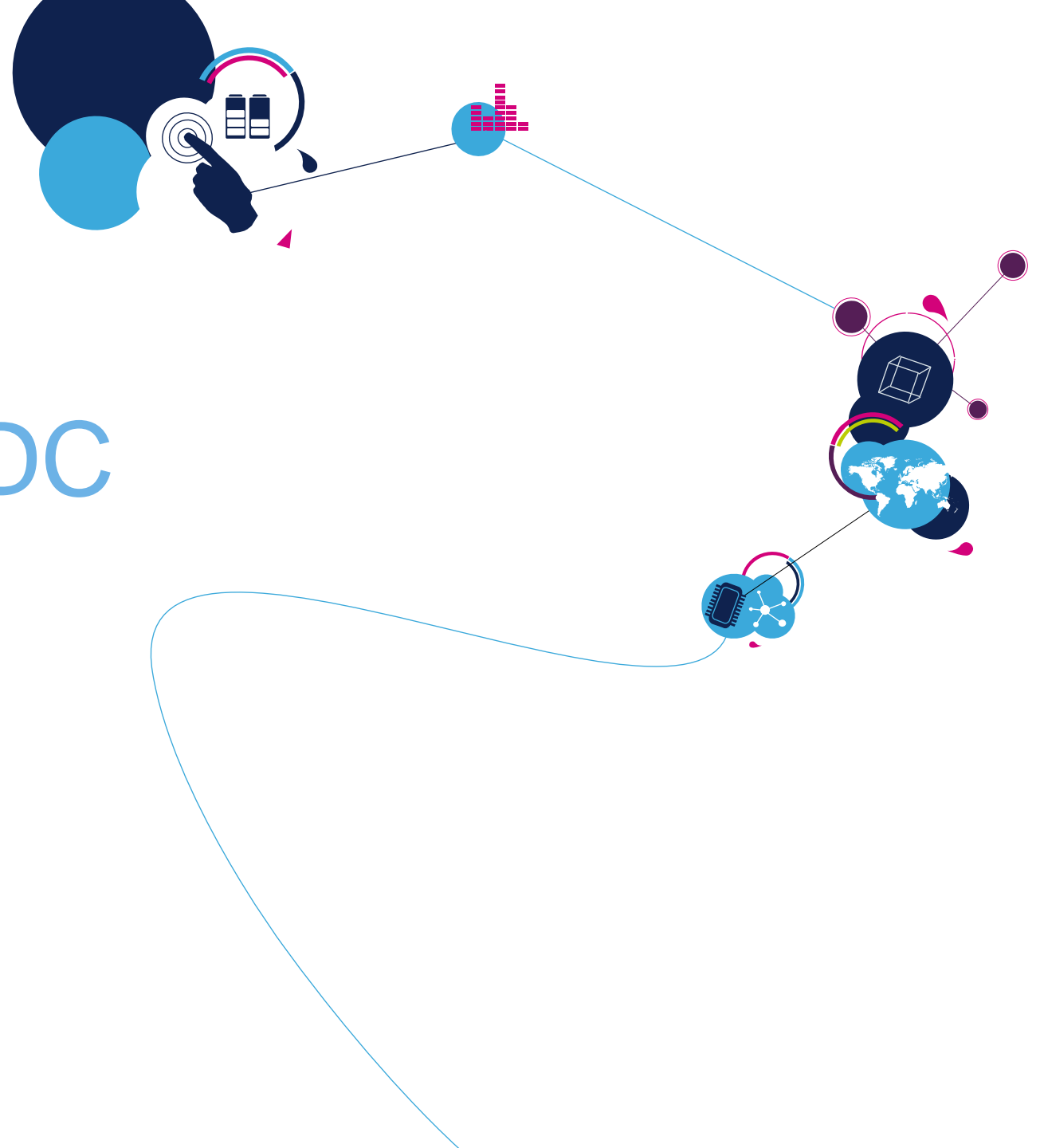


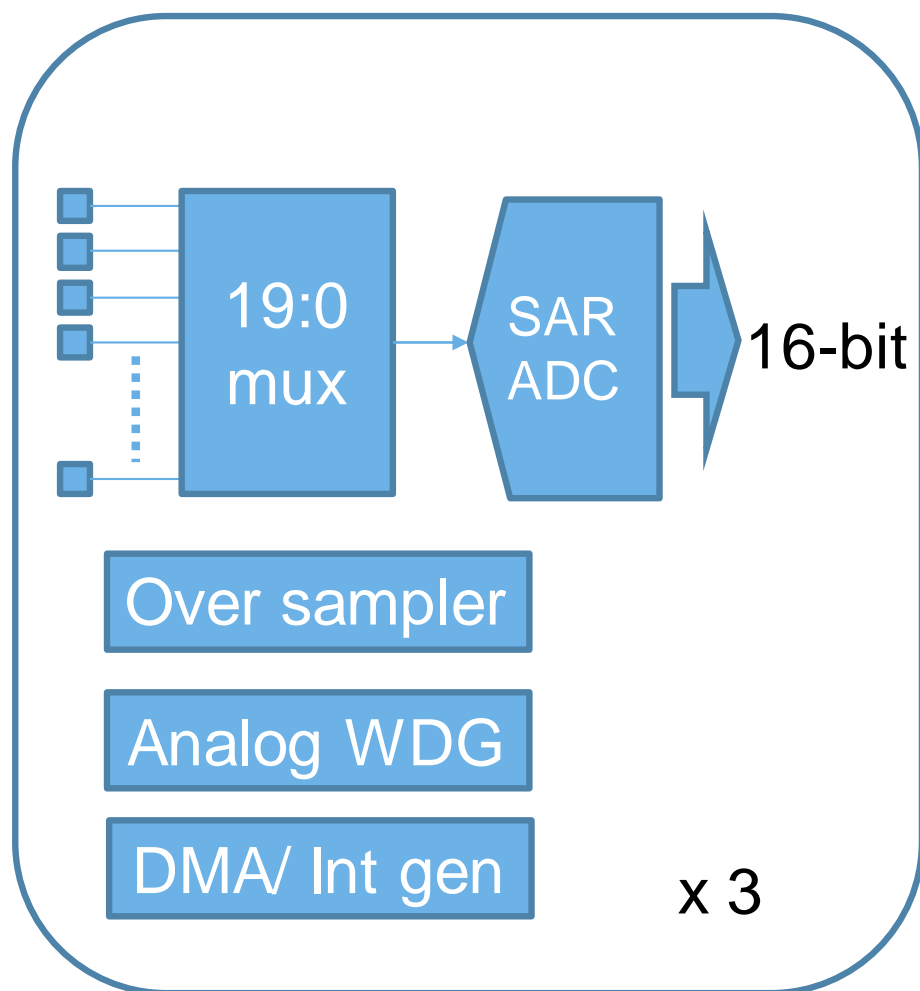
- 必须考虑整个信号通路:衬底,PCB & 走线的长度

STM32H7 – ADC

Analog-to-Digital Converter

Revision 1





- 提供模拟到数字的转换

- 三个 ADCs 高达24个输入通道
- 16位结构,高达21-bit过采样
- 因为噪音水平, ENOB 限制在14 bit
- 8 Msamples/s max. (14-bit)
- 每个ADC有3个模拟看门狗。
- DMA 请求生成
- 中断生成

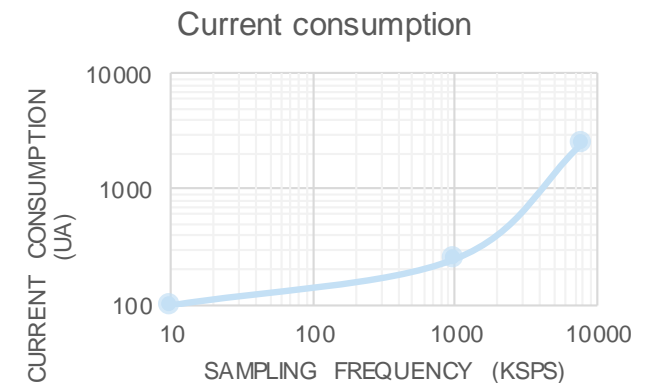
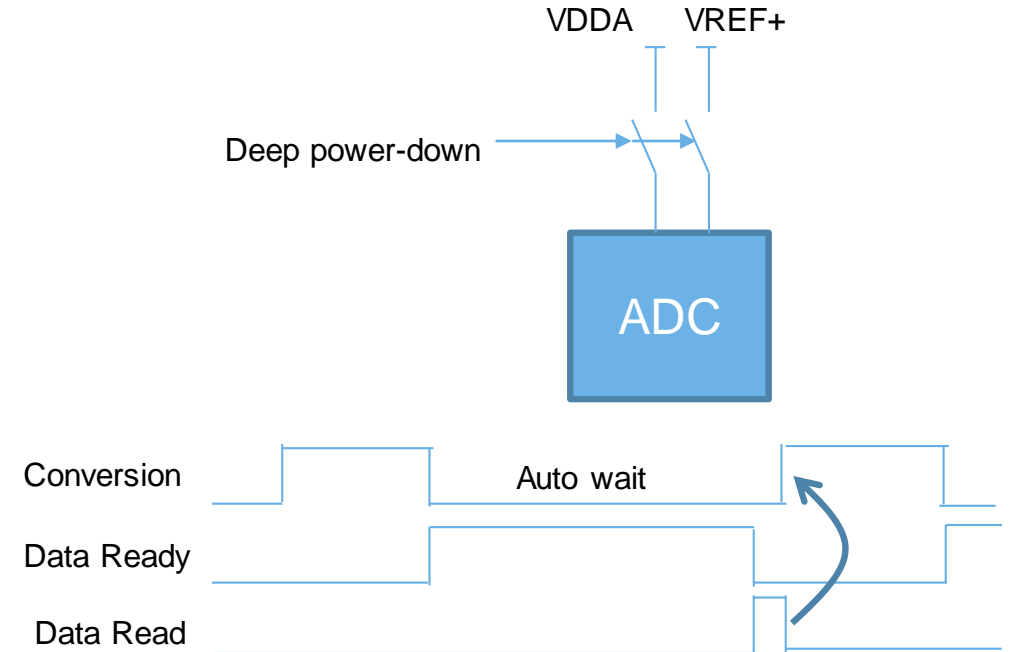
应用好处

- 超低功耗: 210 μ A @ 1 Msample/s
- 灵活的触发, 数据管理, 减轻CPU的负载

ADC 单元	3 模块
输入通道	高达24个外部通道(GPIOs)、单端/差分
技术	16-bit 逐次逼近
转换时间	125nS, 8 Msamples/s (when $f_{\text{ADC_CLK}} = 72 \text{ MHz}$, 14bit)
功能模式	单次、连续扫描、不连续或注入
触发器	软件或外部触发(for Timers & IOs)
特殊功能	硬件过采样过密,模拟看门狗
数据处理	中断生成、DMA请求
低功耗模式	深度掉电,自动延迟、功耗取决于速度

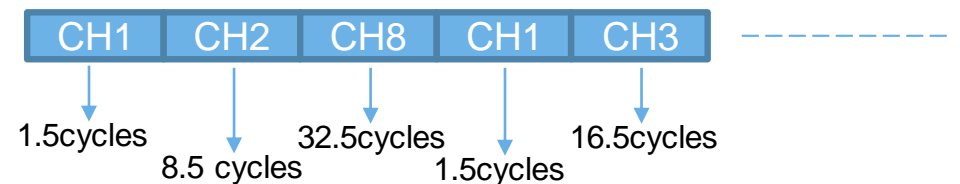
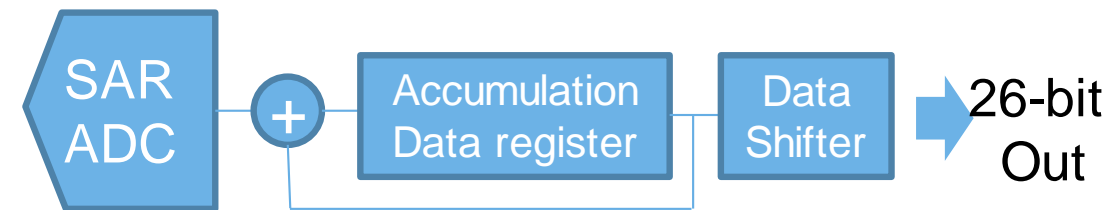
几种低功耗功能实现

- 深度掉电模式
 - 内部供电ADC可以通过电源开关禁用,减少泄漏
- 自动延时转换
 - ADC可以自动等直到最后数据读取
- 功耗取决于采样时间
 - 2.5mA @ 8 Msamples/s,
 - 250 μ A @ 1 Msample/s,
 - 100 μ A @ 10 ksamples/s



一些高性能功能实现

- 8.00 Msamples/s for 72 MHz ADC clock @14bit
- 交错模式可以支持 16 Msamples/s
- 硬件过采样
 - 累加器和位移器能输出26-bit数据，不需要CPU的支持
- 灵活的定序器
- 自动校准减少抵消,更好的线性



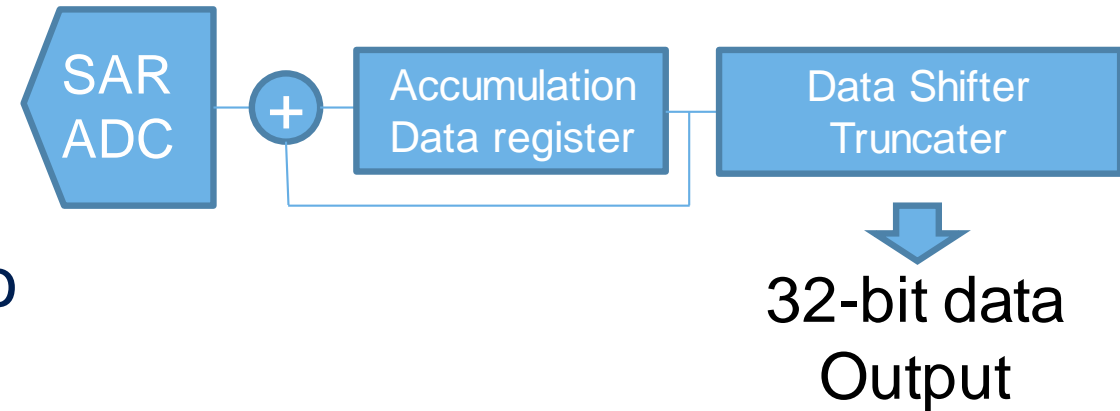
转换速度和精度相关

- ADC 需要最小 $1.5_{\text{ADC_CLKs}}$ 采样周期 和 $7.5_{\text{ADC_CLKs}}$ 转换(14 bits).
- 最大72 MHz时钟与9 cycles 的结果是 8 Msamples / s
- 更低精度，更高速度
 - 16-bit : $8.5_{\text{ADC_CLKs}}(+1.5) \Rightarrow 7.2 \text{ Msamples/s}$
 - 12-bit : $6.5_{\text{ADC_CLKs}}(+1.5) \Rightarrow 9 \text{ Msamples/s}$
 - 10-bit : $5.5_{\text{ADC_CLKs}}(+1.5) \Rightarrow 10.2 \text{ Msamples/s}$
 - 8-bit : $4.5_{\text{ADC_CLKs}}(+1.5) \Rightarrow 12 \text{ Msamples/s}$

Resolution	$t_{\text{Conversion}}$
16 bits	8.5 Cycles
14 bits	7.5 Cycles
12 bits	6.5 Cycles
10 bits	5.5 Cycles
8 bits	4.5 Cycles

数据预处理减轻CPU的负载

- 可编程的过采样率:
x2 to x1024
- 可编程数据移位和截断 Left shift of 0 to 15 bits, right shift of 0 to 11 bits.
- 高达32-bit 数据宽度
- 平均,降低数据率,信噪比提高
- 基本过滤



Oversampling ratio	Output resolution	Equivalent sampling frequency max
x1 (none)	16 bits	7.2Mps
x16	18 bits	450 ksamples/s
x256	20 bits	28.1 ksamples/s
x1024	21 bits	7.0 ksamples/s

每个ADC的特点

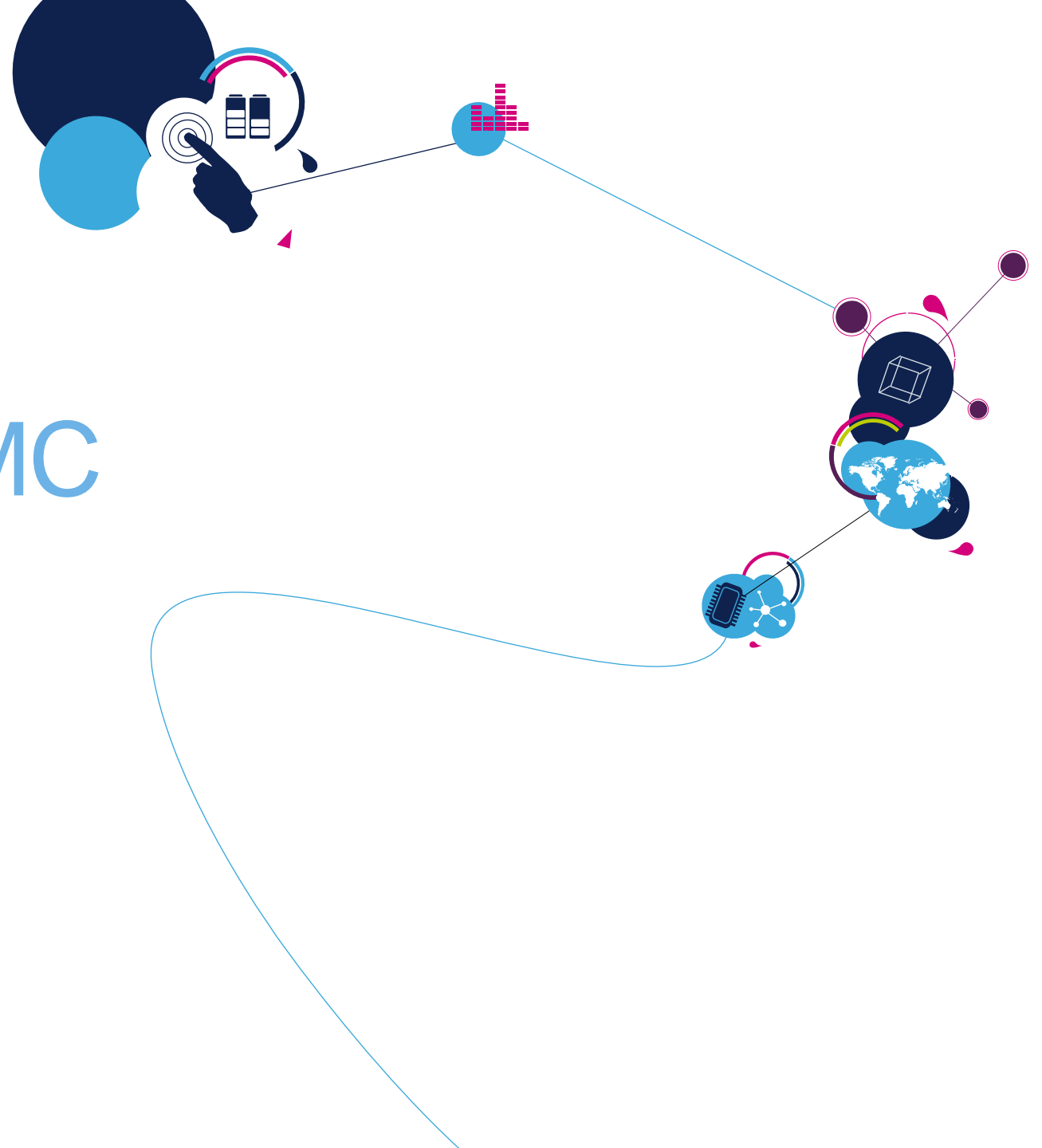
123

ADC features	ADC1	ADC2	ADC3
Dual mode	Master	Slave	-
Interconnect / Domain	AHB1/ Domain 2	AHB1/ Domain 2	AHB4/ Domain 3
Internal channel connection		DAC1 Out DAC2 Out	Bandgap Temp sensor VBAT/4

STM32H7 - FMC

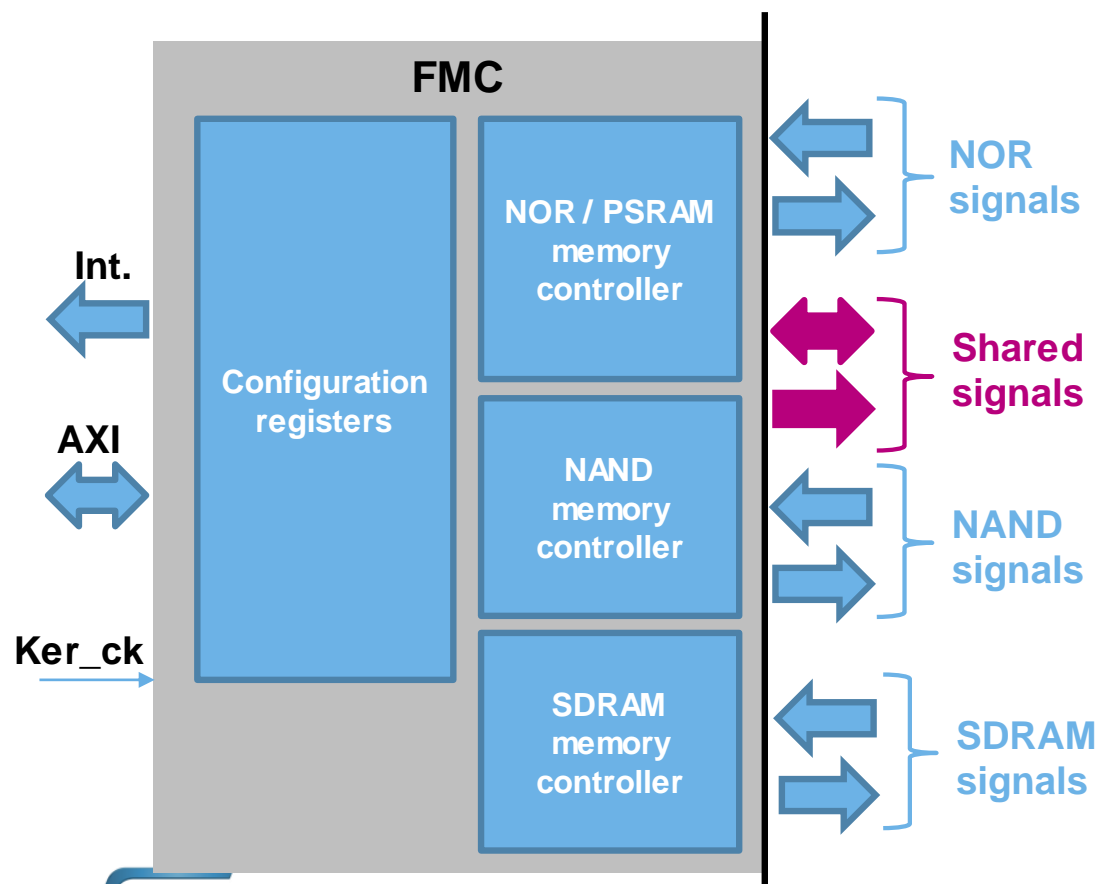
灵活的存储控制器

版本 1



- FMC 支持外部内存通过下列方法

- NOR Flash/PSRAM 控制器
- NAND memory 控制器
- SDRAM memory 控制器
- 独立的内核时钟



应用好处

- RAM 扩展
- Flash memory 扩展
- 并行接口(Intel 8080 / Motorola 6800)

- 完全独立 banks
 - 四个 banks 支持独立的外部 memories
 - 每个内存 bank 有独立的 Chip 选择
 - 每个内存 bank 有独立的 配置
- 配置灵活
 - FMC 外部访问频率高达 HCLK/2
 - 可编程的时序支持宽范围的设备
 - 8- ,16- or 32-bit 数据总线
 - 外部异步等待控制
 - 扩展模式（读时序和协议，不同于写时序）
 - 支持同步设备的突发模式访问(NOR Flash and PSRAM)

兼容各种各样的接口和**memories**

- 静态 memory-mapped 设备包括
 - Static random access memory (SRAM)
 - Read-only memory (ROM)
 - NOR / OneNAND Flash memory
 - PSRAM
- NAND Flash memory
 - 包括 ECC 硬件检查高达 8 Kbytes of data read/written
 - 3 可能的中断源 (level, rising edge and falling edge)
- SDRAM memory
 - 同步DRAM (SDRAM) 接口内存映射

兼容各种各样的接口和**memories**

- 平行的液晶显示模块 (Parallel LCD modules)
 - Intel 8080 and Motorola 6800

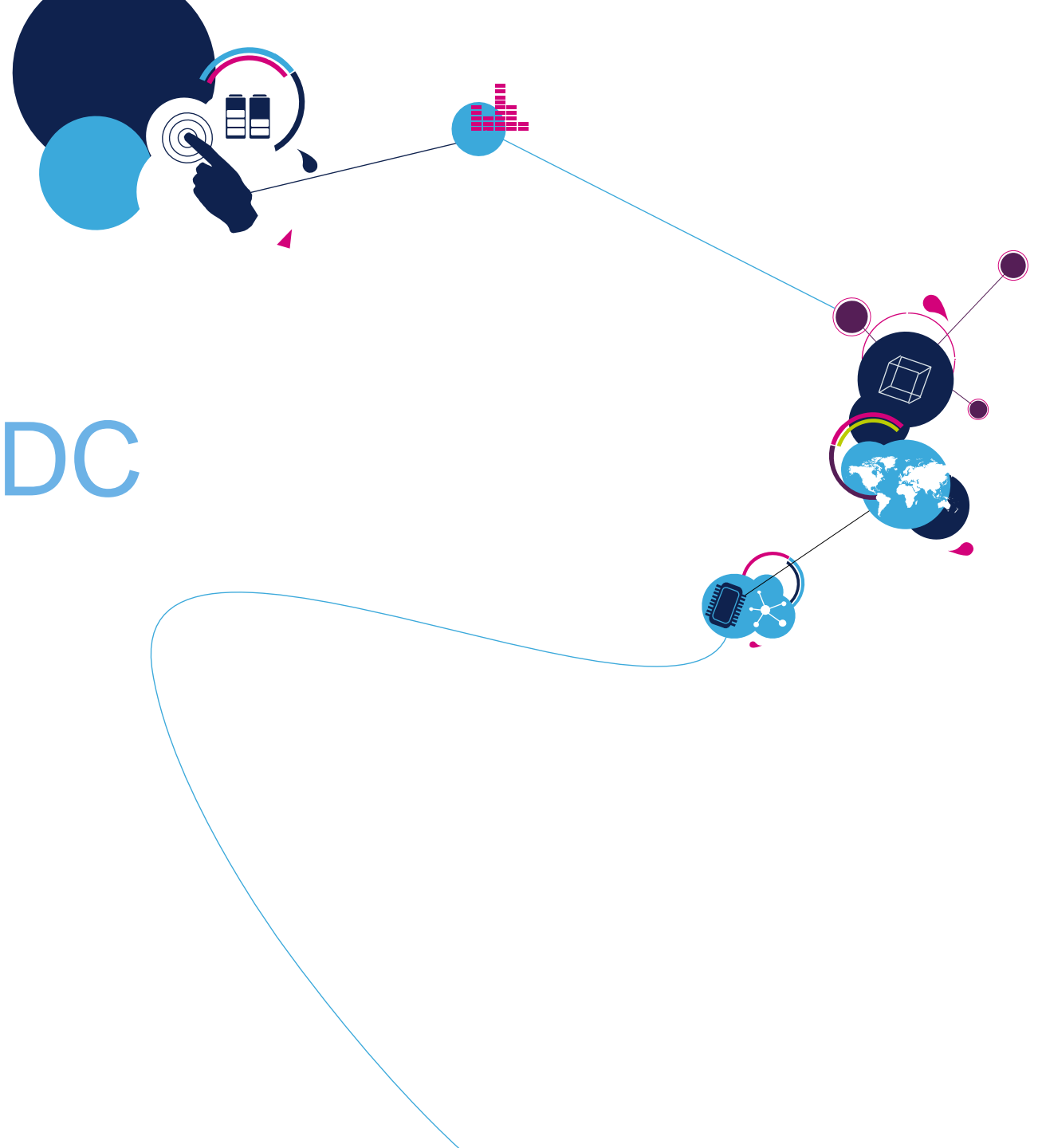
模式	说明
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
(D1)Stop	Frozen. Peripheral registers content is kept.
(D1)Standby	Powered-down. The peripheral must be reinitialized after exiting domain and system Standby mode.

- 参考下列外设培训.
 - Reset and clock control (RCC)
 - Interrupts (NVIC)
 - General-purpose inputs/outputs (GPIO)

STM32H7 - LTDC

LCD-TFT 显示控制器

版本 1



- LCD TFT显示控制器

- 高度可配置
- 标准并行RGB接口



应用好处

- 灵活的可编程显示参数
- 片上存储器或外部存储器可以作为帧缓冲

- 24-bit RGB 并行像素输出；每像素8位(RGB888)
- **AXI master interface** 用突发16字访问方式可以用任何系统内存
 - 每层有专用的 FIFO (深度是 64 word)
- 可编程的时序用于大范围的显示器接口
 - HSYNC width, VSYNC width, VBP, HBP, VFP, HFP
- 可编程极性：
 - HSYNC, VSYNC, not Data Enable, Pixel clock.
- 仅支持 TFT (no STN)

Thank you



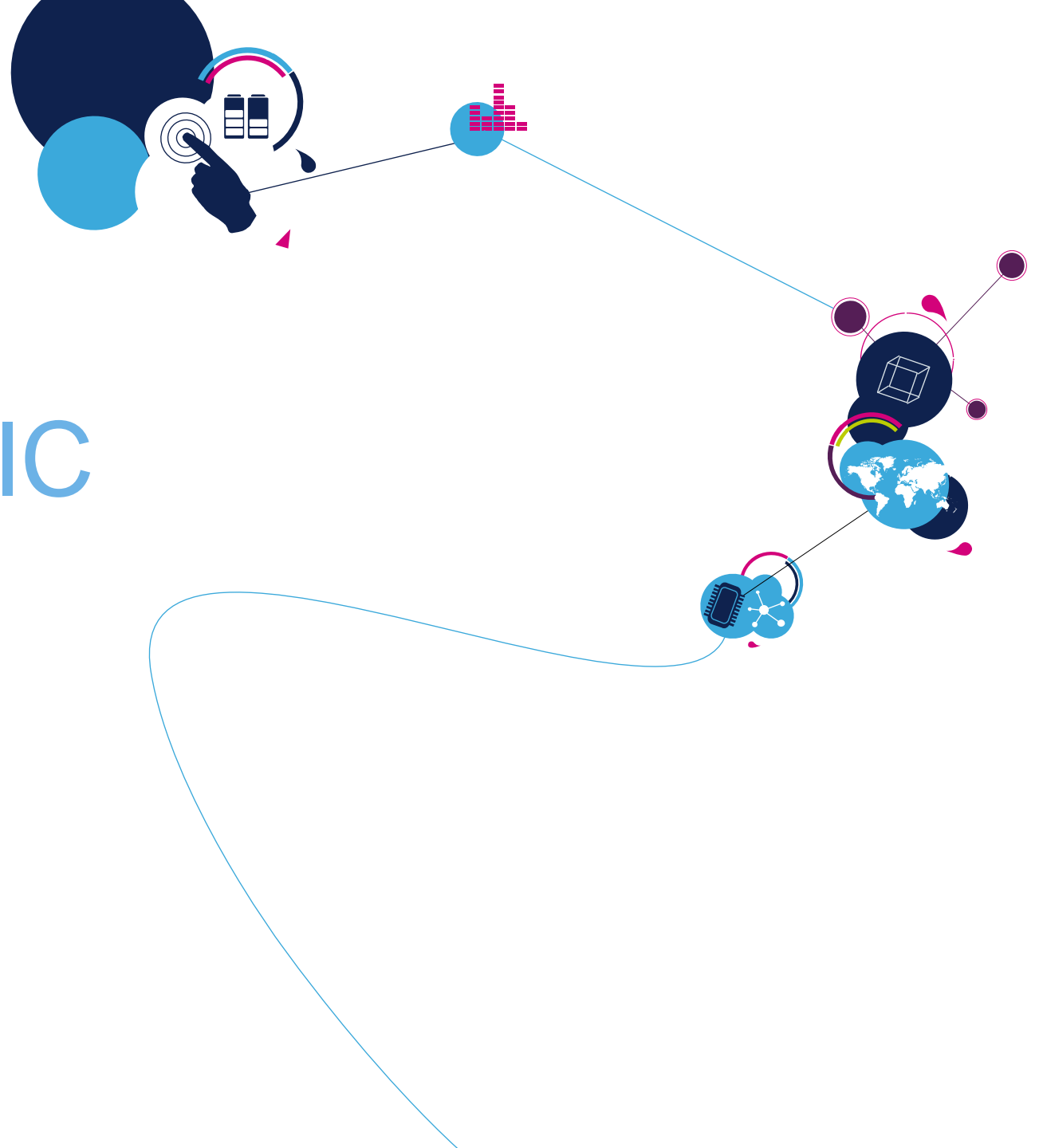
 /STM32

 @ST_World

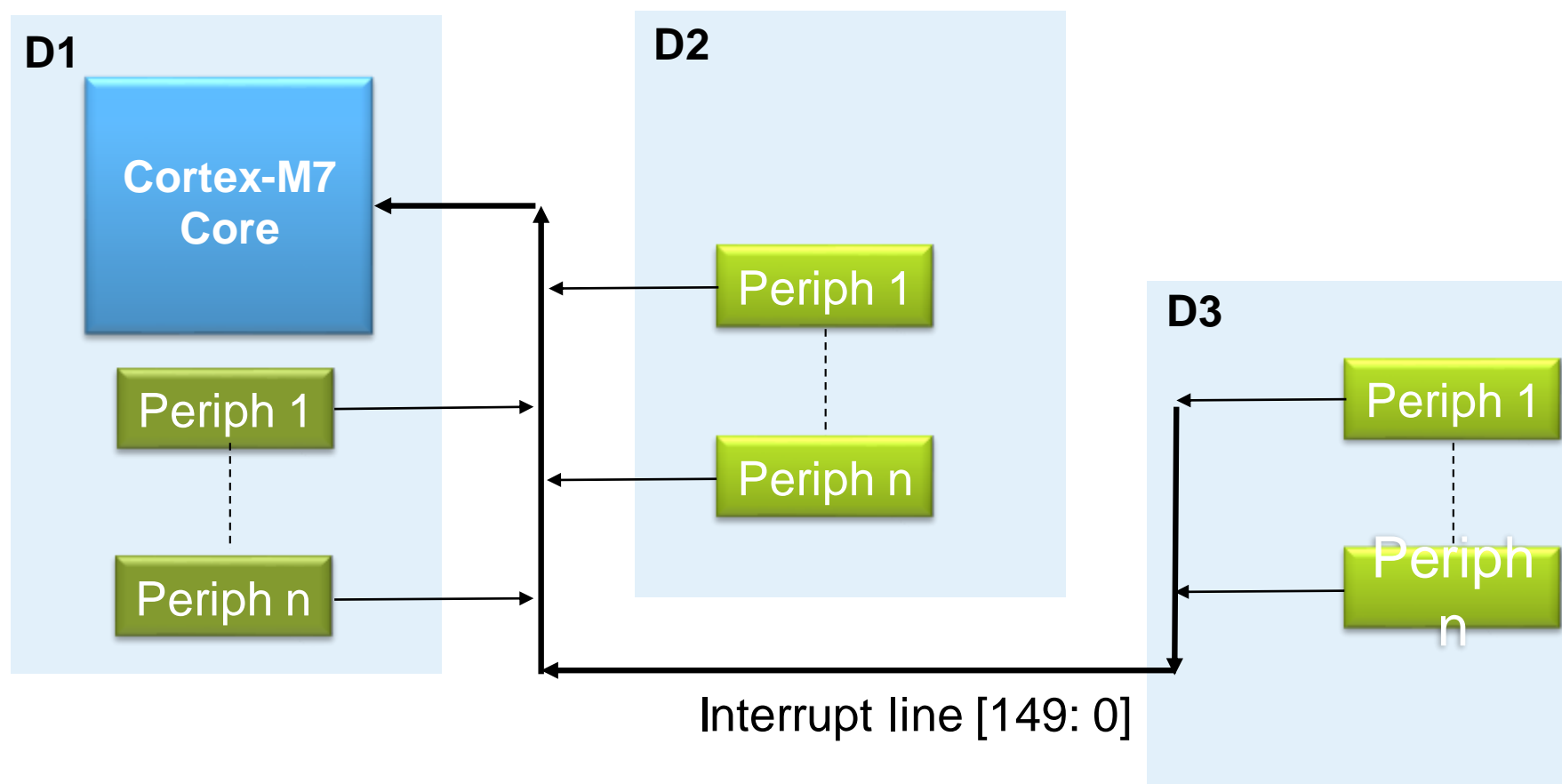
 st.com/e2e

STM32H7- NVIC

嵌套矢量中断控制器

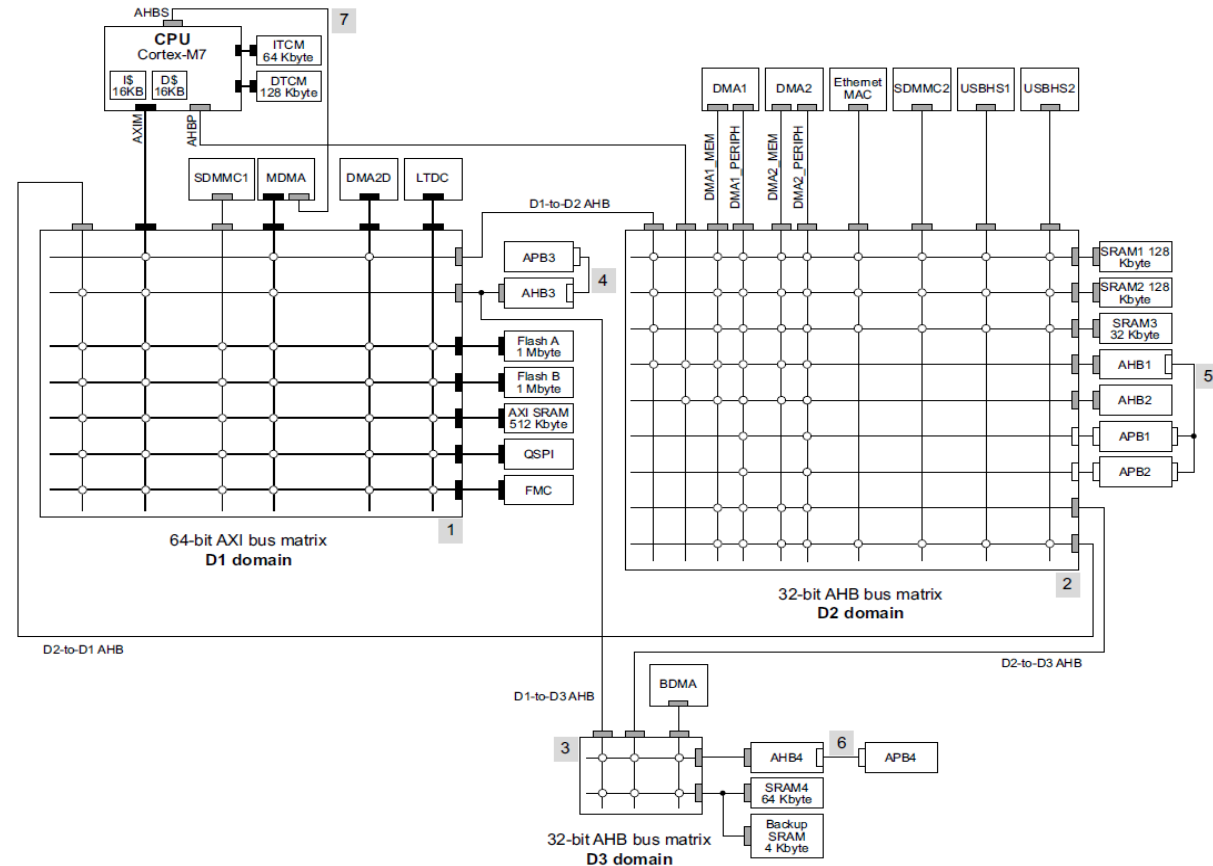
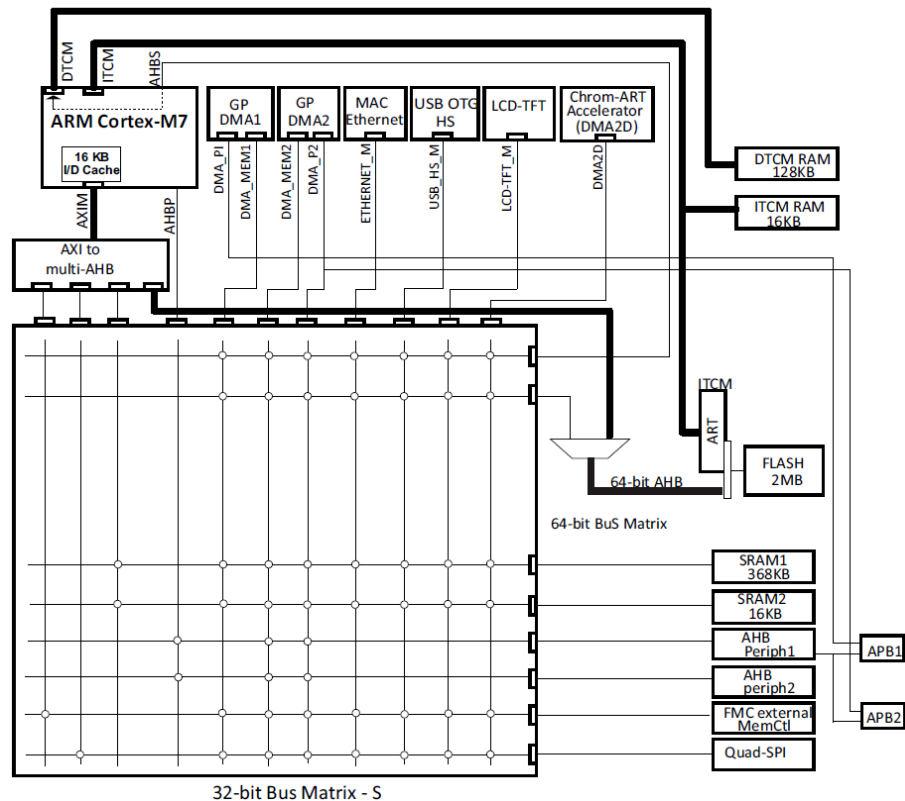


- 150 中断线
- 16 可编程优先级



系统结构 STM32F7 Vs STM32H7x3

137



1 Domain
STM32F7



life.augmented

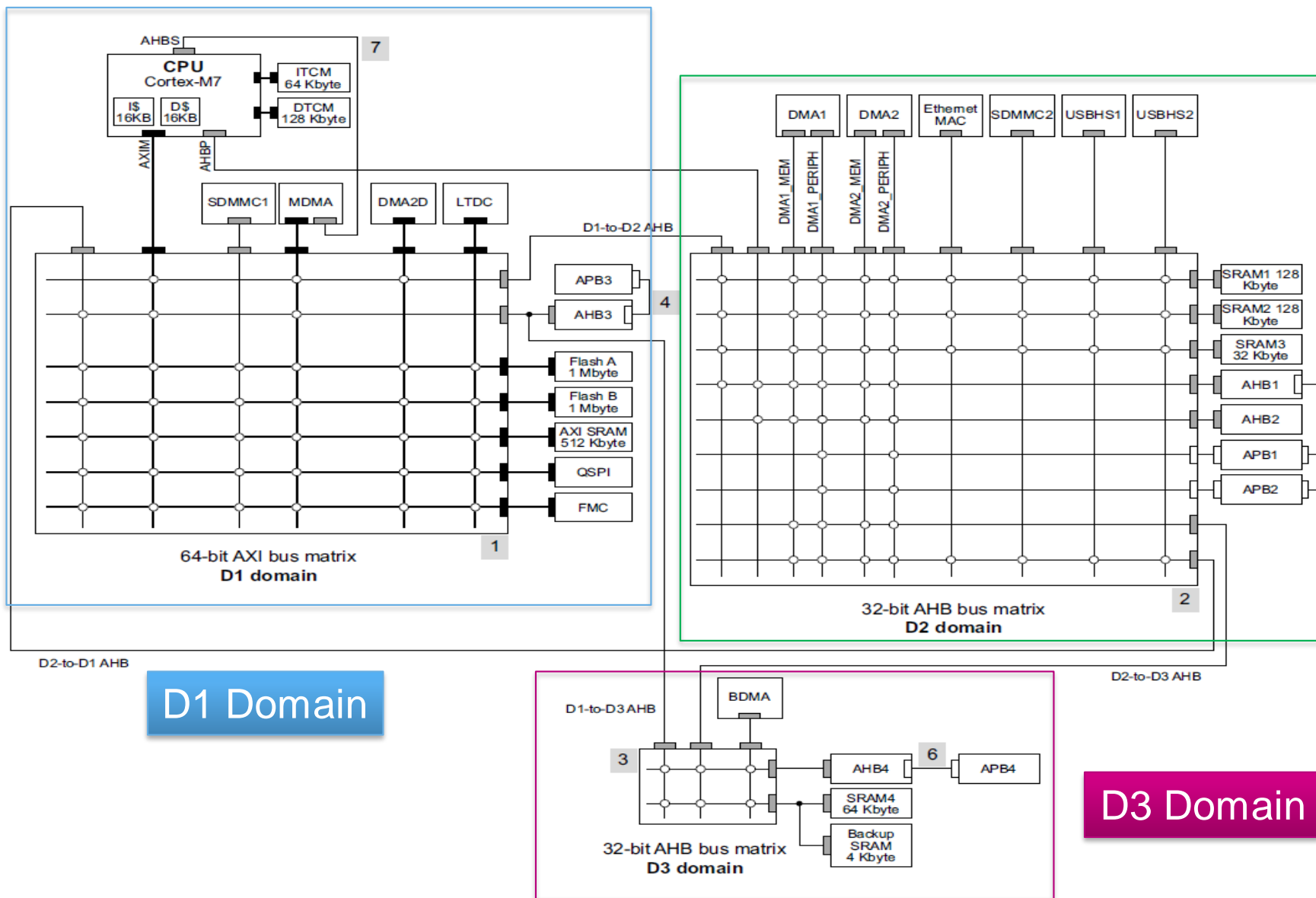
3 Domains
STM32H7



- 3 power domains to reduce overall leakage and propose efficient low power modes
- D2 AHB bus matrix clock gating when no resource is used in D2

系统结构 STM32F7 Vs STM32H7x3

138



D2 Domain

D1 Domain

D3 Domain

STM32F7 Vs STM32H7x3

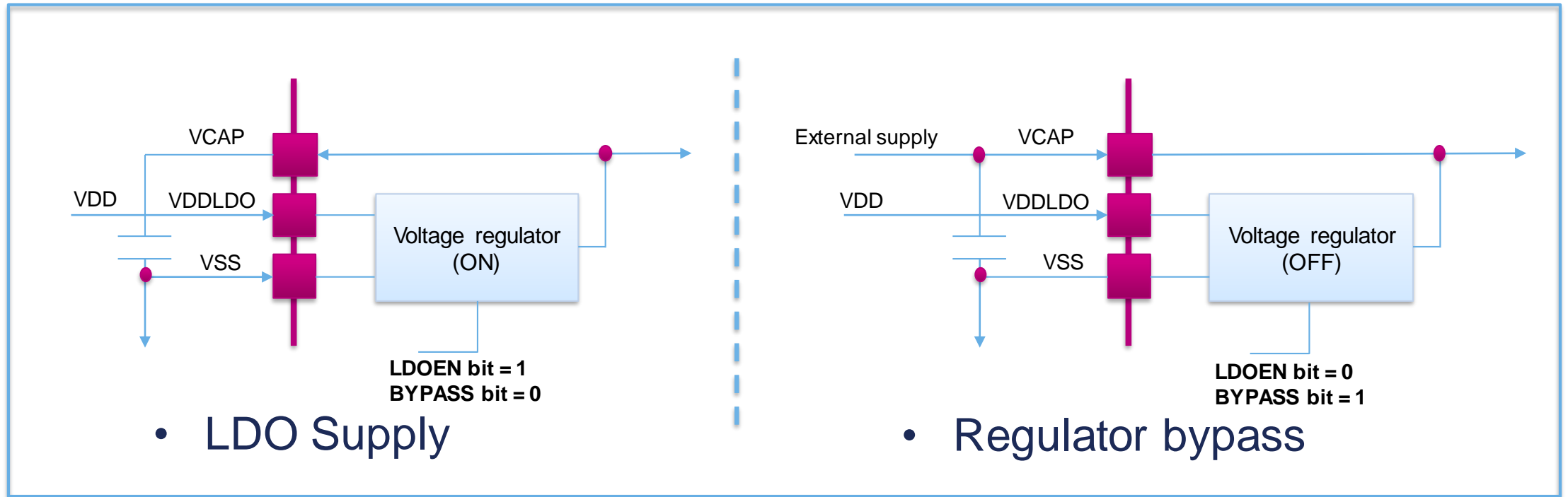
Scale	STM32F7 max Freq	STM32H7 Max freq			Unit
		Max CPU	Max D1	Max D2/D3	
Scale 1	216	400	200	200	MHz
Scale 2	180	300	150	150	
Scale 3	144	200	100	100	

Scale	STM32F7 max Vdd	STM32H7 Max Vdd	Unit
Scale 1	1.26 – 1.40	1.15 - 1.26	V
Scale 2	1.20 – 1.32	1.05 - 1.15	
Scale 3	1.08 – 1.20	0.95 - 1.05	

系统的电源配置 STM32H7x3

140

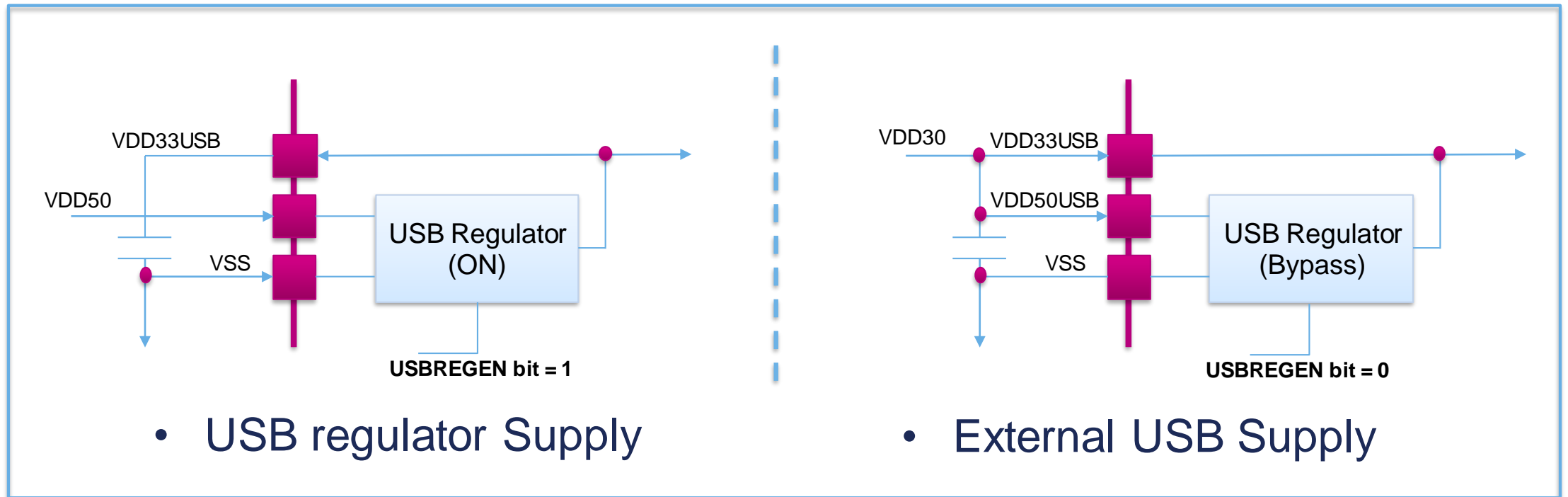
- 新增功能 STM32H7x3 : 稳压器旁路通过软件 LDOEN and BYPASS bits in PWR_CR3 Register
- 在 STM32F7无效



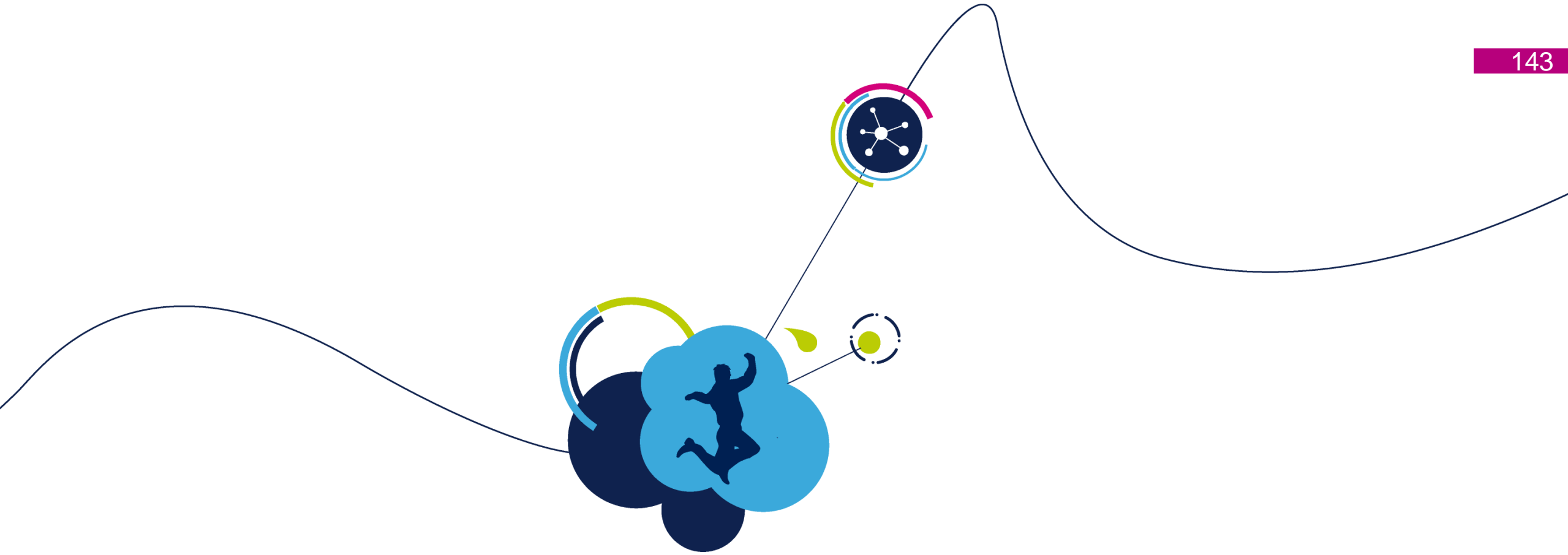
USB 电源配置 STM32H7x3

141

- 加入的新功能 STM32H7x3 : USB 稳压器
- 在 STM32F7无效



- STM32H7x3 供电监管中添加的新特性
 - 模拟电压检测(AVD)
 - 通过比较PWR_CR1 register ALS [1:0]位 选定的阈值，监测VDDA供电
 - AVD 启用是通过设置AVDEN bit in (PWR_CR1).
 - 电池电压阈值(VBAT thresholds)
 - 指示， VBAT 高于或低于阈值
 - VBAT电源监控可以启用/禁用通过 MONEN bit in (PWR_CR2).
 - 温度阈值
 - 温度监测可以启用/禁用通过 MONEN bit in (PWR_CR2).
 - 指示设备温度是否高于或低于阈值



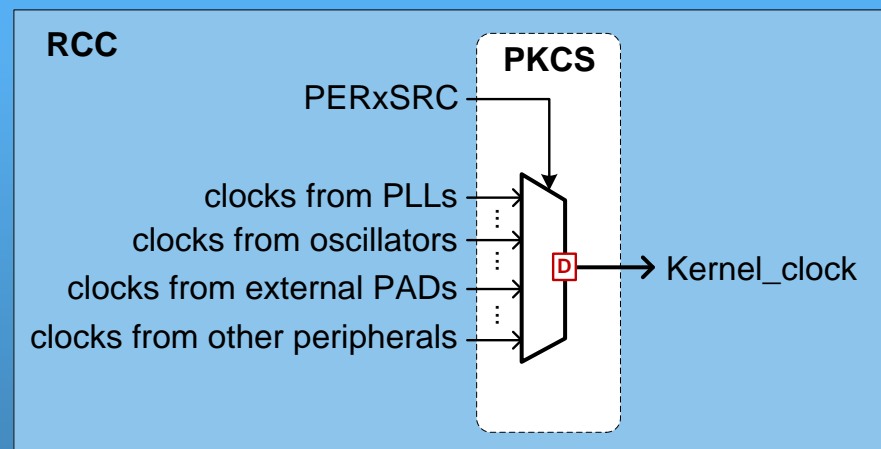
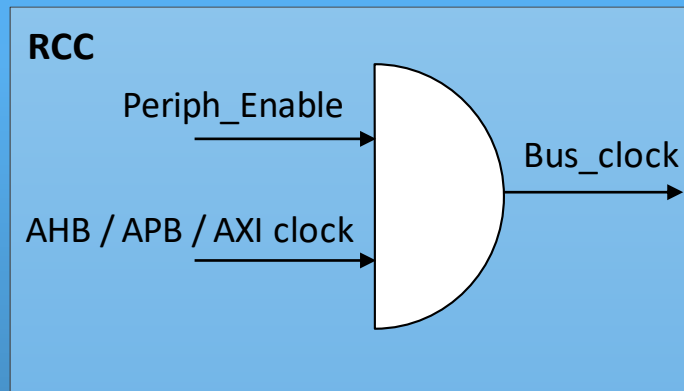
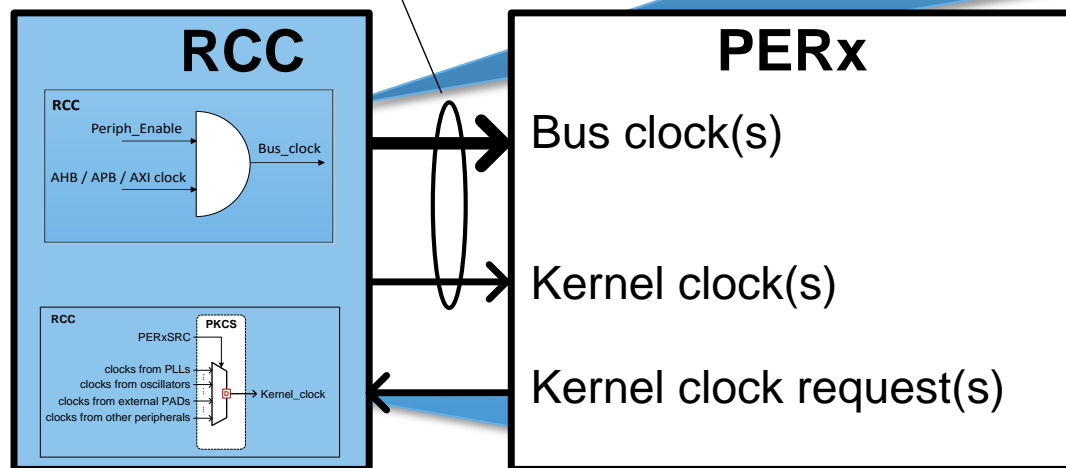
Reset and Clock Control (RCC)

STM32F7 Vs STM32H7x3

Source clock		STM32F7	STM32H7
Internal oscillators	HSI	16 MHz	64 MHz
	RC		48 MHz
	CSI	NA	4 MHz
	LSI	17~47kHz	32 kHz
External oscillators	HSE	4-26 MHz	1-48 MHz
	LSE	32.768 kHz	
PLLs		x3 without fractional mode	x3 With fractional mode (13-bit fractional multiplication factor)

- 外设普遍接受:
 - 一个或多个总线时钟
 - 一个或多个内核时钟

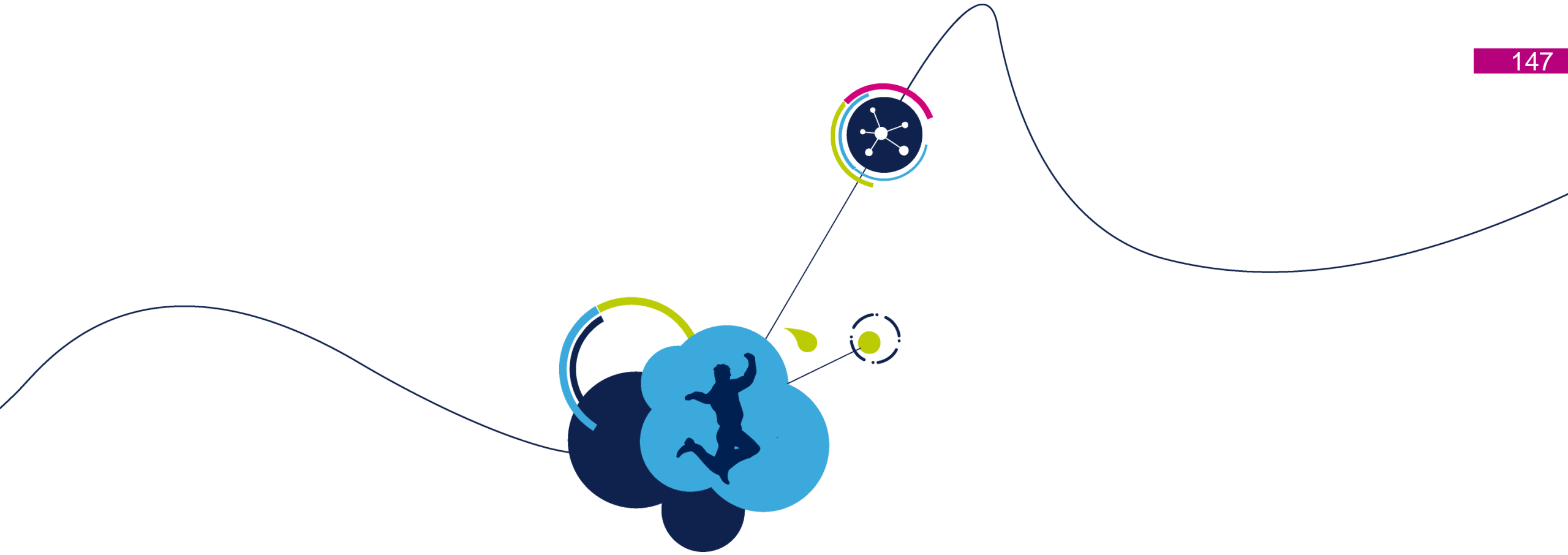
Peripheral clocks



外围时钟分布的例子

146

Peripheral	STM32F7	STM32H7x3
SPI1	APB2_Clock	Bus clock <ul style="list-style-type: none">• APB2_Clock Kernel clock <ul style="list-style-type: none">• Pll1_q_ck/Pll2_p_ck/Pll3_p_ck/I2S_CKIN/Per_ck
USART1	Bus clock <ul style="list-style-type: none">• APB2_Clock Kernel clock <ul style="list-style-type: none">• LSE• HSI• SYSCLK• PCLK2	Bus clock <ul style="list-style-type: none">• APB2_Clock Kernel clock <ul style="list-style-type: none">• Pll2_q_ck/pll3_q_ck/hsi_ker_ck/csi_ker_ck/lse_ck



EXTI

- 异步事件输入分类成两组:
 - 可配置的事件(I/Os 或 外设信号能够产生一个脉冲)
 - **直接事件**(从其他外设来的中断和唤醒源,需要在外设清除)

New

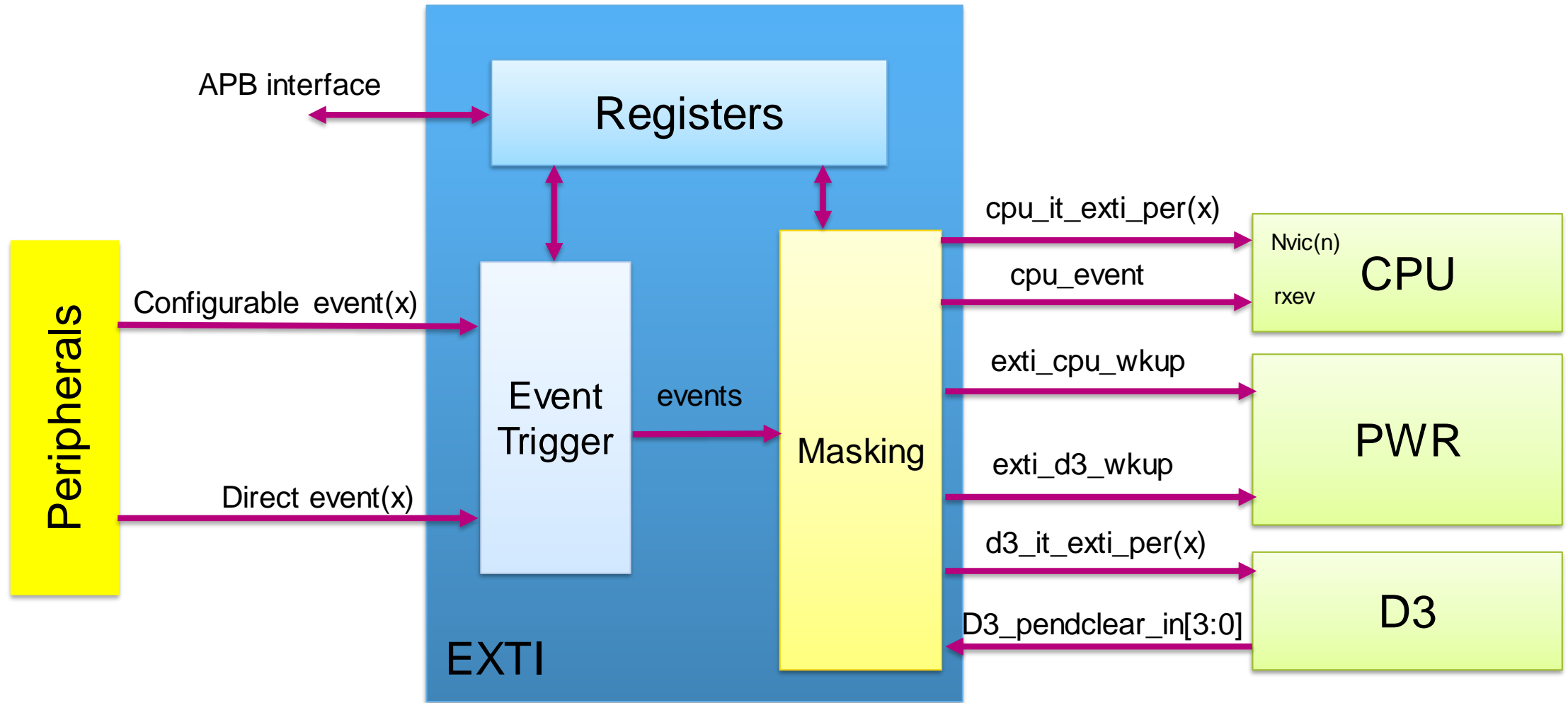
- 可配置的事件(从I/Os或外设来的信号能够产生一个脉冲),他们有在下列特性:
 - 可选择主动触发边缘
 - 中断挂起状态寄存器
 - 独立中断和事件产生掩码
 - 可能的软件触发
 - 可配置的系统D3域, 唤醒事件有D3悬挂掩码和状态寄存器和可能的D3中断信号 .

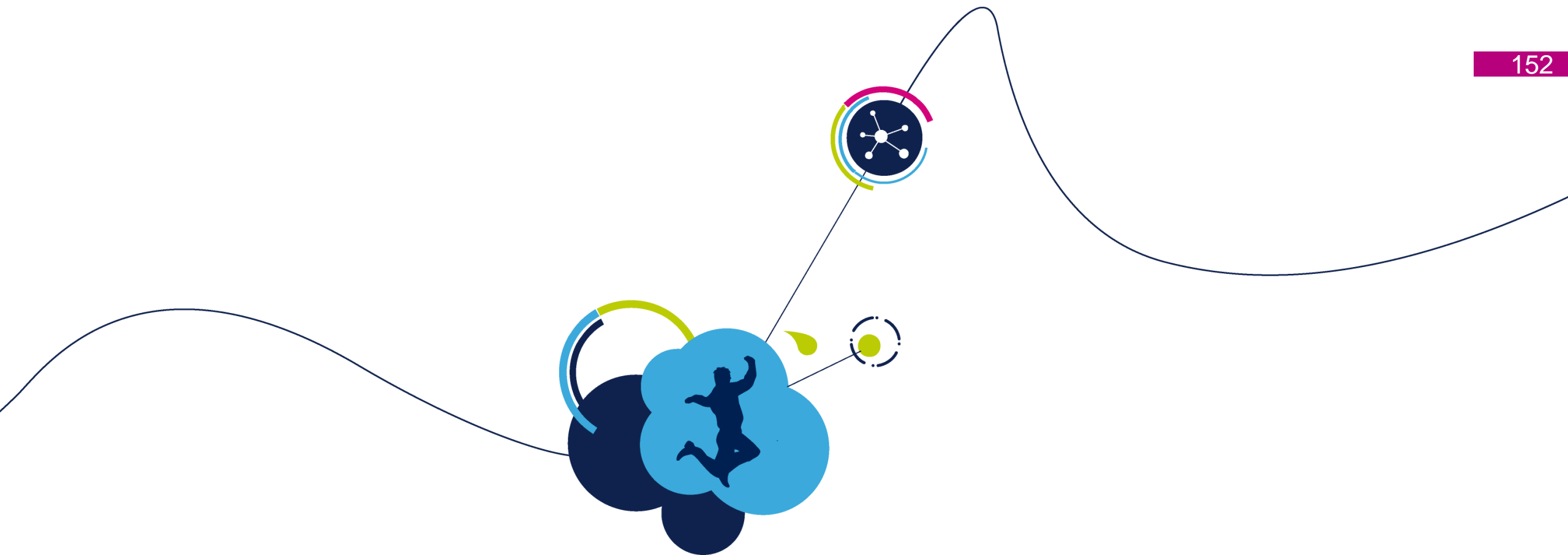
New

- **直接事件**(从其他外设来的中断和唤醒源,需要在外设清除)
 - 固定的上升沿主动触发
 - **EXTI** 没有中断挂起状态寄存器位 (中断挂起状态是由外设生成事件提供)
 - 独立中断和事件产生掩码
 - 没软件触发
 - 直接系统**D3**域唤醒事件, 有 **D3** 悬挂掩码和状态寄存器和可能 **D3** 中断信号

EXTI 框图 in STM32H7x3

151





存储器组织

RAM size: STM32F7 Vs STM32H7x3

Memory	STM32F7	STM32H7	Unit
ITCM-RAM	16	64	Kbytes
DTCM-RAM	128	128	
AXI-SRAM	-	512	
SRAM1	368	128	
SRAM2	16	128	
SRAM3	NA	32	
SRAM4	NA	64	
Backup SRAM	4	4	
Total*	512 Kbyte	1 Mbyte	

内存映射(1/2)

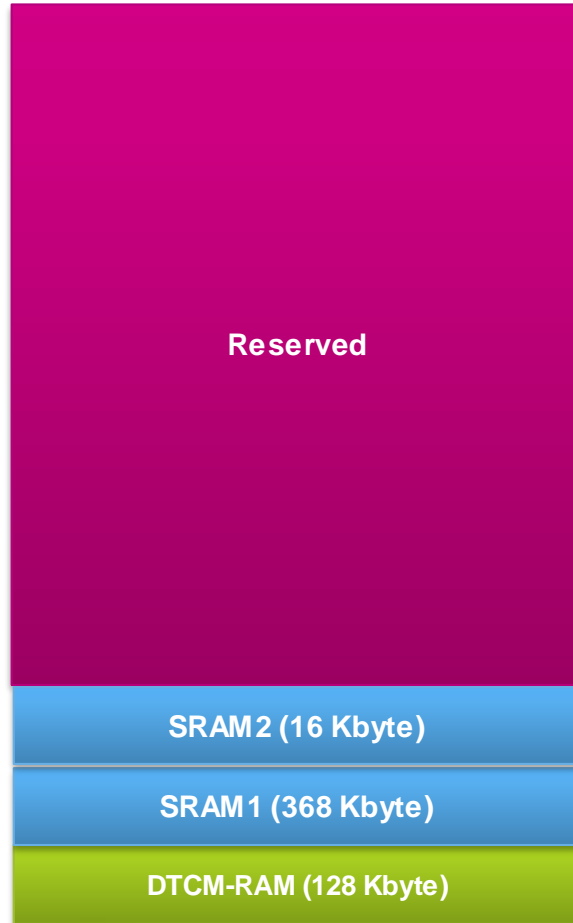
154

Memory	STM32F7	STM32H7	Compatibility
ITCM-RAM	0x0000 0000 – 0x0000 3FFF	0x0000 0000 – 0x0000 FFFF	✓
DTCM-RAM	0x2000 0000 – 0x2001 FFFF	0x2000 0000 – 0x2001 FFFF	✓
FLASH	Flash - AXI 0x0800 0000 – 0x081F FFFF	Flash A 0x0800 0000 – 0x080F FFFF Flash B 0x0810 0000 – 0x081F FFFF	✓
	Flash - ITCM 0x0020 0000 – 0x002F FFFF	NA	✗
System memory	0x1FF0 0000 – 0x1FF0 EDBF	Bank 1 0x1FF0 0000 – 0x1FF1 FFFF Bank 2 0x1FF4 0000 – 0x1FF5 FFFF	✗
	0x0010 0000 – 0x0010 EDBF	NA	✗

内存映射(2/2)

SRAM 组织

155



0x3801 0000 – 0x3FFF FFFF

0x3800 0000 – 0x3800 FFFF

0x3004 0000 – 0x3004 7FFF

0x3002 0000 – 0x3003 FFFF

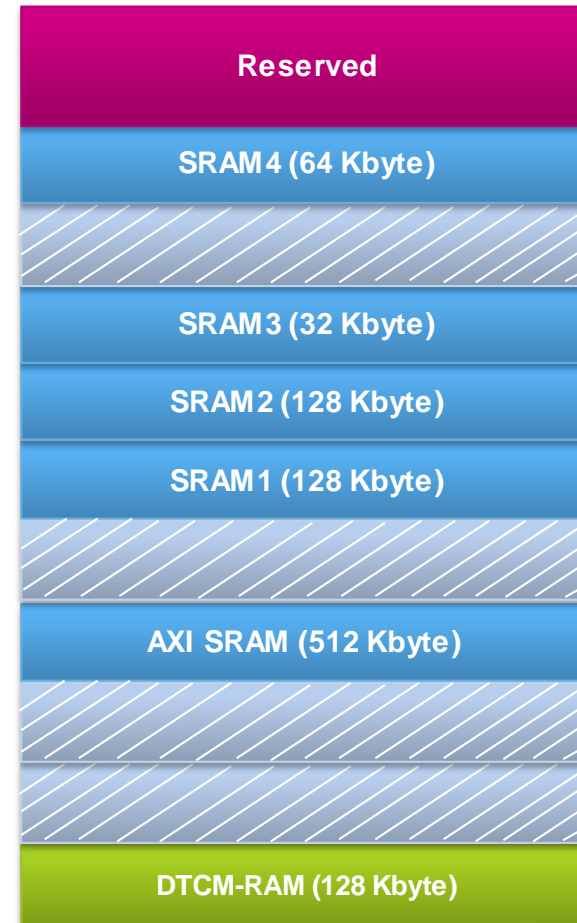
0x3000 0000 – 0x3001 FFFF

0x2400 0000 – 0x2407 FFFF

0x2007 C000 – 0x2007 FFFF

0x2002 0000 – 0x2007 BFFF

0x2000 0000 – 0x2001 FFFF



Reserved

SRAM4 (64 Kbyte)

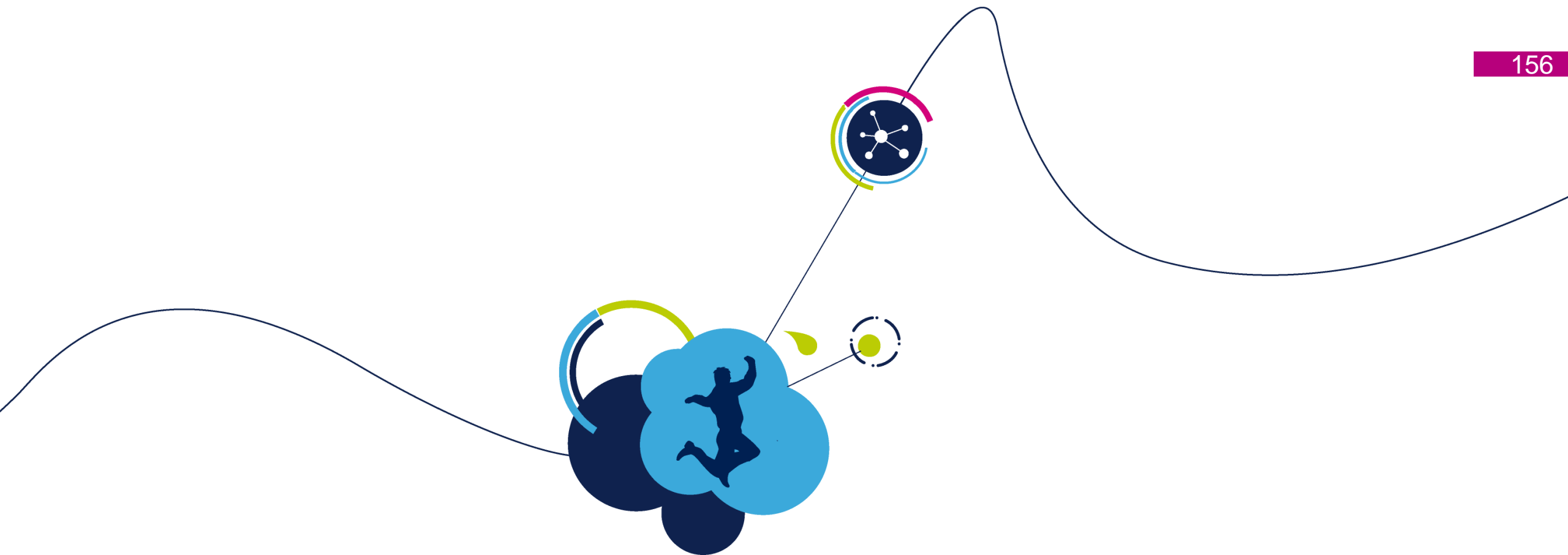
SRAM3 (32 Kbyte)

SRAM2 (128 Kbyte)

SRAM1 (128 Kbyte)

AXI SRAM (512 Kbyte)

DTCM-RAM (128 Kbyte)



嵌入式引导加载程序

嵌入式引导加载程序(1/2)

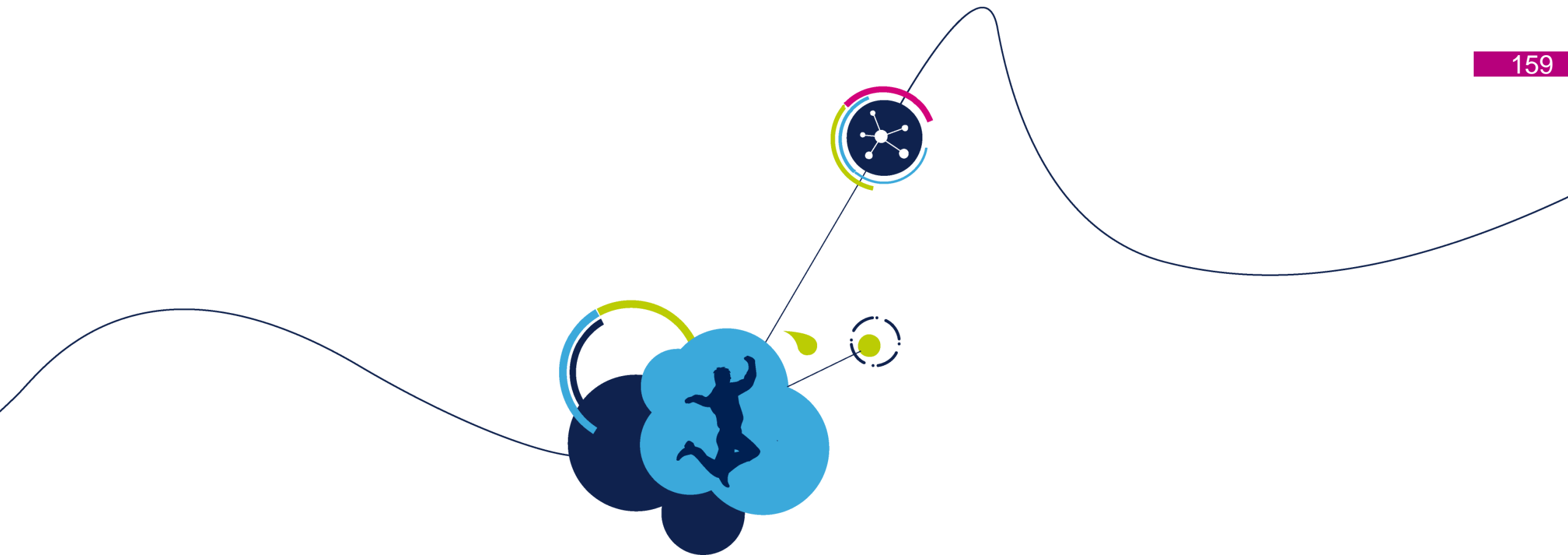
157

Boot loader peripherals	STM32F756xx	STM32H7x3
DFU	USB OTG FS (PA11 / PA12) in device mode	
USART1	PA9 / PA10	
USART2	NA	PA3 / PA2
USART3	PB10 / PB11	
	PC10 / PC11	NA
I2C1	PB6 / PB9	
I2C2	PF0 / PF1	
I2C3	PA8 / PC9	

嵌入式引导加载程序(2/2)

158

Boot loader peripherals	STM32F756xx	STM32H7x3
I2C4	NA	PD12 / PD13
SPI1		PA7/ PA6 / PA5 / PA4
SPI2		PI3 / PI2 / PI1 / PI0
SPI3		PC12 / PC11/ PC10 / PA15
SPI4		PE14 / PA13 / PA12 / PA11



系统配置控制器 (SYSCFG)

SYSCFG 主要特点 : STM32F7 Vs STM32H7x3

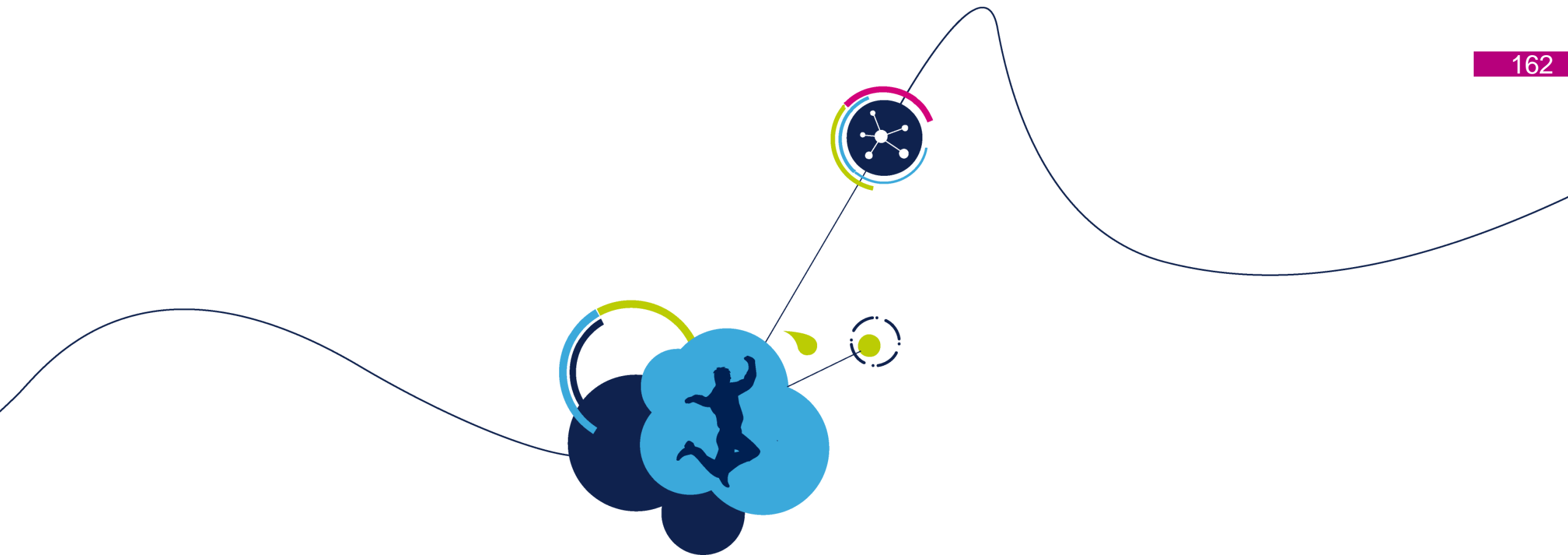
160

STM32F7	STM32H7x3
<ul style="list-style-type: none">• Remap the memory areas• Managing Class B feature(*)	NA
<ul style="list-style-type: none">• Select the Ethernet PHY interface• Managing the external interrupt line connection to the GPIOs• Managing I/O compensation cell feature• I2C Fast mode + configuration	

SYSCFG 主要特点 STM32F7 Vs STM32H7x3

161

- **New features added in STM32H7x3**
 - Analog switch configuration management
 - Getting readout protection and Flash memory bank swap informations
 - Management of boot sequences and boot addresses
 - Management BOR reset level
 - Management of Flash memory secured and protected sector status
 - Management Flash memory write protections status
 - Management of DTCM secured section status
 - Management of independent watchdog behavior (hardware or software / freeze)
 - Reset generation in Stop and Standby mode
 - Secure mode enabling/disabling.



外设汇总

STM32F7 Vs STM32H7

163

		STM32F7	STM32H7
Power supply		1.71 V to 3.6 V	<ul style="list-style-type: none"> • 1.62 V to 3.6 V • Voltage scaling in Run and Stop mode • Backup regulator (~0.9 V)
Maximum Frequency		216MHz	400MHz
Flash		2MB	
		Dual Bank	
SRAM	System	512KB	~1MB (992KB)
	Instruction	16KB	64KB
	Backup	4KB	
Common Peripherals	FMC	Yes	
	QSPI	Yes	
	Ethernet	Yes	

STM32F7 Vs STM32H7

164

		STM32F7	STM32H7
Timers	High resolution	NA	1
	General purpose	10	
	PWM	2	
	Basic	2	
	Low power	1	5
RNG		Yes	
Communication interfaces	SPI/I ² S	4/3 (simplex) 6/3(simplex)	6/3(simplex)
	I2C	4	
	USART/UART	4/4	
	LPUART	NA	1
	SAI	2	4
	SPDIFRX	4 inputs	

STM32F7 Vs STM32H7

165

		STM32F7	STM32H7
Communication interfaces	SWPMI	NA	Yes
	MDIO	Yes	
	SDMMC	Yes	
	CAN	3 x CANs (2.0B Active)	2x CAN FD (FDCAN1 supports TTCAN)
	USB OTG FS	Yes	
	USB OTG HS	Yes	
	HDMI-CEC	Yes	

STM32F7 Vs STM32H7

166

		STM32F7	STM32H7
Digital Camera interface		Yes	
MIPI-DSI Host		Yes	No
Graphics	LCD-TFT	Yes	
	Chrom-ART Accelerator™ (DMA2D)	Yes	
	JPEG Codec	Yes	
GPIOs		Up to 168 I/O ports with interrupt capability	
		<ul style="list-style-type: none"> – Up to 164 fast I/Os up to 108 MHz – Up to 166 5 V-tolerant I/Os 	<ul style="list-style-type: none"> – Up to 4 fast I/Os up to 166 MHz – Up to 89 I/Os up to 83 MHz – Up to 164 5 V-tolerant I/Os

STM32F7 Vs STM32H7

167

		STM32F7	STM32H7
Analog peripherals	ADC	X3 (12-bit) Number of channels: up to 24	X3 (8 to 16-bit) Number of channels: 20
	12-bit DAC	Yes	
	Operational amplifiers	NA	2
	ULTRA Low power Comparator	NA	2
	DFSDM	Yes	

STM32F7 Vs STM32H7

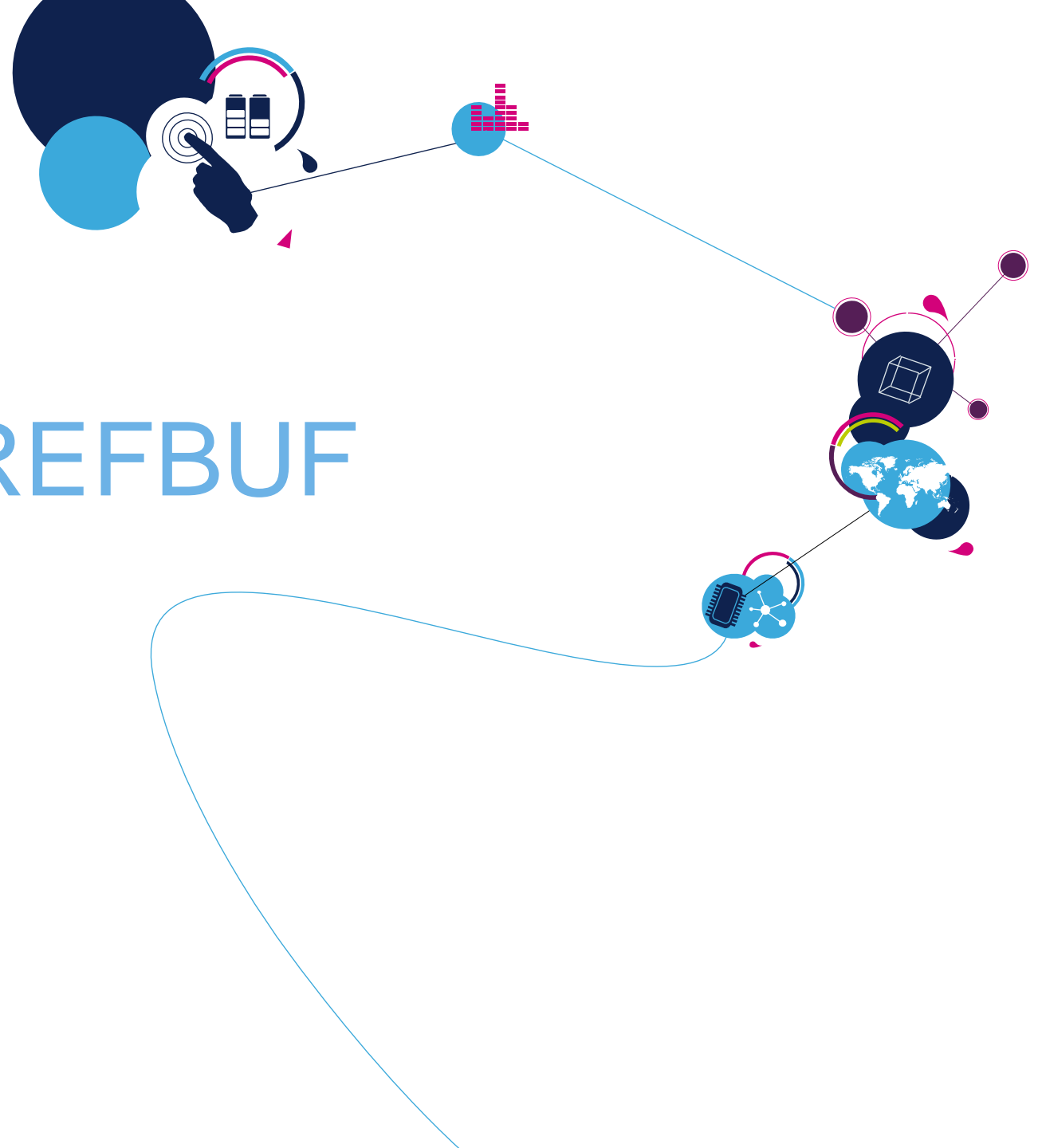
168

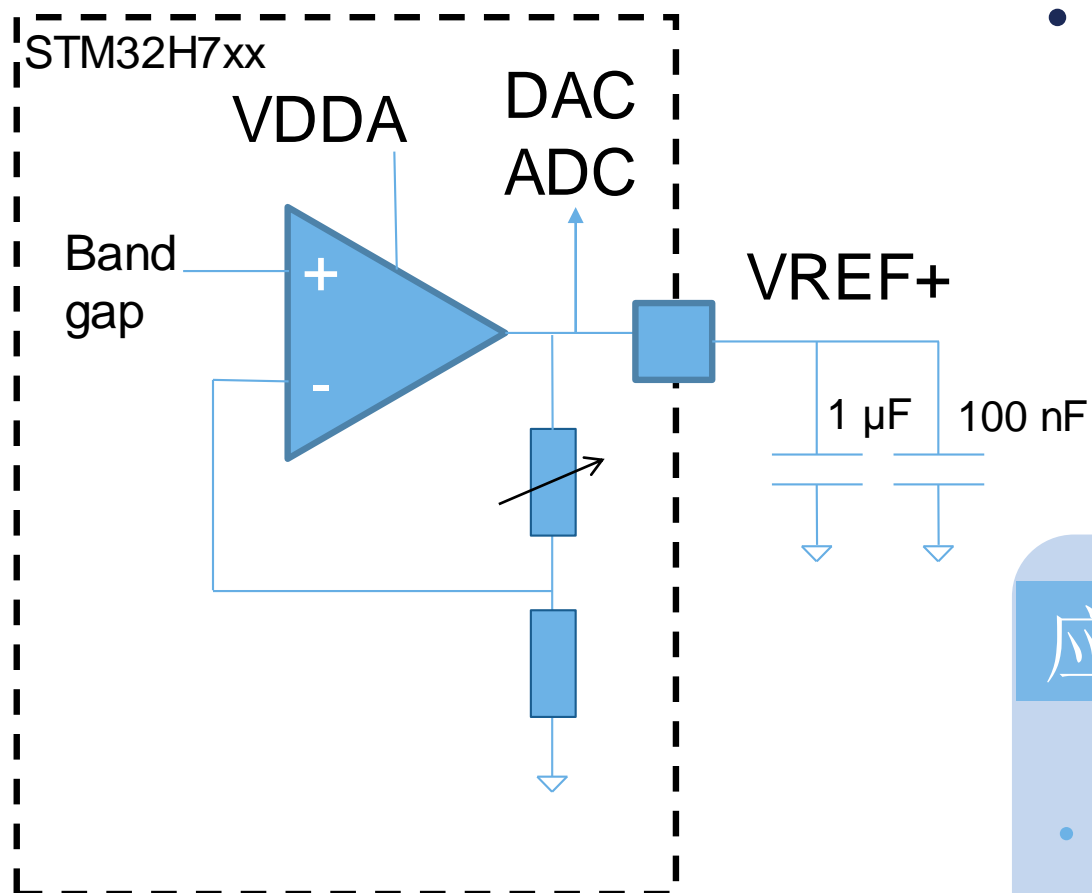
	STM32F7	STM32H7
DMA	General-purpose DMA: 16-stream DMA controller with FIFOs and burst support	4 DMA controllers to unload the CPU <ul style="list-style-type: none">• 1x high-speed general-purpose master direct memory access controller (MDMA)• 2x dual-port DMAs with FIFO and request router capabilities for optimal peripheral management• 1 basic DMA with request router capabilities
Cryptographic acceleration	<ul style="list-style-type: none">• AES 128, 192, 256, TDES,• HASH (MD5, SHA-1, SHA-2), HMAC• 2x true random number generator (3 oscillators each)	
Security	<ul style="list-style-type: none">• ROP, active tamper	<ul style="list-style-type: none">• ROP, PC-ROP, active tamper

STM32H7 - VREFBUF

参考电压缓冲器

版本 1





- 提供了一个模拟参考电压
 - 2.5 / 2.048 / 1.8 / 1.5 V 参考电压 for ADC/DAC
 - 可以提供参考电压，和用低静态电流支持外部负载达4 mA
 - 定量模式

应用好处

- 不需要外部参考电压电路
- 片上VREF发生器提供VDDA-独立的参考电压

Mode	Description
Run	Active.
Stop	Active.
Standby	Standby mode. The peripheral must be reinitialized after exiting standby mode.

Symbol	Condition	Typical	Unit
V_{DDA}	$V_{REF} = 1.5$	1.8~3.6	V
	$V_{REF} = 1.8$	2.1~3.6	V
	$V_{REF} = 2.048$	2.4~3.6	V
	$V_{REF} = 2.5$	2.8~3.6	V
I_{load}	Max. load current	4	mA
I_{VDDA}	$I_{LOAD} = 0 \mu A$	15	μA
	$I_{LOAD} = 500 \mu A$	16	μA
	$I_{LOAD} = 4 \text{ mA}$	32	μA
PSRR	DC	60	dB
t_{start_up}	$C_{LOAD} = 1.1 \mu F$	500	μs

- 更多信息请参考这些与这个外设相关的培训
 - Analog-to-digital converter (ADC)
 - Digital-to-analog converter (DAC)