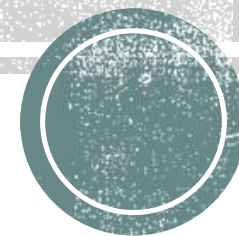


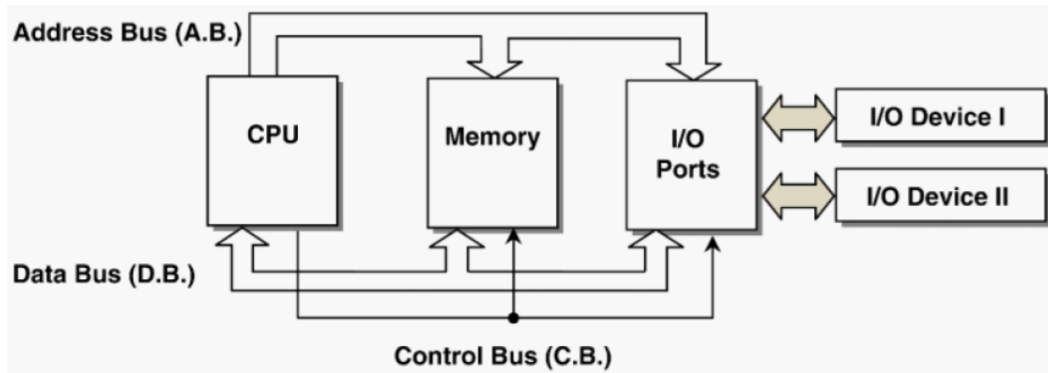
# فصل اول : آشنایی با ساختار میکروکنترلرها



# معماری کلی میکروکنترلرها

## ■ سه بخش اصلی در یک میکروکنترلر

- واحد پردازش مرکزی (CPU)
- اجرای دستورات برنامه و ارسال سیگنال‌های کنترلی لازم به قطعات درونی میکروکنترلر
- حافظه
- ذخیره اطلاعات کاربر شامل دستورات و اطلاعات
- درگاه‌های ورودی و خروجی (I/O Ports)
- واسط ارتباطی بین واحد پردازش مرکزی میکروکنترلر و قطعات جانبی درونی و بیرونی

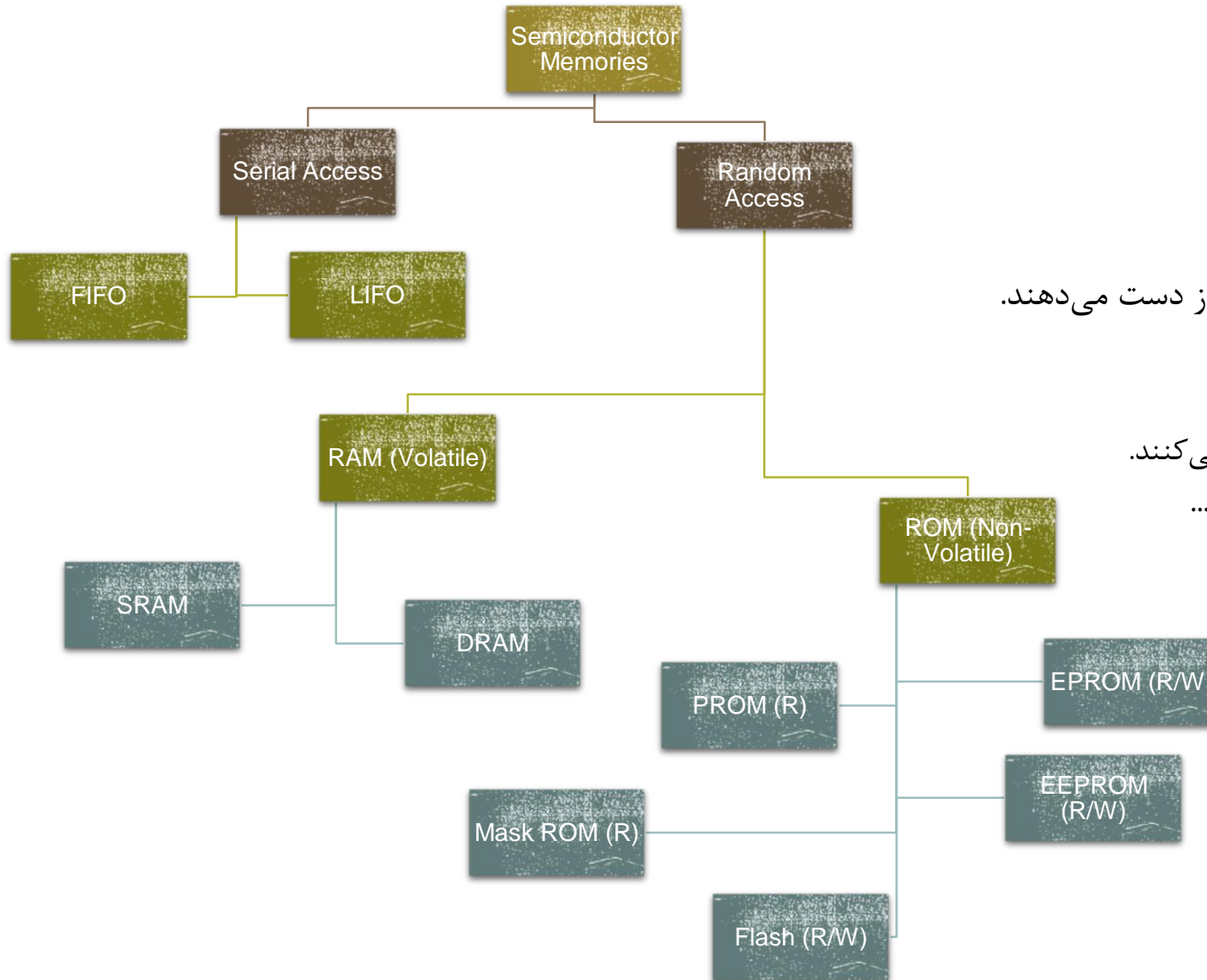


## ■ اتصال اجزای اصلی از طریق گذرگاه (BUS)

- گذرگاه آدرس (Address Bus)
- گذرگاه داده (Data Bus)
- گذرگاه کنترل (Control Bus)



# انواع حافظه



## ■ حافظه از دیدگاه پایداری اطلاعات

### ■ حافظه‌های Volatile

- این حافظه‌ها پس از قطع منبع تغذیه، اطلاعات خود را از دست می‌دهند.
- حافظه‌های RAM، Cache، FIFO، LIFO و ...

### ■ حافظه‌های Non-Volatile

- این حافظه‌ها پس از قطع منبع تغذیه، اطلاعات خود را حفظ می‌کنند.
- انواع حافظه‌های خانواده ROM، HDD، SSD و ...

## ■ حافظه از دیدگاه نحوه دسترسی

### ■ دسترسی سریال (Serial Access)

#### ■ LIFO، FIFO

- کاربرد؟!

### ■ دسترسی تصادفی (Random Access)

#### ■ Cache، ROM، RAM



# انواع حافظه‌های ROM



## ■ حافظه Mask ROM (Read Only Memory)

برنامه ریزی شده توسط کارخانه سازنده و غیرقابل تغییر

## ■ حافظه PROM

قابل برنامه ریزی فقط برای یکبار توسط تولید کننده محصولات دیجیتال

## ■ حافظه EPROM

دارای قابلیت پاک کردن اطلاعات با تاباندن نور فرابنفش



## ■ حافظه EEPROM (Electrically Erasable Programmable Read-Only Memory)

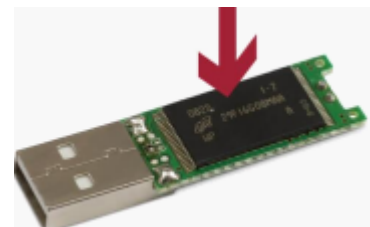
دارای قابلیت پاک کردن و برنامه‌ریزی مجدد با استفاده از سیگنال‌های الکتریکی



## ■ حافظه Flash

نسل جدید حافظه‌های EEPROM با ظرفیت ذخیره سازی متوسط

NOR Flash ،NAND Flash



# انواع حافظه‌های RAM

## Static RAM (SRAM) ■

تشکیل شده از فلیپ-فلاپ‌ها؟!

توان مصرفی زیاد و پایداری اطلاعات ذخیره شده در وجود منبع تغذیه

## Dynamic RAM (DRAM) ■

تشکیل شده از خازن‌ها

توان مصرفی کم و ناپایداری اطلاعات ذخیره شده

نیاز به Refresh کردن محتوای ذخیره شده در خازن‌ها

مفاهیم ترانزیستور و خازن؟!  
فیدبک مثبت و منفی؟!  
پایداری؟!

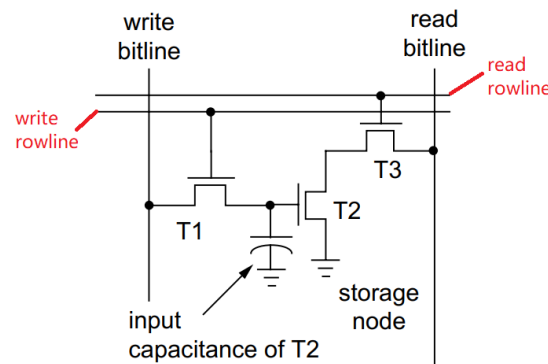
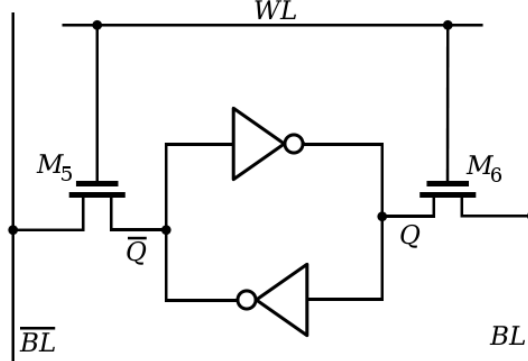
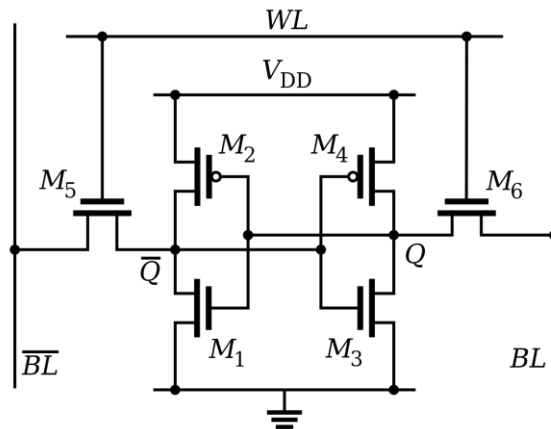
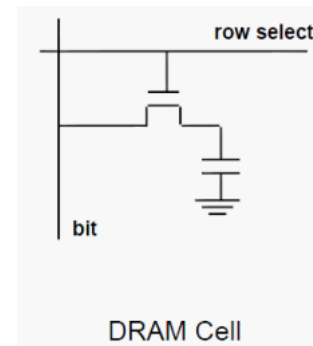


FIGURE 8.3: 3T1C DRAM cell.



# مقایسه حافظه‌های RAM و ROM

	RAM	ROM
<b>Data-Retention</b>	RAM is a volatile memory that could store the data as long as the power is supplied.	ROM is a non-volatile memory that the could retain the data even when the power is turned off.
<b>Speed</b>	It is a high-speed memory.	It is much slower than the RAM.
<b>CPU Interaction</b>	CPU can easily access data stored in RAM.	CPU cannot easily access data stored in ROM.
<b>Cost</b>	RAM is more costlier than ROM.	ROM is cheaper than RAM.
<b>Use</b>	Used to store the data that has to be currently processed by CPU temporarily.	It is typically used to store firmware or microcode, which is used to initialize and control hardware components of the computer.
<b>Function</b>	Used for the temporary storage of data currently being processed by the CPU.	Used to store firmware, BIOS, and other data that needs to be retained.



# حافظه‌های مورد نیاز در یک Embedded System

- حافظه با قابلیت ذخیره دیتا به صورت پایدار (Non-Volatile)

- ذخیره برنامه
- ذخیره پارامترها و متغیرهای ثابت
- ذخیره پارامترهای کاربر (حافظه محدود)

- حافظه با سرعت پردازش بالا به منظور اجرای سریع دستورات

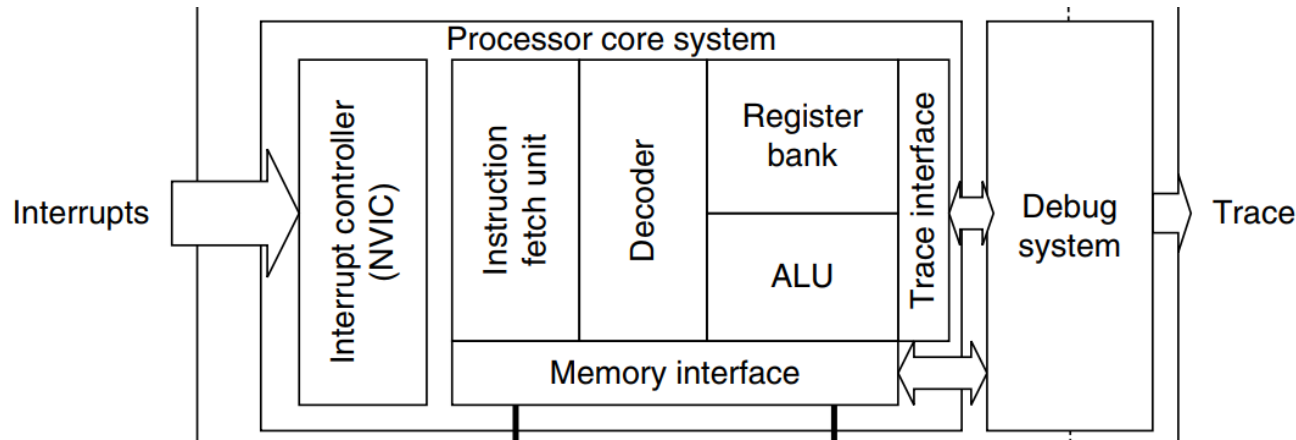
- دستورات اجرای برنامه
- متغیرهای ایجاد شده به واسطه برنامه

- حافظه Cache!?

- کاربرد!?



# Processor Core System



## ■ شماتیک پردازنده نسل Cortex M3

- پردازنده مبتنی بر معماری Harvard
- 32-bit microprocessor
- 32 bit data path
- 32 bit register bank
- 32 bit memory interface
- بیشترین حافظه قابل درایو توسط باس ۳۲ بیت معادل با 4GB می باشد.
- پشتیبانی از فرمت های Big endian و Little endian
- ابزارهای خطایابی
- Break point
- Watch point
- Instruction Trace
- بانک ثبات (Register) ۱۵ ثباتی
- واحد کنترل وقفه (Nested Vectored Interrupt Controller)
- Arithmetic Logic Unit (ALU)
- Memory Interface





# ثبات‌های داخلی پردازنده

Name	Functions (and banked registers)
R0	General-purpose register
R1	General-purpose register
R2	General-purpose register
R3	General-purpose register
R4	General-purpose register
R5	General-purpose register
R6	General-purpose register
R7	General-purpose register
R8	General-purpose register
R9	General-purpose register
R10	General-purpose register
R11	General-purpose register
R12	General-purpose register
R13 (MSP)	Main Stack Pointer (MSP), Process Stack Pointer (PSP)
R13 (PSP)	
R14	Link Register (LR)
R15	Program Counter (PC)

Low registers

High registers

پردازنده‌های Cortex M3 شامل 16 ثبات داخلی می‌باشند (ثبات R0 تا R15).

۱۳ ثبات عمومی شامل R0 تا R12

۱ ثبات پشته (Stack Register) به نام R13

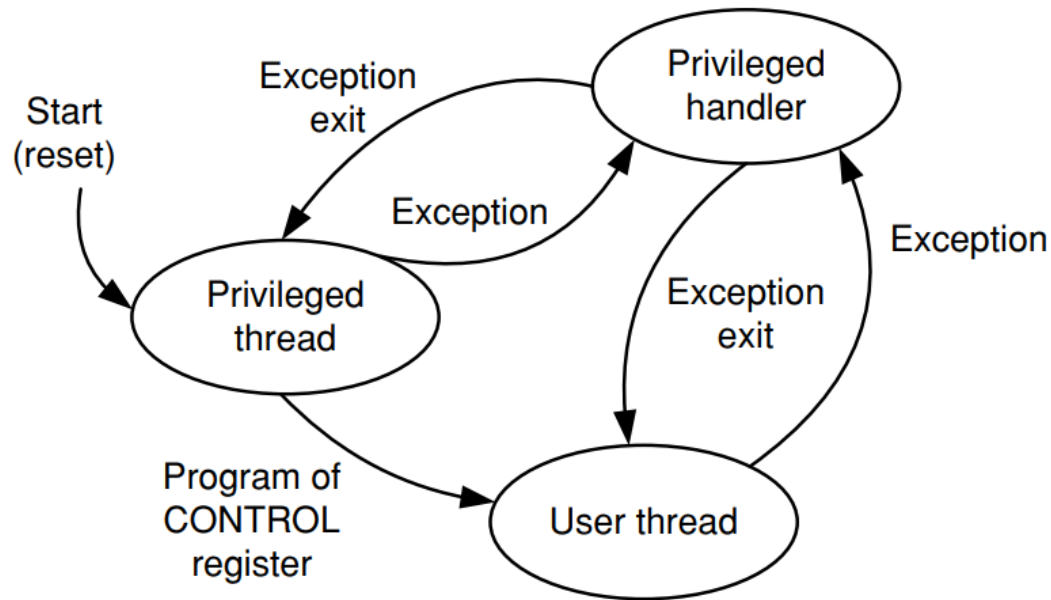
PSP, MSP

۱ ثبات ارتباط (Link Register) به نام R14

۱ ثبات شمارنده برنامه (Program Counter (PC)) به نام R15



# حالات کارکرد پردازنده Cortex M3



پردازنده‌های Cortex M3 دارای دو حالت کاری اصلی است

▪ Thread Mode

▪ Handler Mode

پردازنده دارای دو سطح دسترسی نیز می‌باشد.

▪ سطح ممتاز (Privileged Level)

▪ سطح کاربر (User Level)

▪ سطح دسترسی یک مکانیزم ابتدایی برای محافظت از بخش‌های بحرانی حافظه می‌باشد.

▪ در سطح ممتاز، برنامه به تمام نقاط حافظه دسترسی دارد (بجز مناطق محدود شده توسط MPU)

▪ با ترکیب حالات فوق ؟

▪ نقطه شروع برنامه!؟

▪ برنامه در حالت thread می‌تواند در سطح دسترسی ممتاز یا کاربر باشد اما برنامه در حالت handler بایستی در سطح ممتاز باشد.

▪ انتقال از سطح ممتاز به سطح کاربر به کمک رجیستر Control! قابل انجام است



# Memory Protection Unit (MPU)

- در پردازنده‌های Cortex M3 این واحد اجباری نیست!
- وضع قوانین برای دسترسی به بخش‌هایی از حافظه
- محدود کردن دسترسی به بخش‌هایی از حافظه
  - در صورت درخواست دسترسی کاربر در برنامه !؟
- بطور کلی MPU در پردازنده‌های دارای سیستم عامل وجود دارد که توسط سیستم عامل نیز استفاده می‌شود تا از بخش‌های مهم حافظه محافظت نماید.
- محدود کردن حافظه در سطح دسترسی ممتاز در برابر برنامه‌های غیر مطمئن
- قابلیت تغییر بخشی از حافظه به فقط خواندنی (Read-Only) به منظور عدم پاک شدن اطلاعات و یا دستکاری در برنامه‌های Multi-Tasking



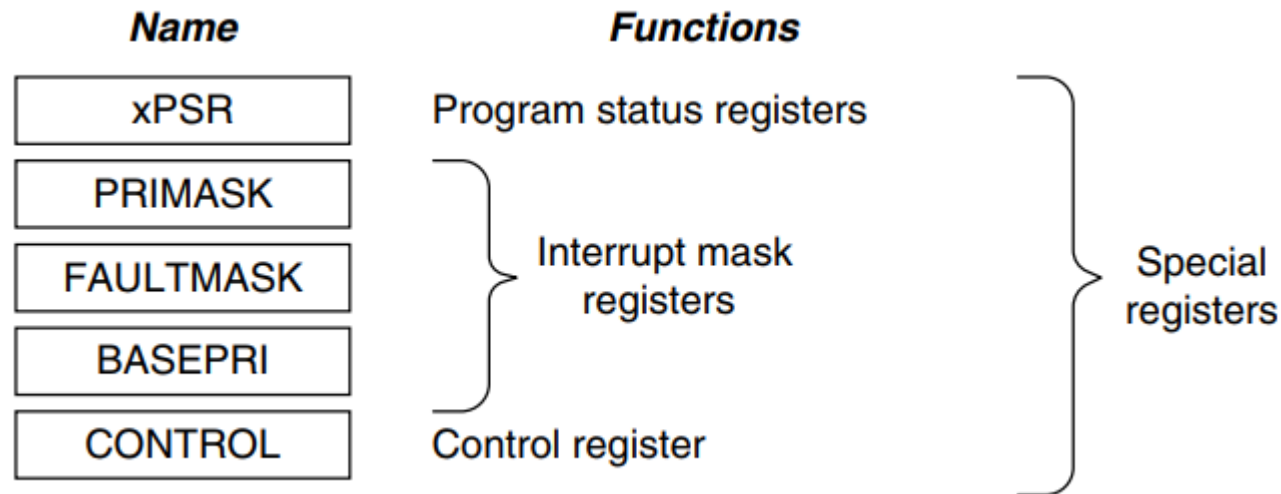
# Nested Vectored Interrupt Controller (NVIC)

- واحد کنترل وقفه یکی از مهمترین قابلیت‌های هر پردازنده‌ای می‌باشد.
- وقفه این قابلیت را به کاربر می‌دهد تا در ازای پیش‌آمد حالات خاص، دستورات اصلی برنامه متوقف شود و ابتدا دستورات تعریف شده در وقفه انجام پذیرد سپس ادامه برنامه اصلی اجرا گردد.
- قابلیت اجرای وقفه‌های تو در تو
- قابلیت اجرای برنامه با افزودن دستورات اجرای وقفه به برنامه
- قابلیت اولویت بندی در پذیرش وقفه
- **Interrupt Masking!?**



# رجیسترهای اختصاصی (Special Registers)

- تعدادی رجیستر اختصاصی در پردازنده‌های Cortex M3 وجود دارد. این رجیسترها دارای وظایف خاصی هستند.
- روش دسترسی به این رجیسترها با استفاده از دستورات اختصاصی است (استثناها یا وقفه‌های سخت‌افزاری)



# Memory Map

0xFFFFFFFF	System level	Private peripherals including build-in interrupt controller (NVIC), MPU control registers, and debug components
0xE0000000 0xDFFFFFFF	External device	Mainly used as external peripherals
0xA0000000 0x9FFFFFFF	External RAM	Mainly used as external memory
0x60000000 0x5FFFFFFF 0x40000000	Peripherals	Mainly used as peripherals
0x3FFFFFFF 0x20000000	SRAM	Mainly used as static RAM
0x1FFFFFFF 0x00000000	CODE	Mainly used for program code. Also provides exception vector table after power up

4GB = 8 \* 512MB ■

حافظه Flash؟! ■

از دید پردازنده همه‌ی المان‌های داخلی یک حافظه هستند. بنابراین بایستی آدرسی به آن‌ها اختصاص دهد.



# Bus Interfaces

- معمولاً در پردازنده‌های Cortex M3 باس‌های

- Code memory buses

- I-Code

- D-Code

- این باس‌ها به منظور دریافت دستورات و دیتاها به صورت بهینه تعبیه شده‌اند تا راندمان اجرای برنامه را بهبود بخشند.

- System bus

- دسترسی به حافظه SRAM و سطوحی از System memory

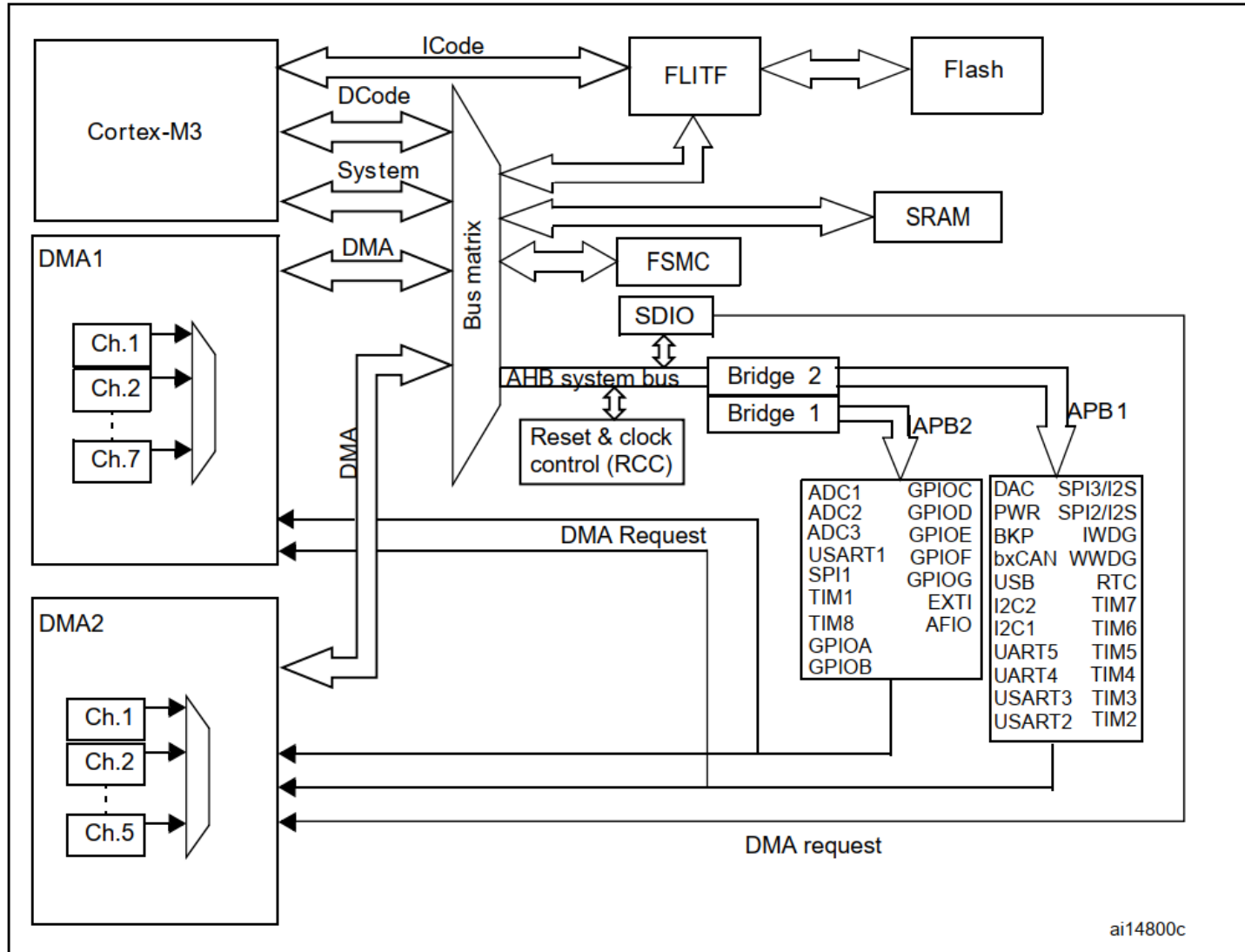
- دسترسی به Peripherals

- Private bus

- برای دسترسی به سطوحی از System Memory نظیر واحد عیب‌یابی استفاده می‌شود.



# آشنایی با ساختمان میکروکنترلر STM32F10X



## Bus Matrix

اتصال باس master مطلوب به باس slave مطلوب

## Master ۴

ICode

DCode

System

Direct Memory Address (DMA)

## Slave ۴

SRAM داخلی

حافظه Flash داخلی

Flexible Static Memory Controller (FSMC)

AHB





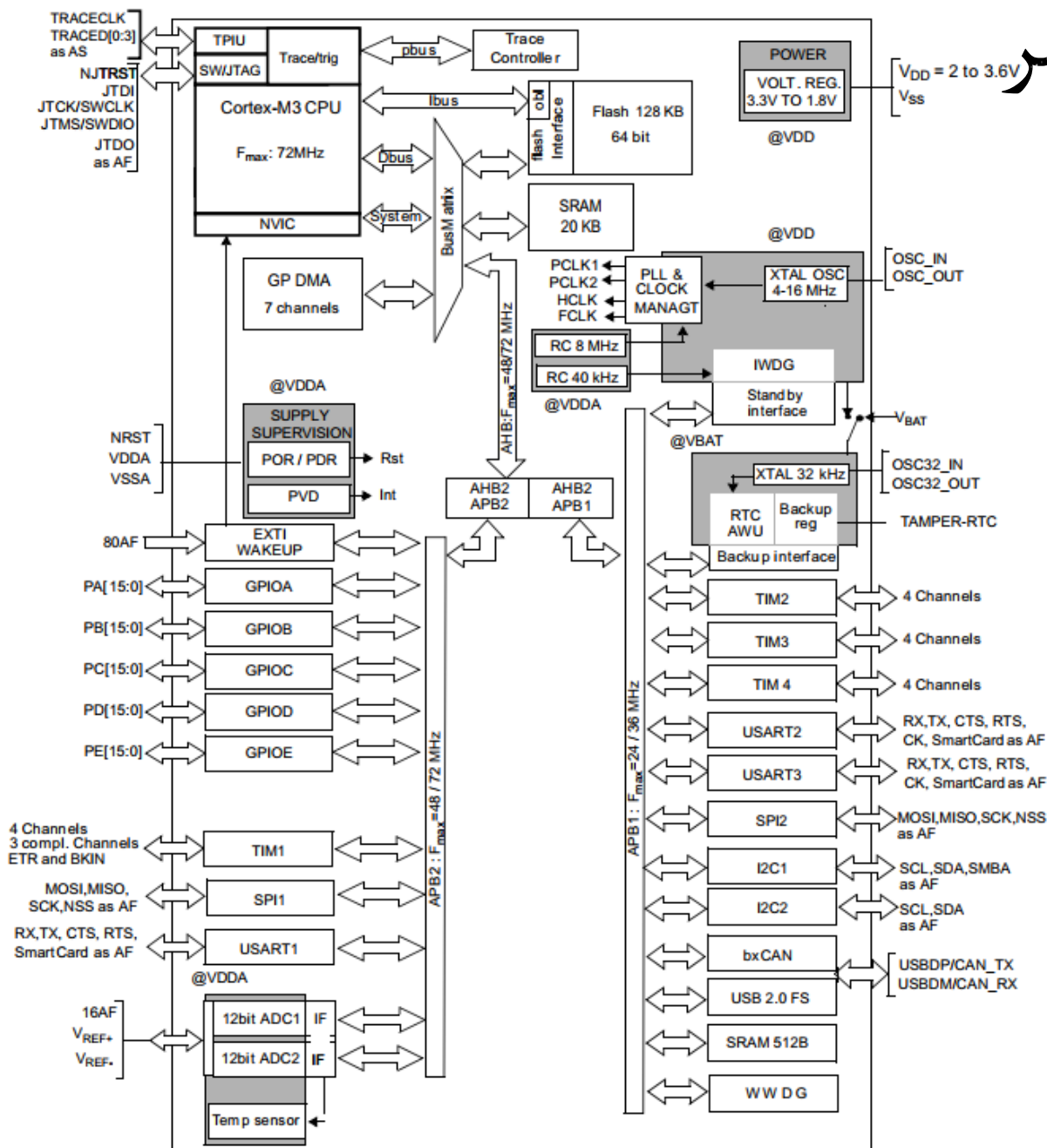
# آشنایی با ساختمان میکروکنترلر STM32F103Cx

▪ واحد RCC در فصل بعد توضیح داده می شود.

▪ پایه های Debugger

Serial Wire (SW) ▪

JTAG ▪



- بخش‌های مختلف و آدرس‌های متناظر با هر یک
- توجه شود که آدرس‌های فوق، آدرس شروع رجیسترهای هر Peripheral می‌باشد (Base Address).



# جمع بندی

- در این فصل با ساختار کلی یک پردازنده Cortex M3 آشنا شدیم.
- واحد پردازش مرکزی (CPU)
  - اجرای دستورات برنامه و ارسال سیگنال‌های کنترلی لازم به قطعات درونی میکروکنترلر
  - حافظه
    - ذخیره اطلاعات کاربر شامل دستورات و اطلاعات
  - درگاه‌های ورودی و خروجی (I/O Ports)
  - واسط ارتباطی بین واحد پردازش مرکزی میکروکنترلر و قطعات جانبی درونی و بیرونی
- با واحدهای پردازش مرکزی و حافظه تا حدودی آشنا شدیم. در ادامه و فصول بعد به معرفی Peripherals خواهیم پرداخت.
- فصل دوم : Reset and Clock Control (RCC) ...

