香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

ECE3080 Microprocessors and Computer Systems

Memory and Buses

Instructor：Tin Lun LAM

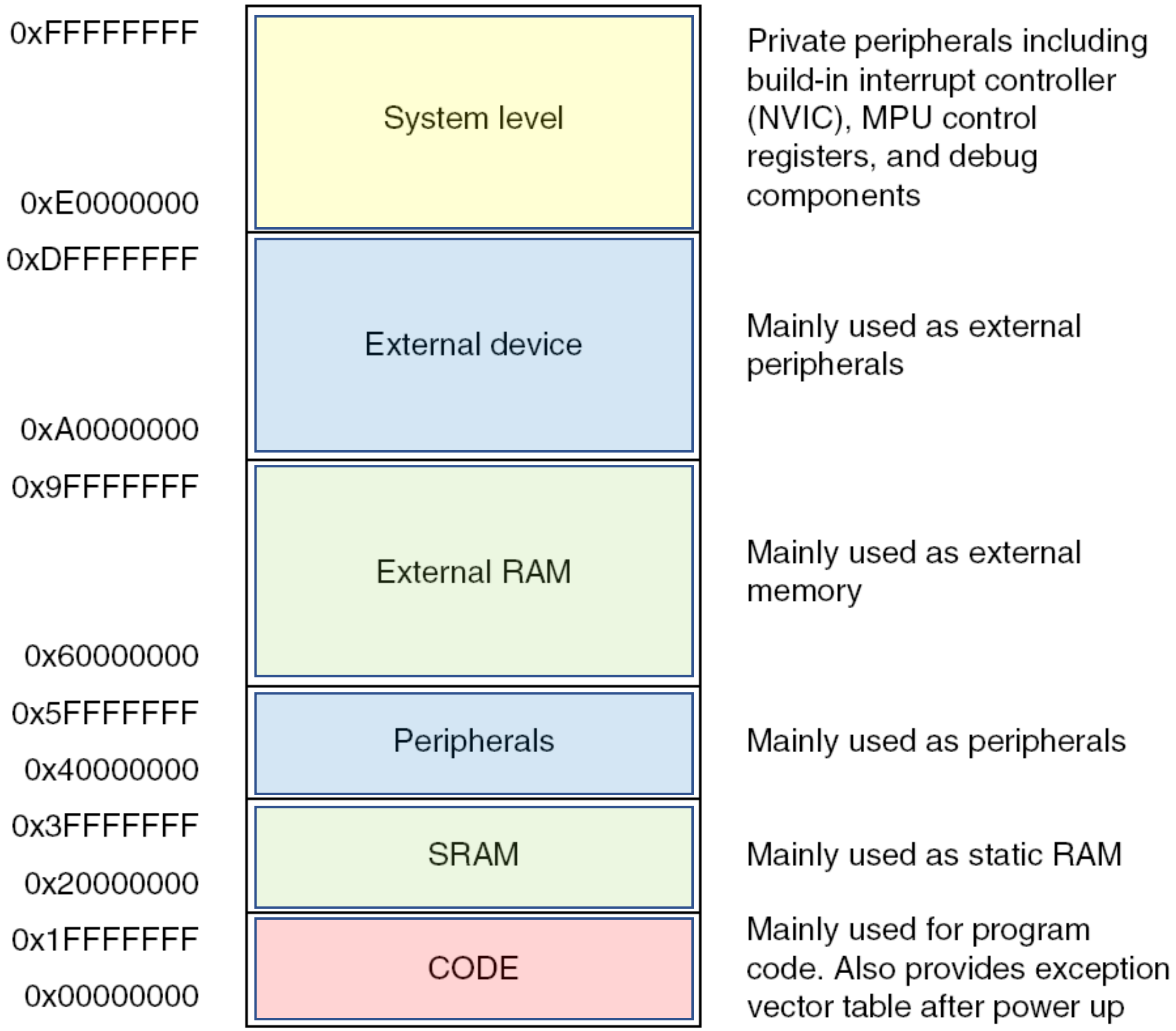
E-mail: tllam@cuhk.edu.cn

URL: https://myweb.cuhk.edu.cn/tllam

Memory Map, Attributes and Access

The Memory Map

- The 4-GB memory space has been separated (fixed address range allocated for different regions) to a few regions for different purposes.



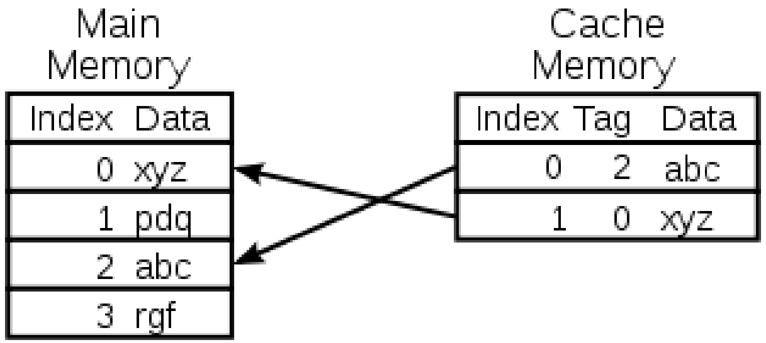
Memory Access Attributes

Memory map also defines the memory attributes of the access:

- Bufferable: Write to memory can be carried out by a write buffer while the processor continues on next instruction execution.
- Executable: The processor can fetch and execute program code from this memory region.
- Shareable: Data in this memory region could be shared by multiple bus masters. Memory system needs to ensure coherency of data between different bus masters in shareable memory region.
- Cacheable: Data obtained from memory read can be copied to a memory cache so that next time it is accessed the value can be obtained from the cache to speed up the program execution.

Cache

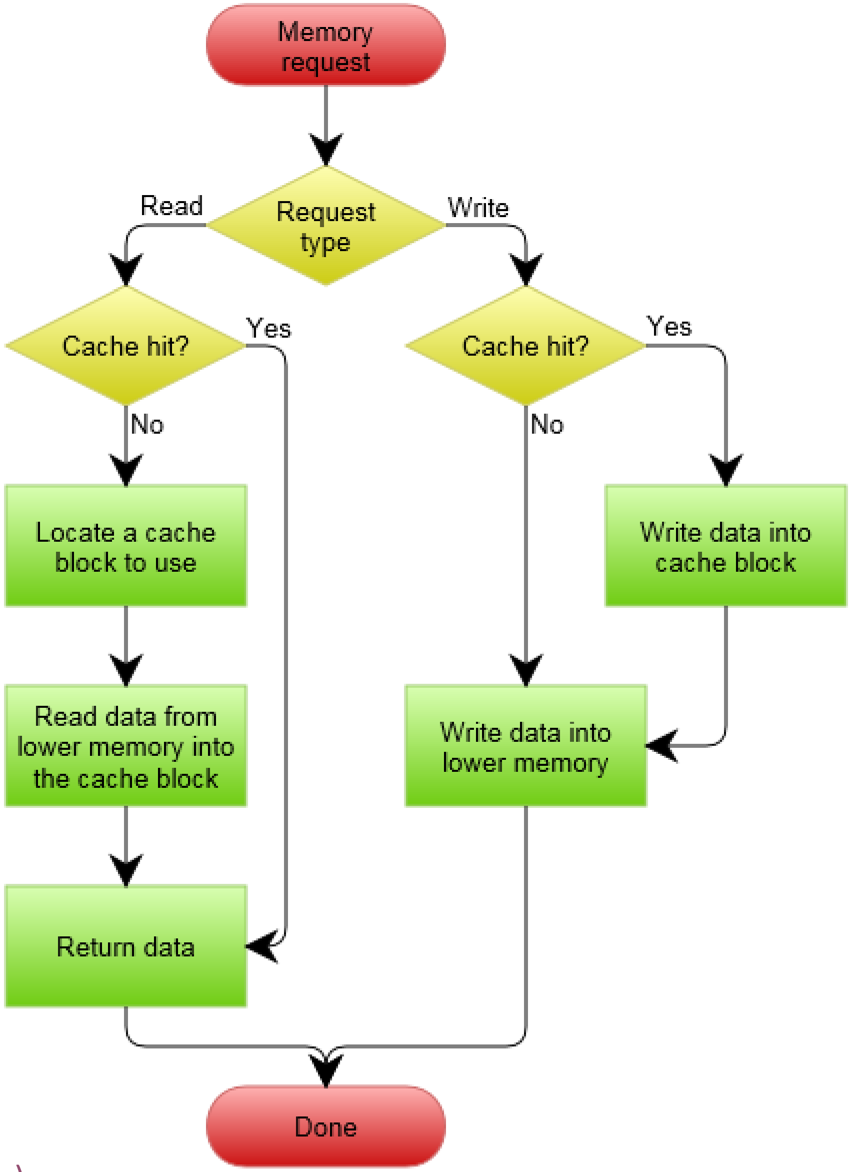
- *A component that transparently stores data so that future requests for that data can be served faster.* (We tend to keep things that we use often close to us).
- When the cache client (a CPU, web browser, operating system) needs a datum in the backing store (the main memory; the lower memory), it first checks the cache to see if there is a copy. Two possible outcomes:
  - **cache hit**: requested datum is contained in the cache, so it can be read from cache quickly without going to the original storage memory.
  - **cache miss**: the data has to be recomputed or fetched from its original storage location; comparatively slower.
- Useful in many areas of computing because access patterns in typical computer applications have locality of reference.



Q: What are the pros and cons of implementing cache, and what are the issues?

Writing policies in Cache

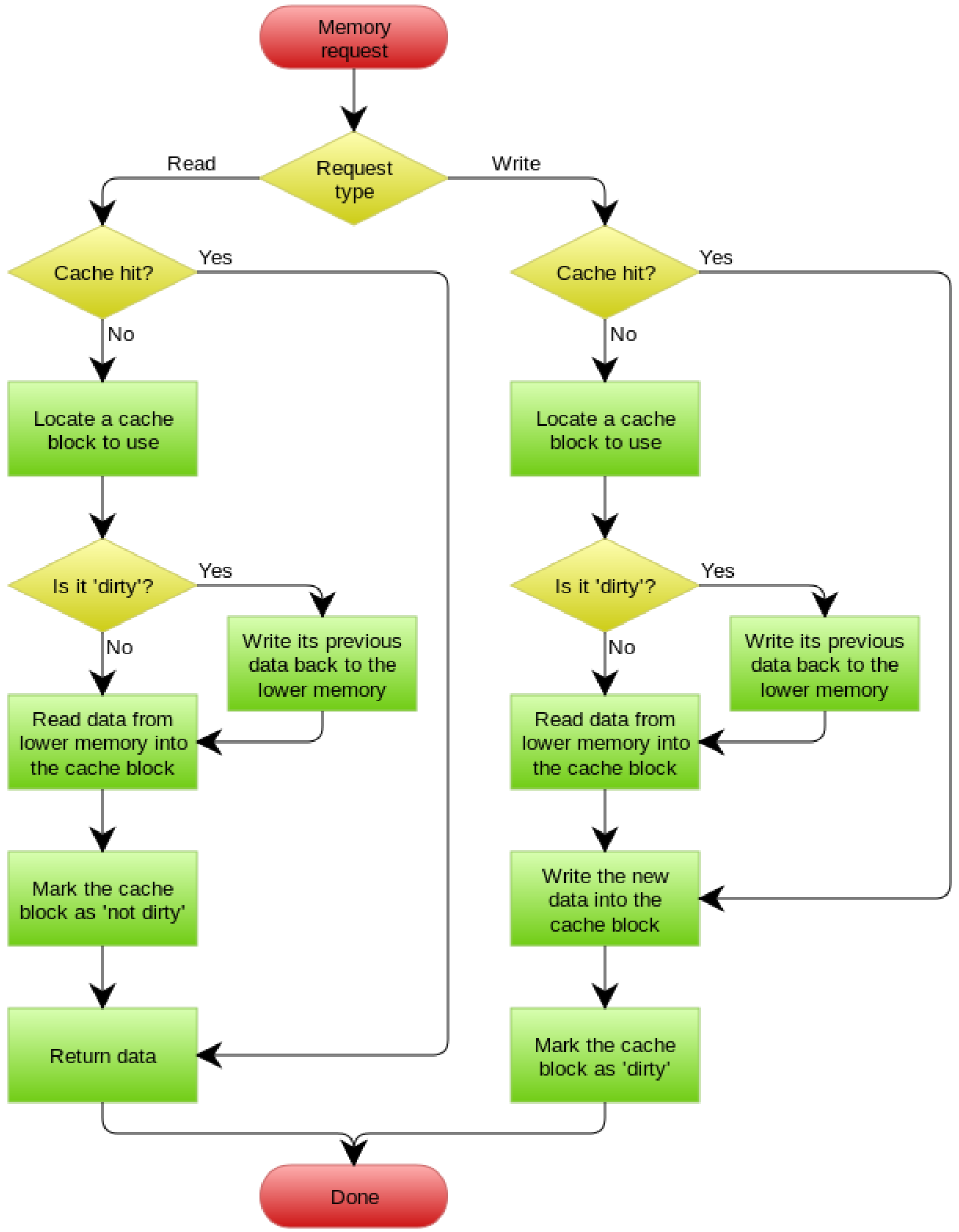
- *Write-through* (WT): write is done synchronously both to the cache and to the backing store.
- *No-write allocate* (also called *write-no-allocate* or *write around*): datum at the missed-write location is not loaded to cache, and is written directly to the backing store. In this approach, only system reads are being cached.



http://en.wikipedia.org/wiki/Cache_(computing)

Writing policies in Cache

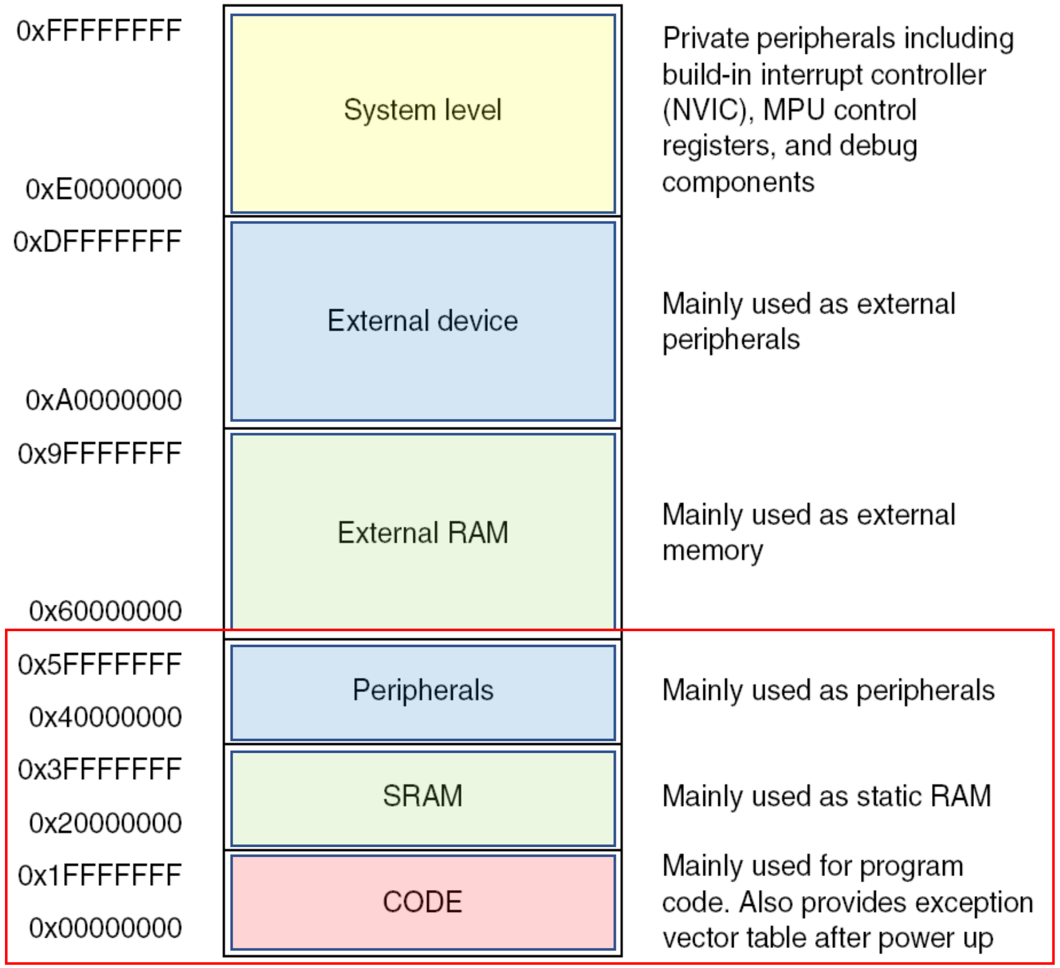
- *Write-back* (WB) (or *write-behind*): initially, writing is done only to the cache. The write to the backing store is postponed until the cache blocks containing the data are about to be modified/replaced by new content.
- *Write allocate* (also called *fetch on write*): datum at the missed-write location is loaded to cache, followed by a write-hit operation. In this approach, write misses are similar to read misses.



http://en.wikipedia.org/wiki/Cache_(computing)

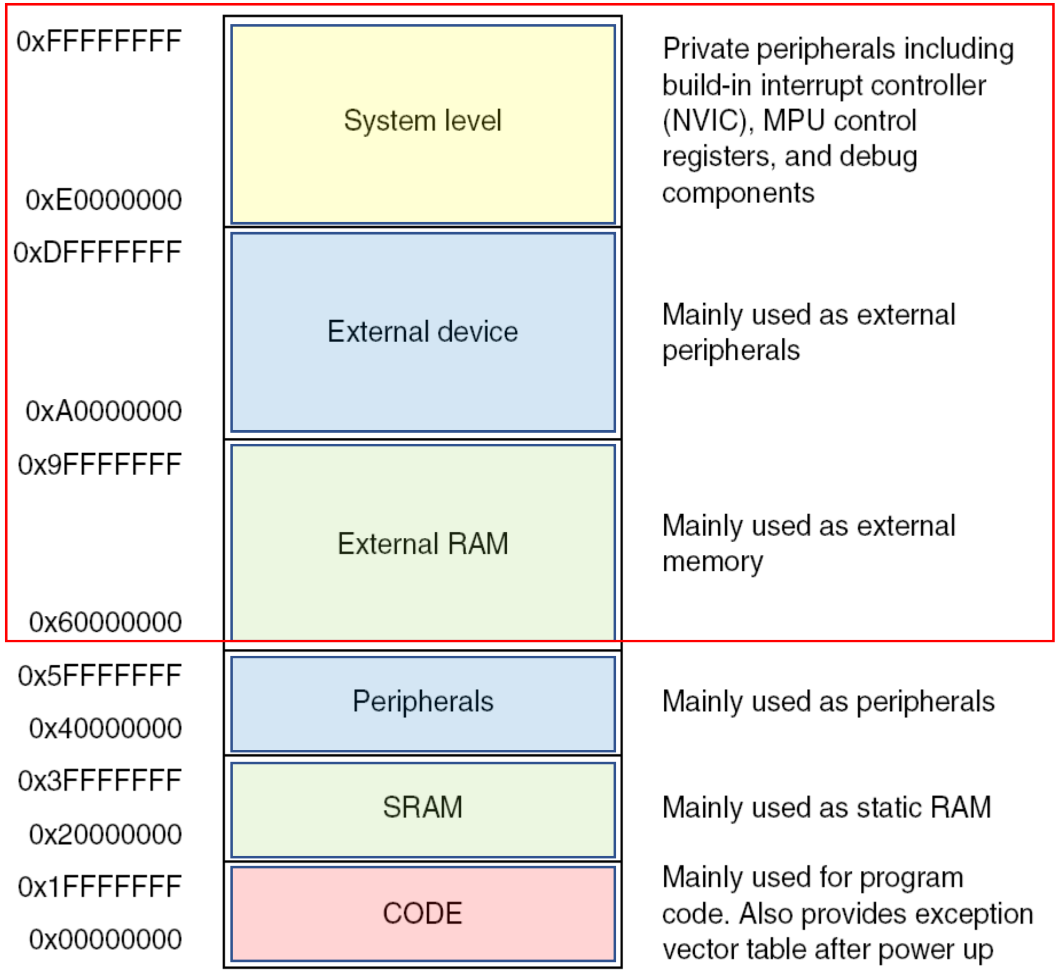
Memory access attributes for each memory region

- *Code memory region* : *executable*; cache attribute is write through (WT); can put data memory here. If data operations are carried out for this region, they will take place via the data bus interface. Write transfers to this region are *bufferable*.
- *SRAM memory region*: intended for on-chip RAM. Write transfers to this region are *bufferable*, and the cache attribute is write back, write allocated (WB-WA) . This region is *executable*, so you can copy program code here and execute it.
- *Peripheral region*: intended for peripherals; *noncacheable*; cannot execute instruction code here (*Execute Never*, or XN in ARM documentation)



Memory access attributes for each memory region

- *External RAM region*: intended for either on-chip or off-chip memory; can *execute* code here. For cache, *cacheable* (WB-WA) for memory (0x60000000–0x7FFFFFFF) and *cacheable* (WT) for (0x80000000–0x9FFFFFFF).
- *External devices*: intended for external devices and/or shared memory that needs *ordering** / *nonbuffered* accesses; *nonexecutable*.
- *System level*: for private peripherals and vendor-specific devices; *nonexecutable*. For the PPB memory range, the accesses are strongly ordered (all fixed) (*noncacheable*, *nonbufferable*). For the vendor-specific memory region, the accesses are bufferable and noncacheable.



Default Memory Access Permission

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

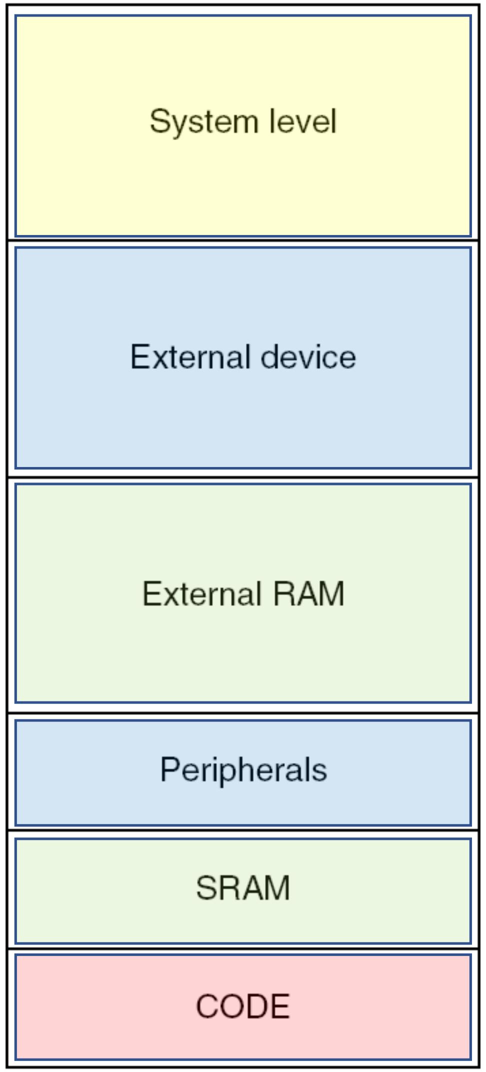
system

PPB

Table 5.1 Default Memory Access Permissions

Memory Region	Address	Access in User Program
Vendor specific	0xE0100000–0xFFFFFFFF	Full access
ROM table	0xE00FF000–0xE00FFFFF	Blocked; user access results in bus fault
External PPB	0xE0042000–0xE00FEFFF	Blocked; user access results in bus fault
ETM	0xE0041000–0xE0041FFF	Blocked; user access results in bus fault
TPIU	0xE0040000–0xE0040FFF	Blocked; user access results in bus fault
Internal PPB	0xE000F000–0xE003FFFF	Blocked; user access results in bus fault
NVIC	0xE000E000–0xE000EFFF	Blocked; user access results in bus fault, except Software Trigger Interrupt Register that can be programmed to allow user accesses
FPB	0xE0002000–0xE0003FFF	Blocked; user access results in bus fault
DWT	0xE0001000–0xE0001FFF	Blocked; user access results in bus fault
ITM	0xE0000000–0xE0000FFF	Read allowed; write ignored except for stimulus ports with user access enabled
External device	0xA0000000–0xDFFFFFFF	Full access
External RAM	0x60000000–0x9FFFFFFF	Full access
Peripheral	0x40000000–0x5FFFFFFF	Full access
SRAM	0x20000000–0x3FFFFFFF	Full access
Code	0x00000000–0x1FFFFFFF	Full access

When a user access is blocked, the fault exception takes place immediately.

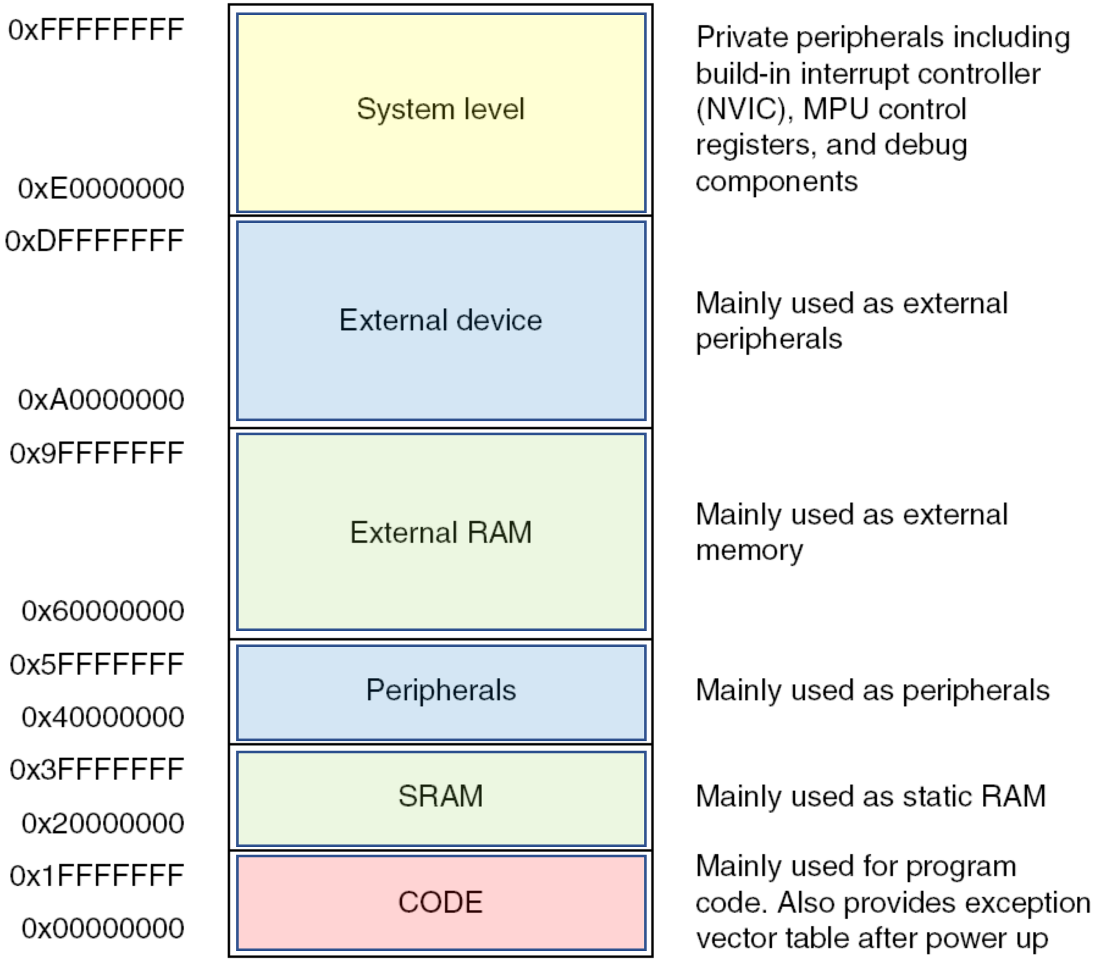


Memory Access Attributes

- The Cortex-M3 bus interfaces output the memory access attributes information to the memory system for each instruction and data transfer.
- No cache memory or cache controller used; however, a cache unit can be added on the microcontroller, then memory attribute information can be used to define the memory access behaviors.

The Memory Map

- To access memory locations, it needs address bus, data bus and control bus.
- Most of memory space is inside in the Cortex microcontroller and the buses have been hidden, meaning that we cannot monitor them.

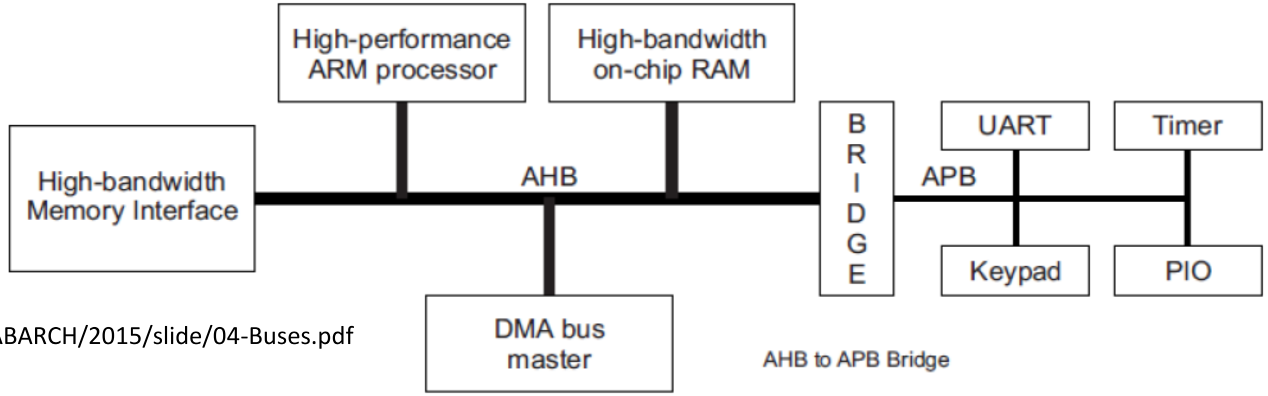


On-chip Buses

Advanced Microcontroller Bus Architecture (AMBA)

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

- Advanced High-performance Bus (AHB)
- Advanced Peripheral Bus (APB)

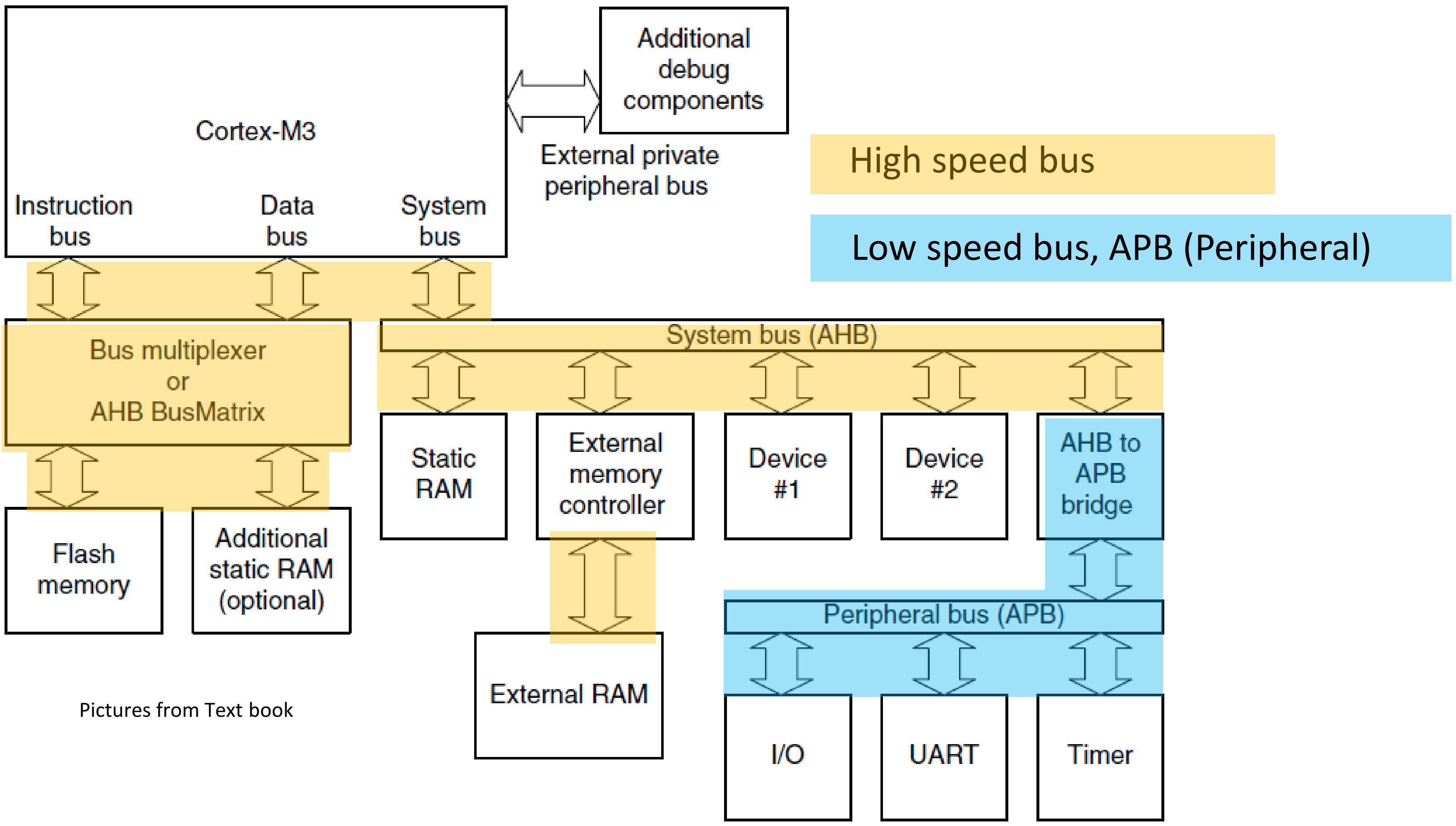


www-micrel.deis.unibo.it/LABARCH/2015/slide/04-Buses.pdf

AHB
- High performance
- Pipelined operation
- Burst transfers
- Multiple bus masters
- Split transactions

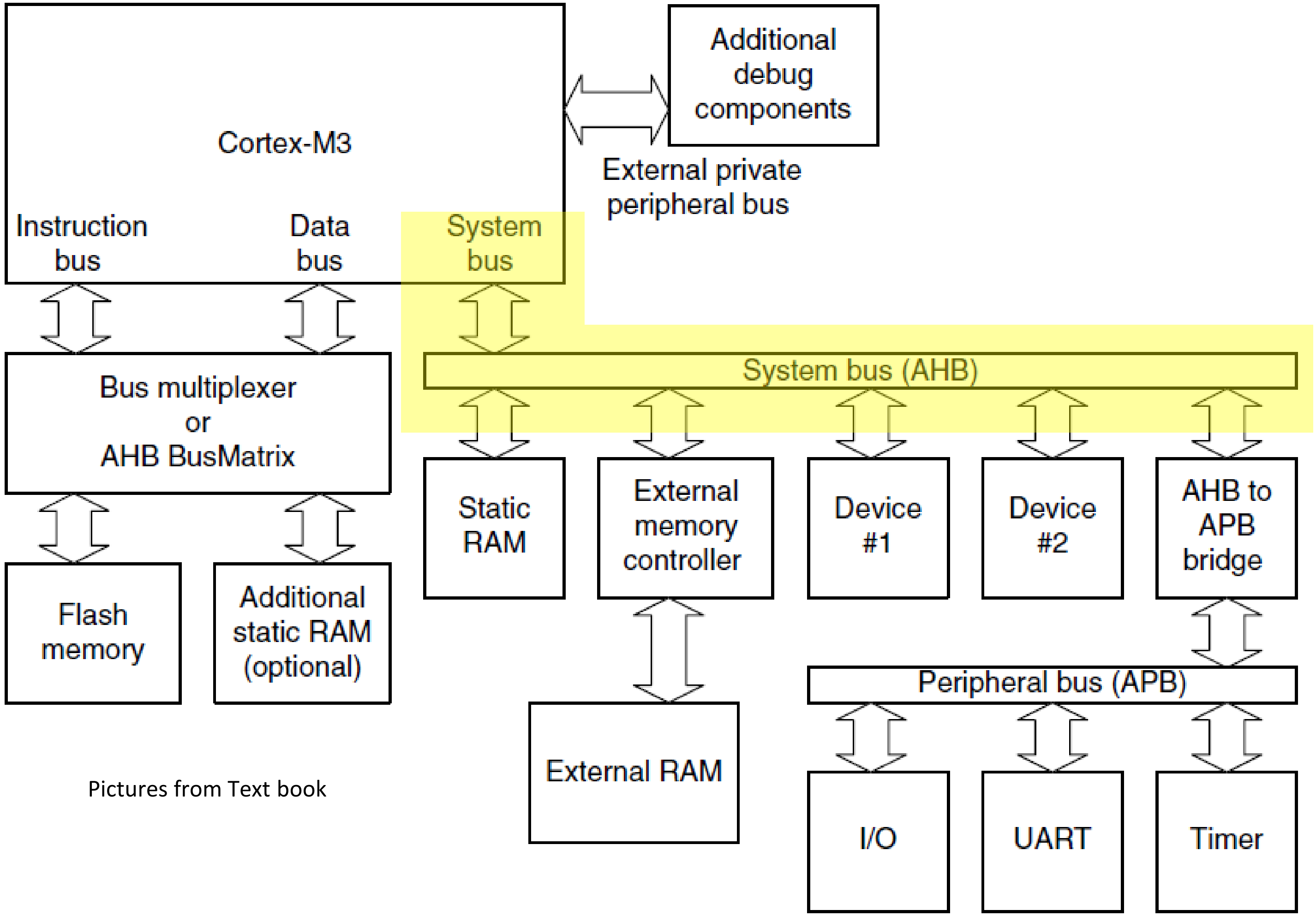
APB
- Low power
- Latched address/control
- Simple interface
- Suitable of many peripherals

Instruction, Data and System Buses



Pictures from Text book

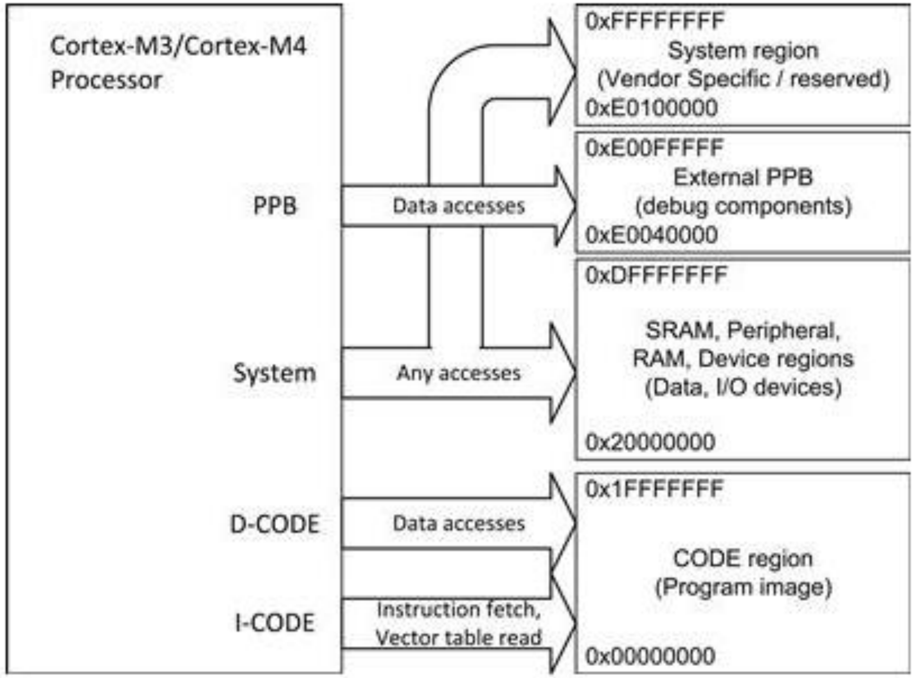
System Buses



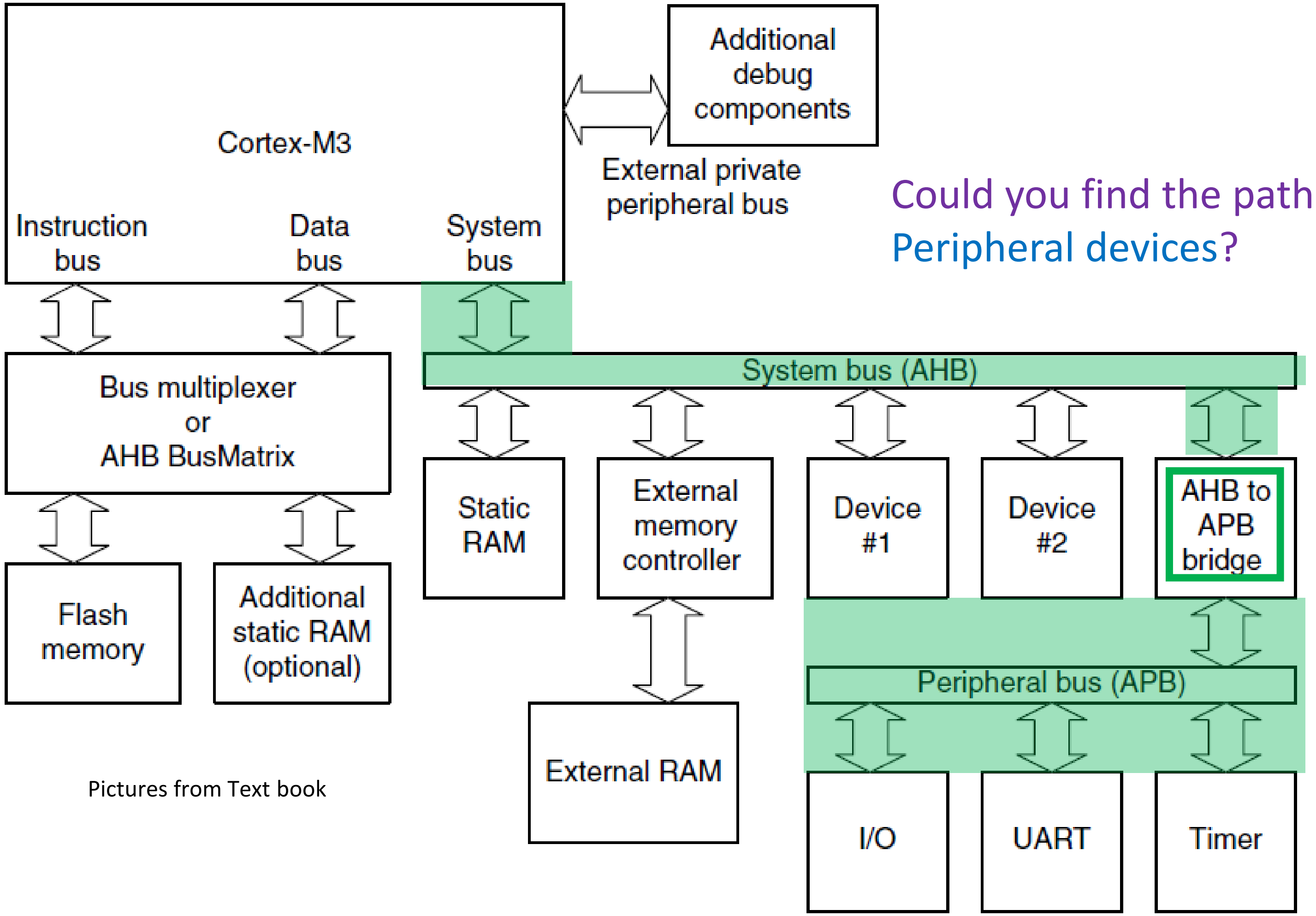
Pictures from Text book

System Bus

- System bus: used to access memory (static RAM, external RAM), peripherals, external devices, and part of the system-level memory regions
- A 32-bit bus based on the AHB-Lite bus protocol; for instruction fetch and data access in memory regions from 0x2000 0000 to 0xDFFF FFFF and 0xE010 0000 to 0xFFFF FFFF.



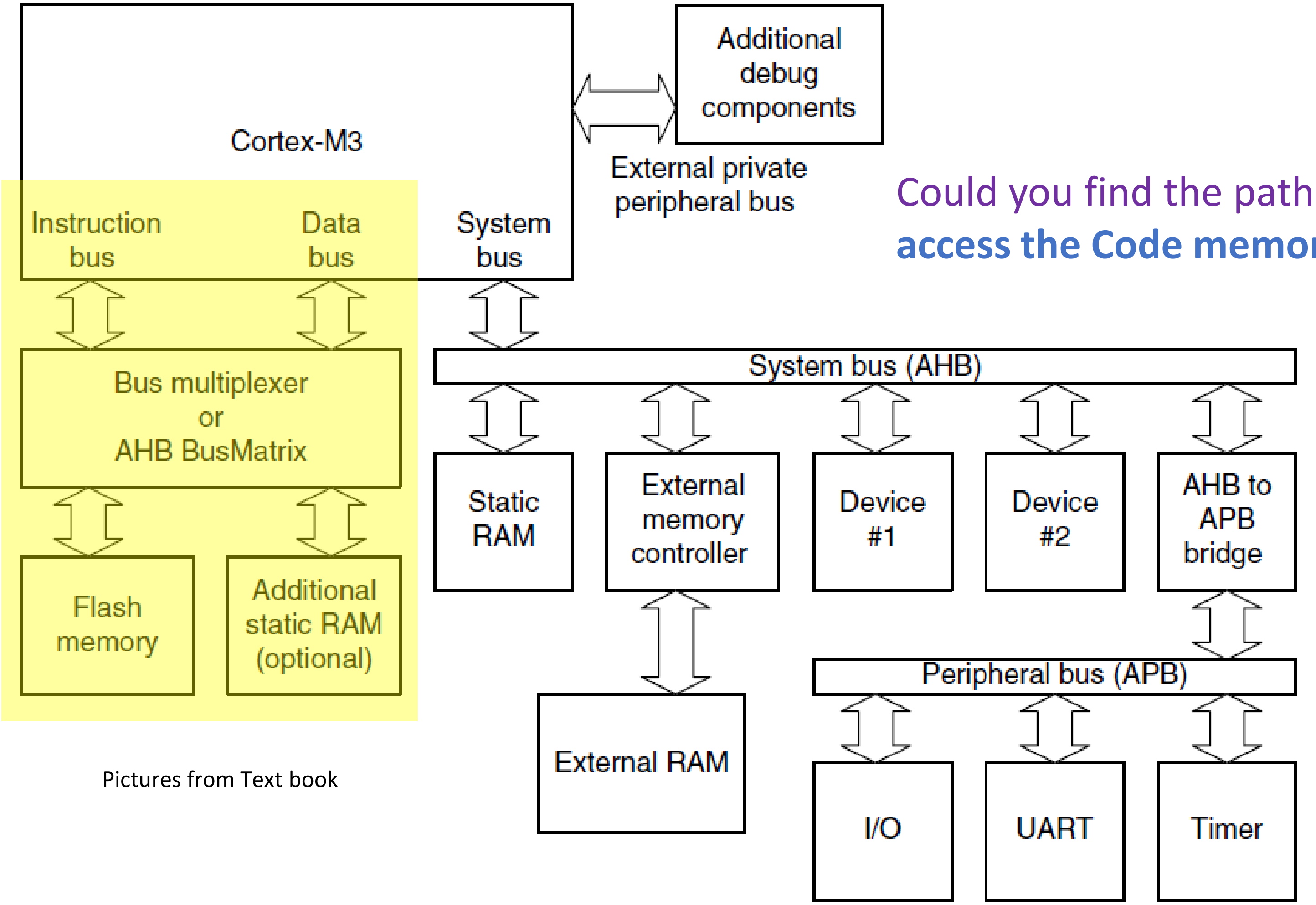
Peripheral Buses



Pictures from Text book

Could you find the path from Cortex to Peripheral devices?

Code Memory Buses



Pictures from Text book

Could you find the path from Cortex to **access the Code memory region**?

Code Memory Buses

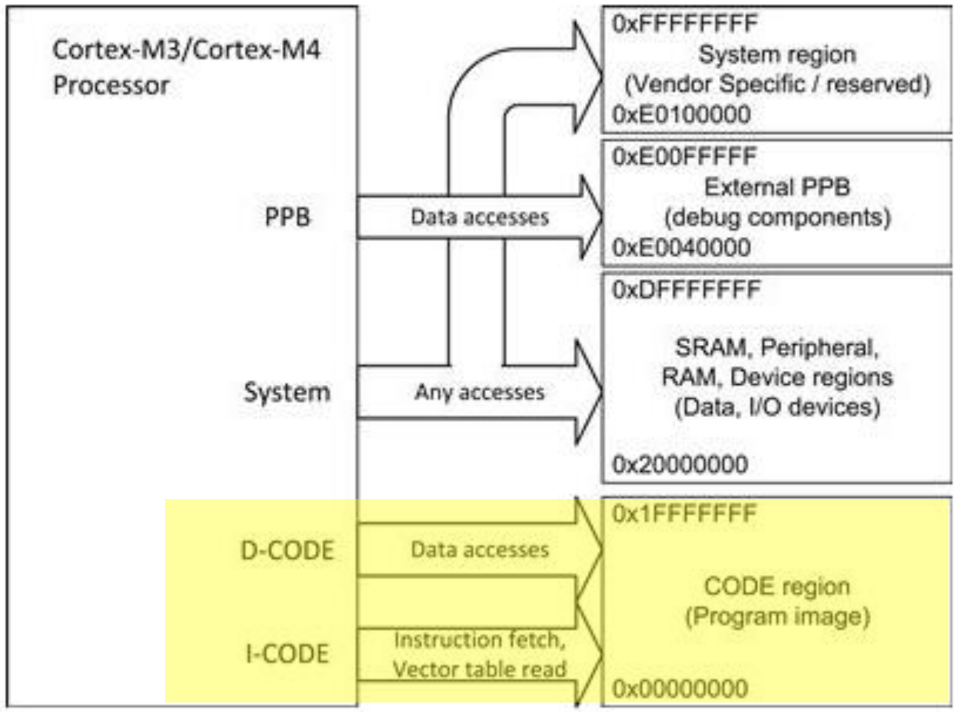
Code memory buses: physically consist of two buses, one called <u>I-Code bus</u> and the other called <u>D-Code bus</u>.

The I-Code Bus

- A 32-bit bus based on the AHB-Lite bus protocol for instruction fetches in memory regions from 0x00000000 to 0x1FFFFFFF.
- Instruction fetches are in word size (even for 16-bit Thumb instructions). → *CPU core could fetch up to two Thumb instructions at a time.*

The D-Code Bus

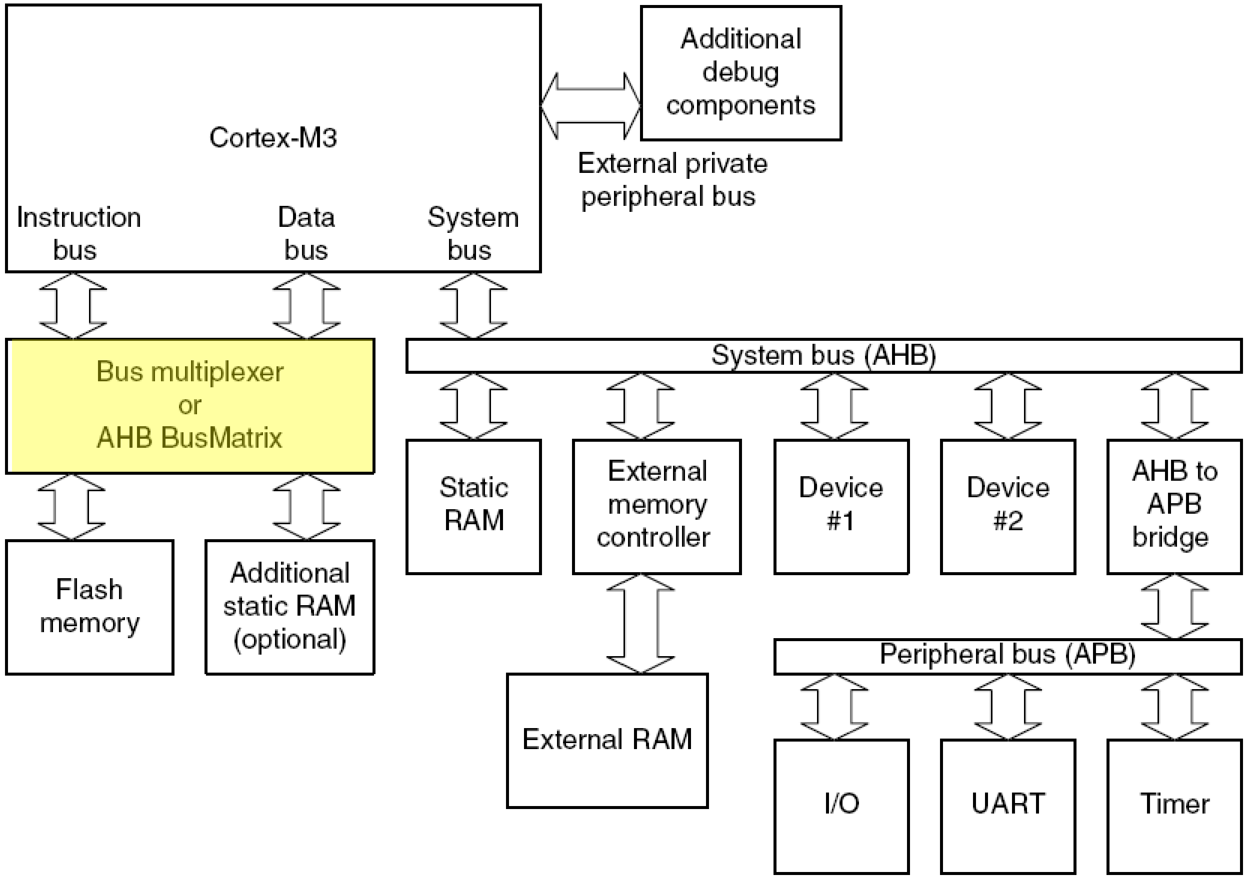
- A 32-bit bus based on the AHB-Lite bus protocol for data access in memory regions from 0x00000000 to 0x1FFFFFFF.



Code Memory Buses

香港中文大學(深圳) The Chinese University of Hong Kong, Shenzhen

- Since the Code memory region can be accessed by the instruction bus (if it is an instruction fetch) and from the data bus (if it is a data access), an AHB bus switch called the *BusMatrix or an AHB* bus multiplexer (simpler) is needed.
- If a bus multiplexer is used, the transfers cannot take place at the same time. However, the circuit size would be smaller.



BusMatrix

- With BusMatrix, if the instruction bus and the data bus want to access different memory devices (for example, an instruction fetch from fetch and a data bus reading data from the additional SRAM), the transfers can be carried out simultaneously.
- Common Cortex-M3 microcontroller designs use system bus for SRAM connection.
- The main SRAM block should be connected through the system bus interface, using the SRAM memory address region. This allows data access to be carried out at the same time as instruction access. It also allows setting up of Boolean data types by using the bit-band.

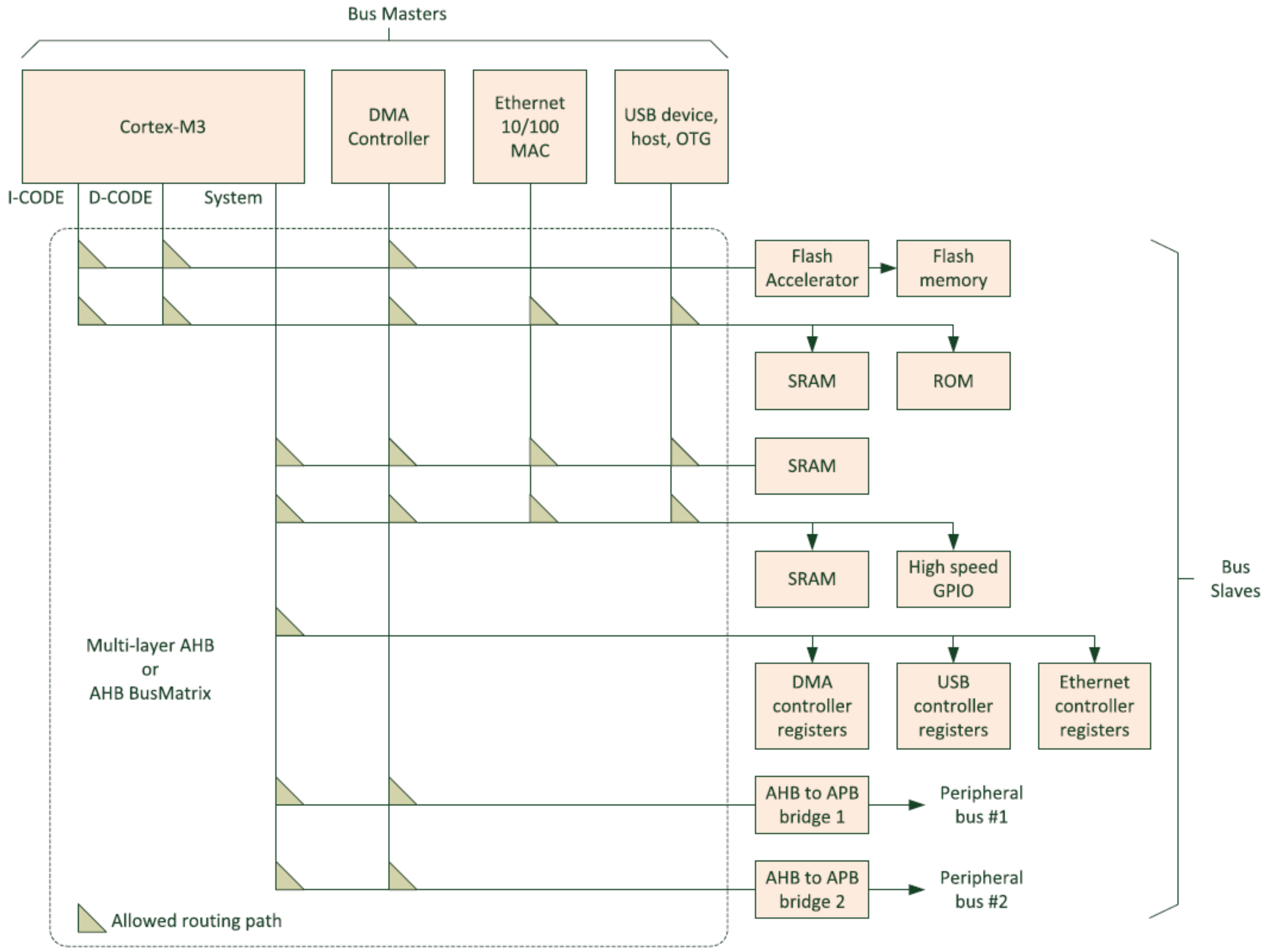
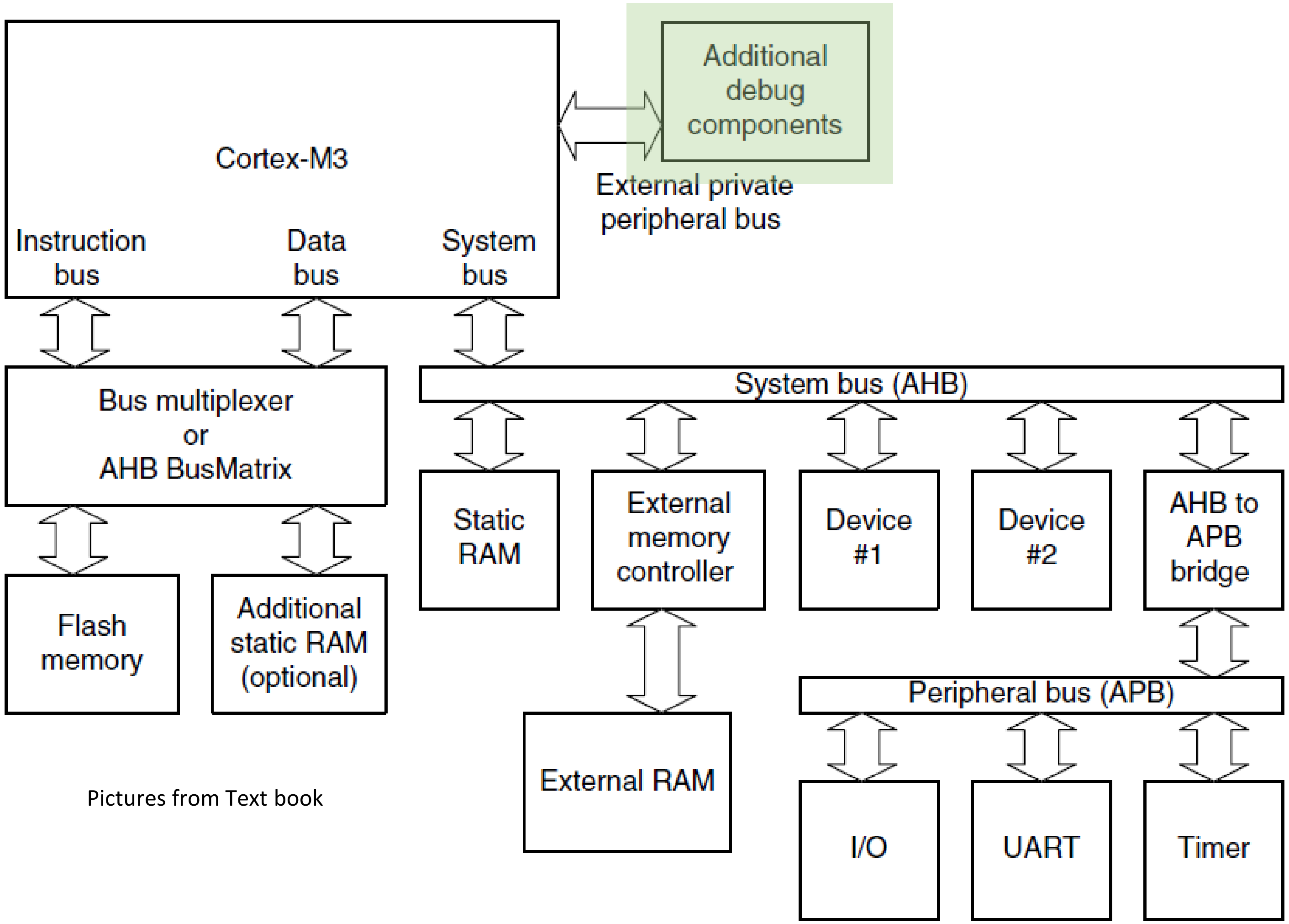


FIGURE 6.5

Multi-layer AHB example (NXP LPC1700)

Ref: *The Definitive Guide to ARM ® Cortex ®-M3 and Cortex-M4 Processors*

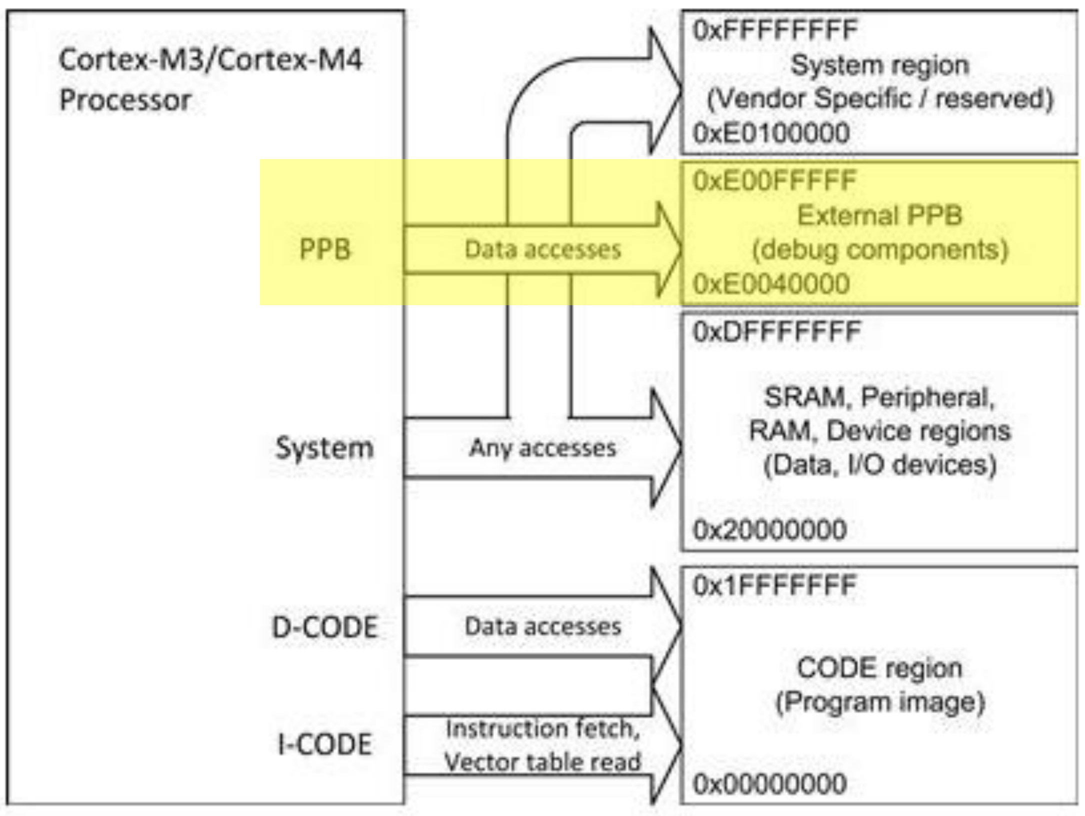
Private Peripheral Bus (PPB)



Pictures from Text book

Private Peripheral Bus (PPB)

- **Private peripheral bus (PPB)**: provides access to a part of the system-level memory dedicated to private peripherals, such as debugging components.
- The External PPB is a 32-bit bus based on the APB bus protocol; for private peripheral accesses in memory regions 0xE004 0000 to 0xE00F FFFF.
- The memory region that can be used for attaching extra peripherals on this bus is only 0xE004 2000 to 0xE00F F000 (as some are used for TPIU, ETM, and the ROM table already).



Ref: *The Definitive Guide to ARM ® Cortex ®-M3 and Cortex-M4 Processors*

Debug Access Port (DAP) Bus

- This is for attaching debug interface blocks such as SWJ-DP (combined JTAG-DP and SW-DP) or SW-DP.
- Do not use this bus for other purposes.

STMicroelectronics ST-LINK

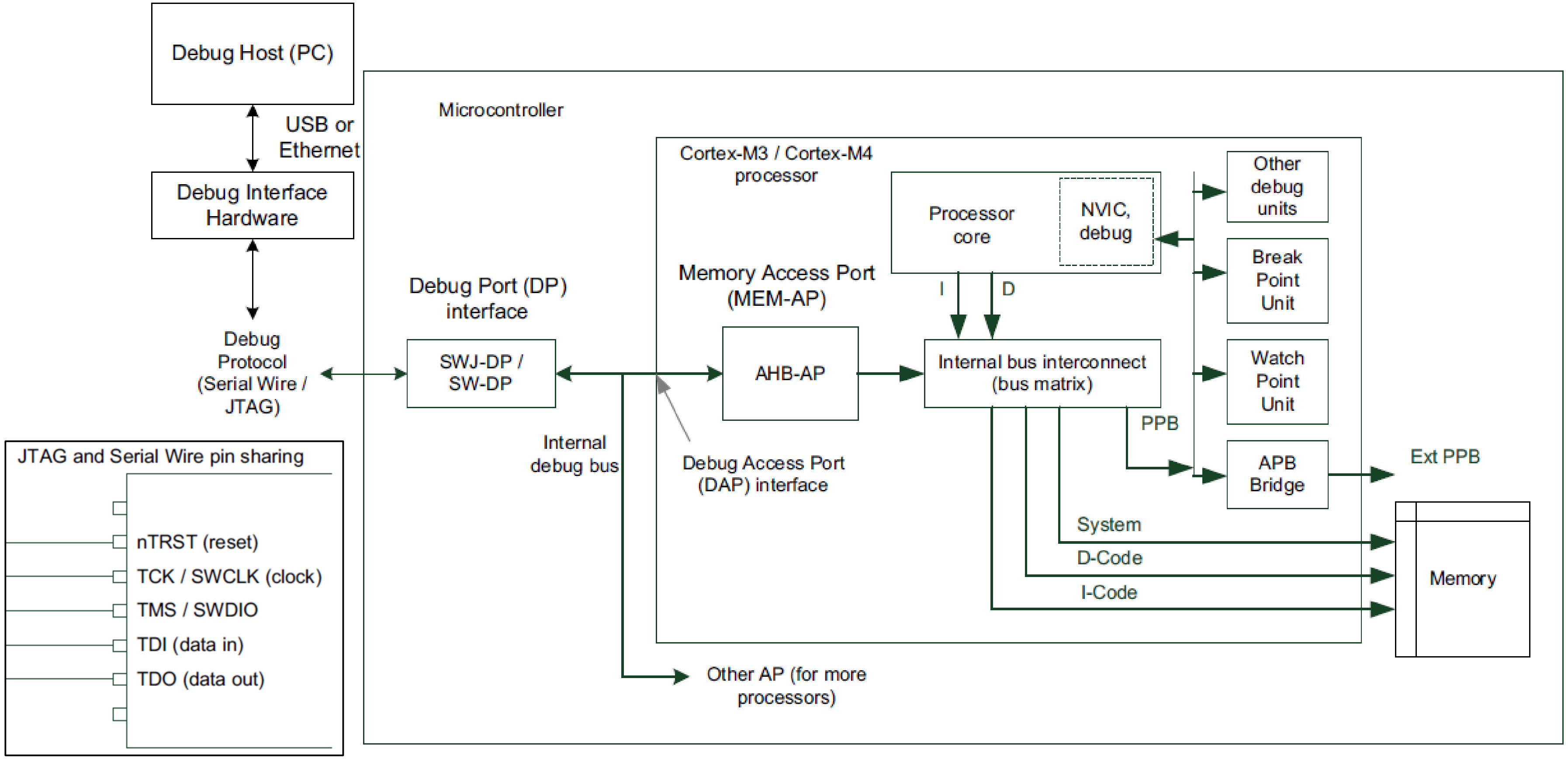
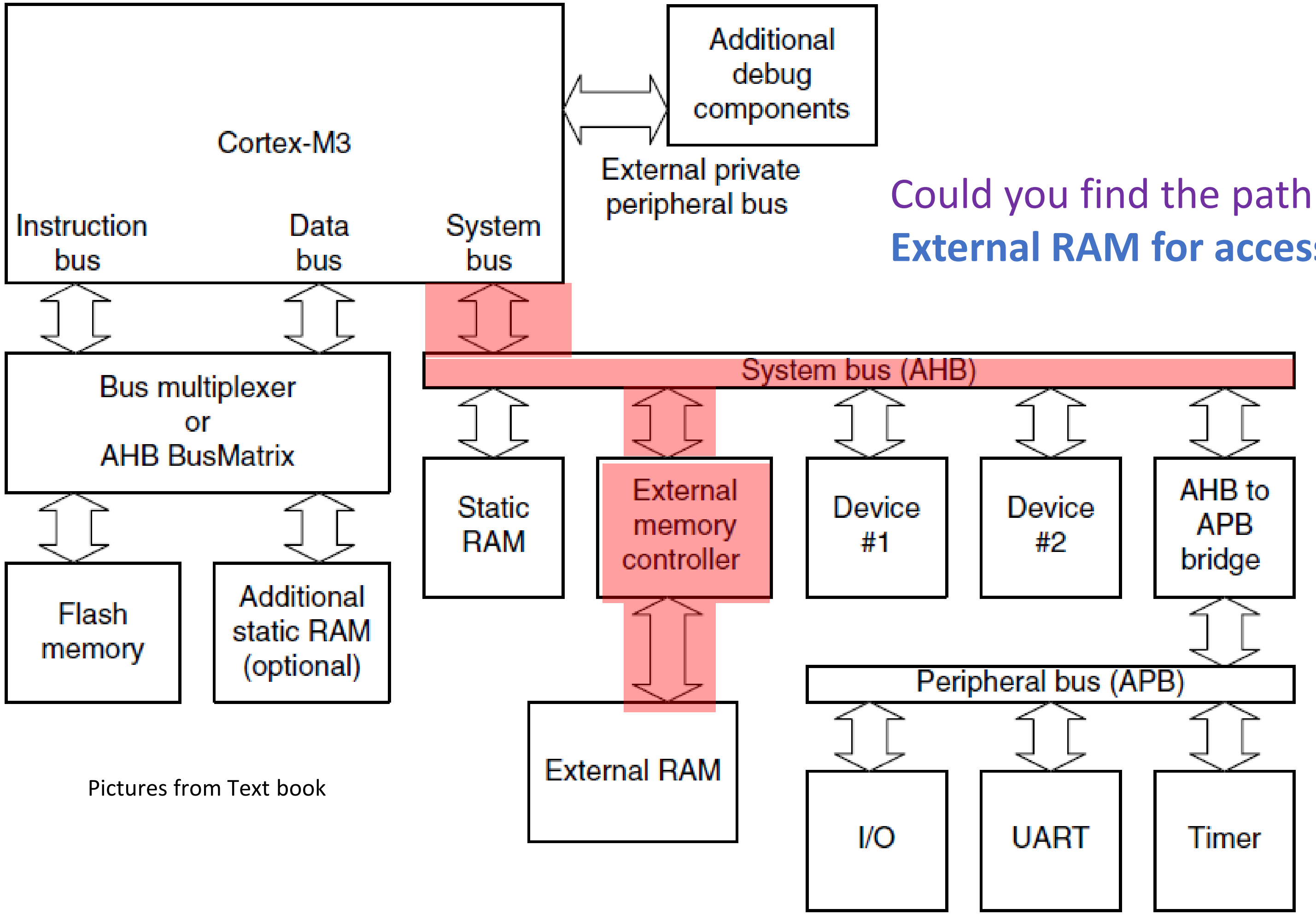


FIGURE 14.5

Connection from debug host to the Cortex®-M processor

Ref: *The Definitive Guide to ARM ® Cortex ®-M3 and Cortex-M4 Processors*

External RAM

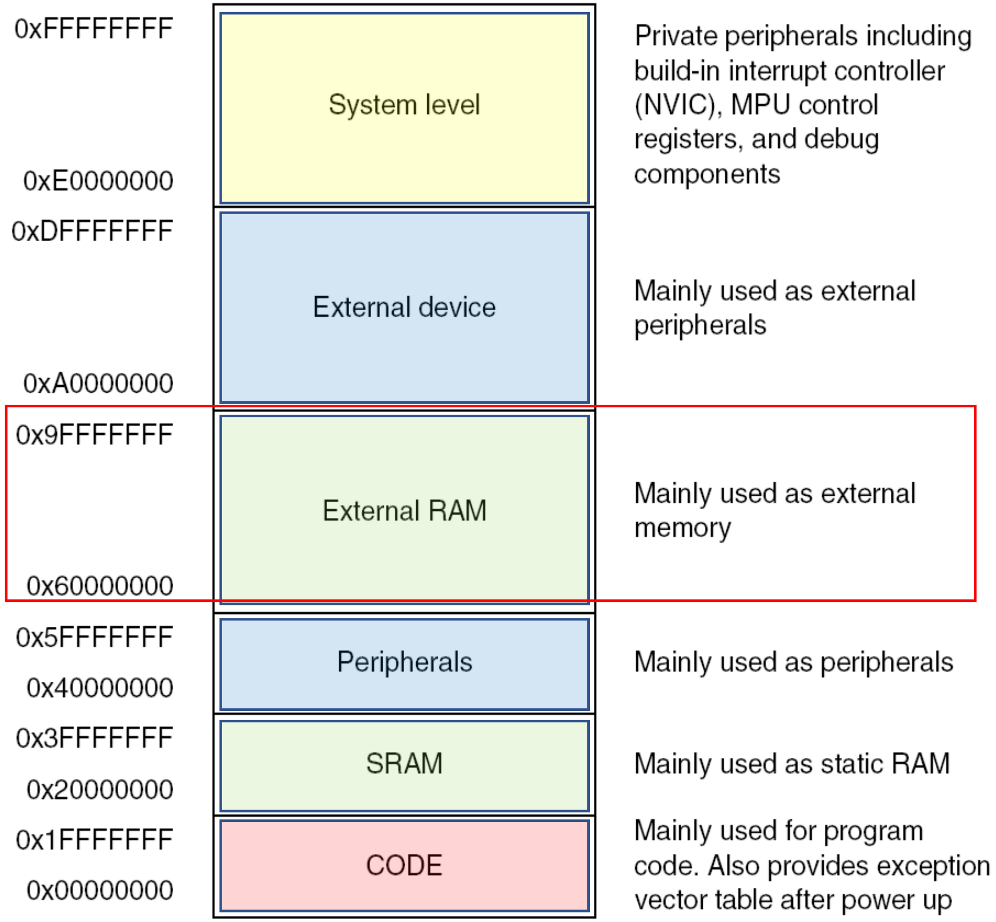


Pictures from Text book

Could you find the path from Cortex to **External RAM for accessing data?**

External RAM

- If an external memory interface (physical pins) is used, an external memory controller (e.g. FSMC) is required, because off-chip memory devices cannot be connected to AHB directly. The external memory controller can be connected to the system bus of the Cortex-M3.



Flexible static memory controller (FSMC)

 30

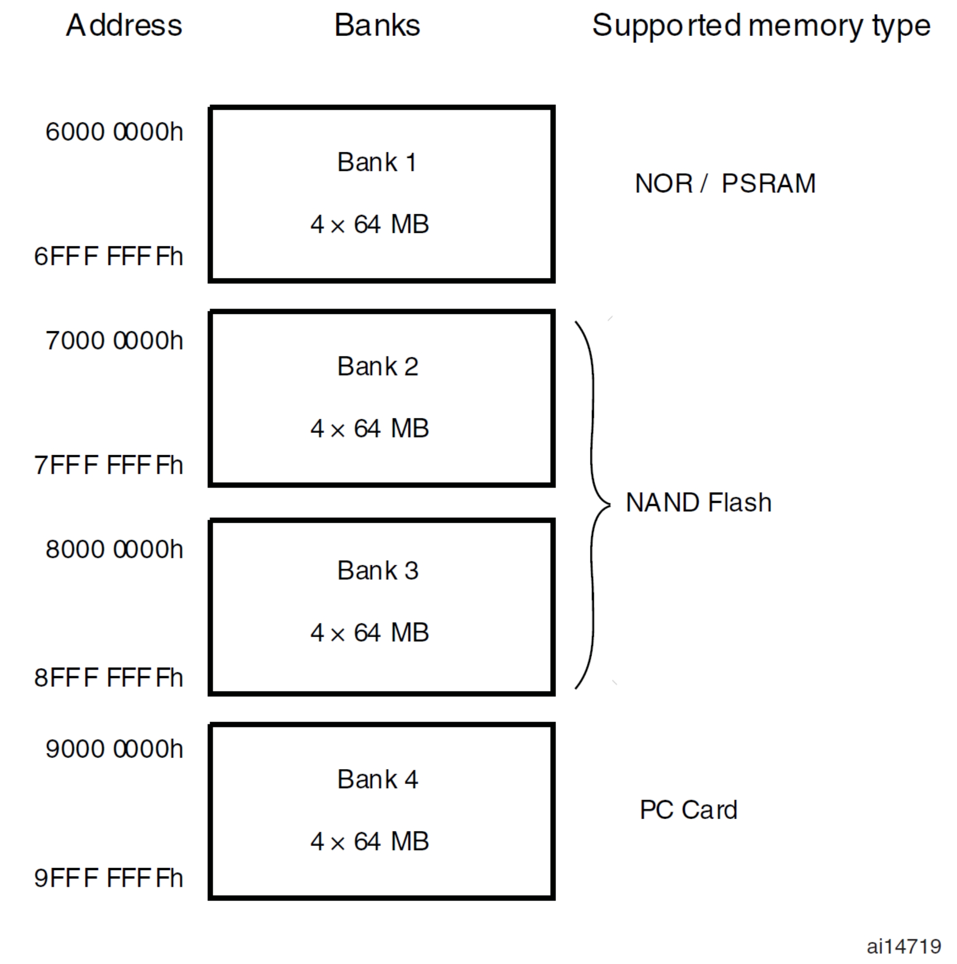
Flexible static memory controller (FSMC)

- The FSMC block is able to interface with synchronous and asynchronous memories and 16-bit PC memory cards. Its main purpose is to:
  - Translate the AHB transactions into the appropriate external device protocol
  - Meet the access timing requirements of the external devices
- All external memories share the addresses, data and control signals with the controller.
- Each external device is accessed by means of a unique chip select. The FSMC performs only one access at a time to an external device.

Memory Access

- Cortex-M FSMC provides a few accessing mode for different types of memory, NOR (PSRAM) Flash, NAND Flash and PC card (PCMCIA card).
- Bank 1 used to address up to 4 NOR Flash or PSRAM (Pseudo Static RAM) memory devices. This bank is split into 4 NOR/PSRAM subbanks with 4 dedicated Chip Selects, as follows:
  - Bank 1 - NOR/PSRAM 1
  - Bank 1 - NOR/PSRAM 2
  - Bank 1 - NOR/PSRAM 3
  - Bank 1 - NOR/PSRAM 4
- Banks 2 and 3 used to address NAND Flash devices (1 device per bank)
- Bank 4 used to address a PC Card device

Figure 186. FSMC memory banks



(Pictures from RM0008)

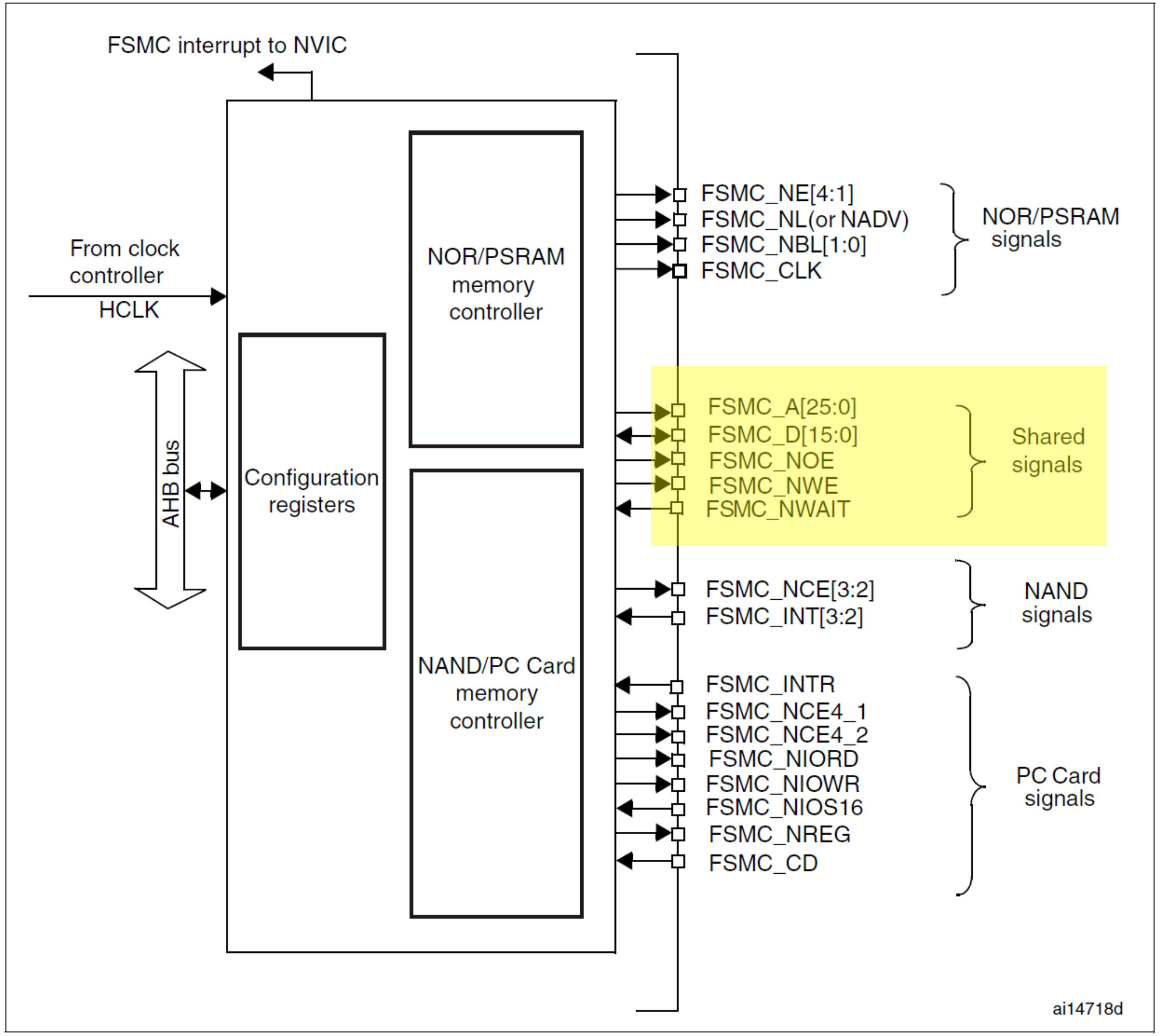
Flexible static memory controller (FSMC)

香港中文大學(深圳) The Chinese University of Hong Kong, Shenzhen

- FSMC provides a few assessing modes to external RAM region. We only focus on a few signals on system bus, which include:
  - A (address; the address bus signal)
  - D (data; the data bus signal)
  - NOE (output enable; one of the control bus signals)
  - NWE (write enable; one of the control bus signals)
  - NEx (chip enable; one of the control bus signals)

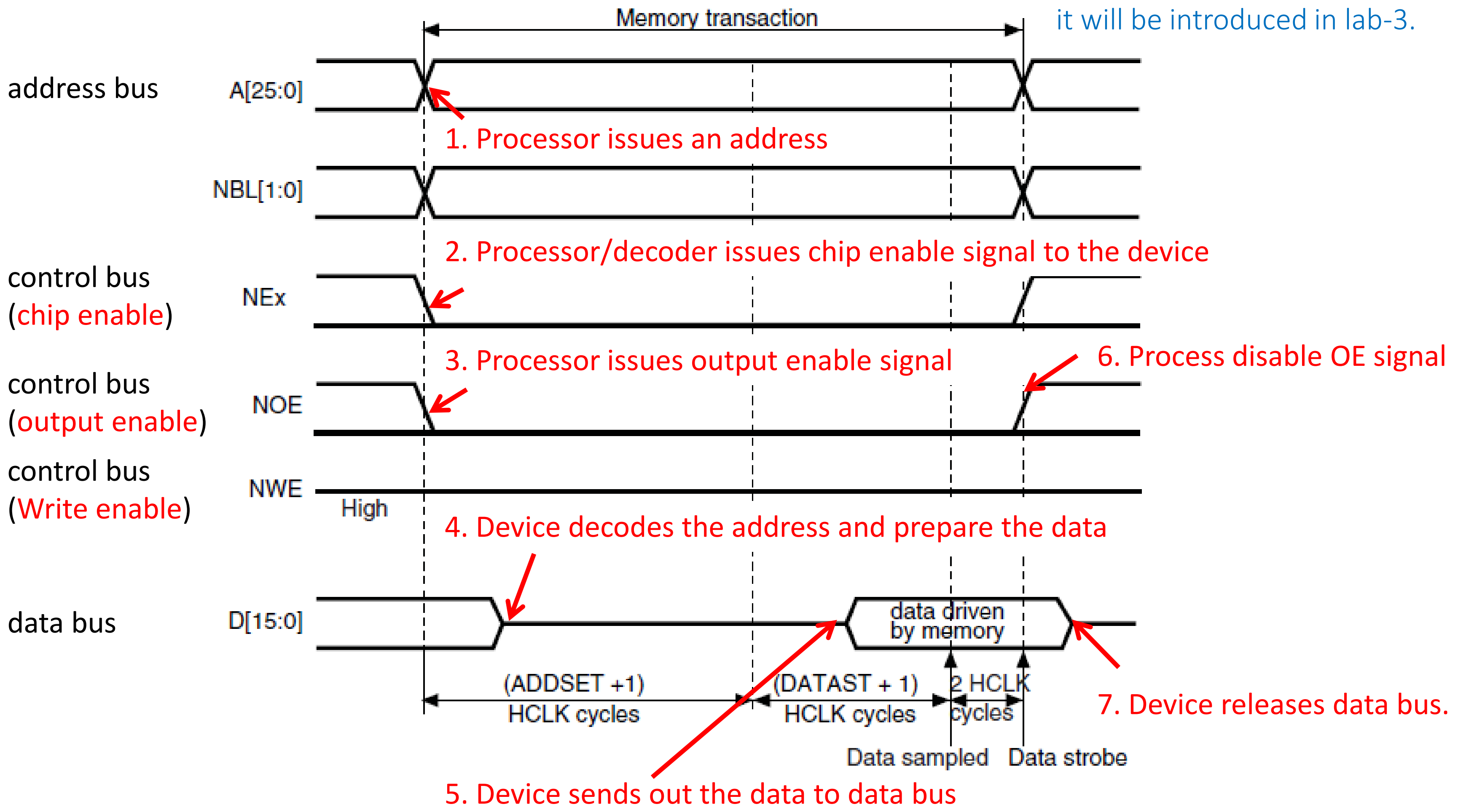
(Prefix N means inverted input)

Figure 185. FSMC block diagram



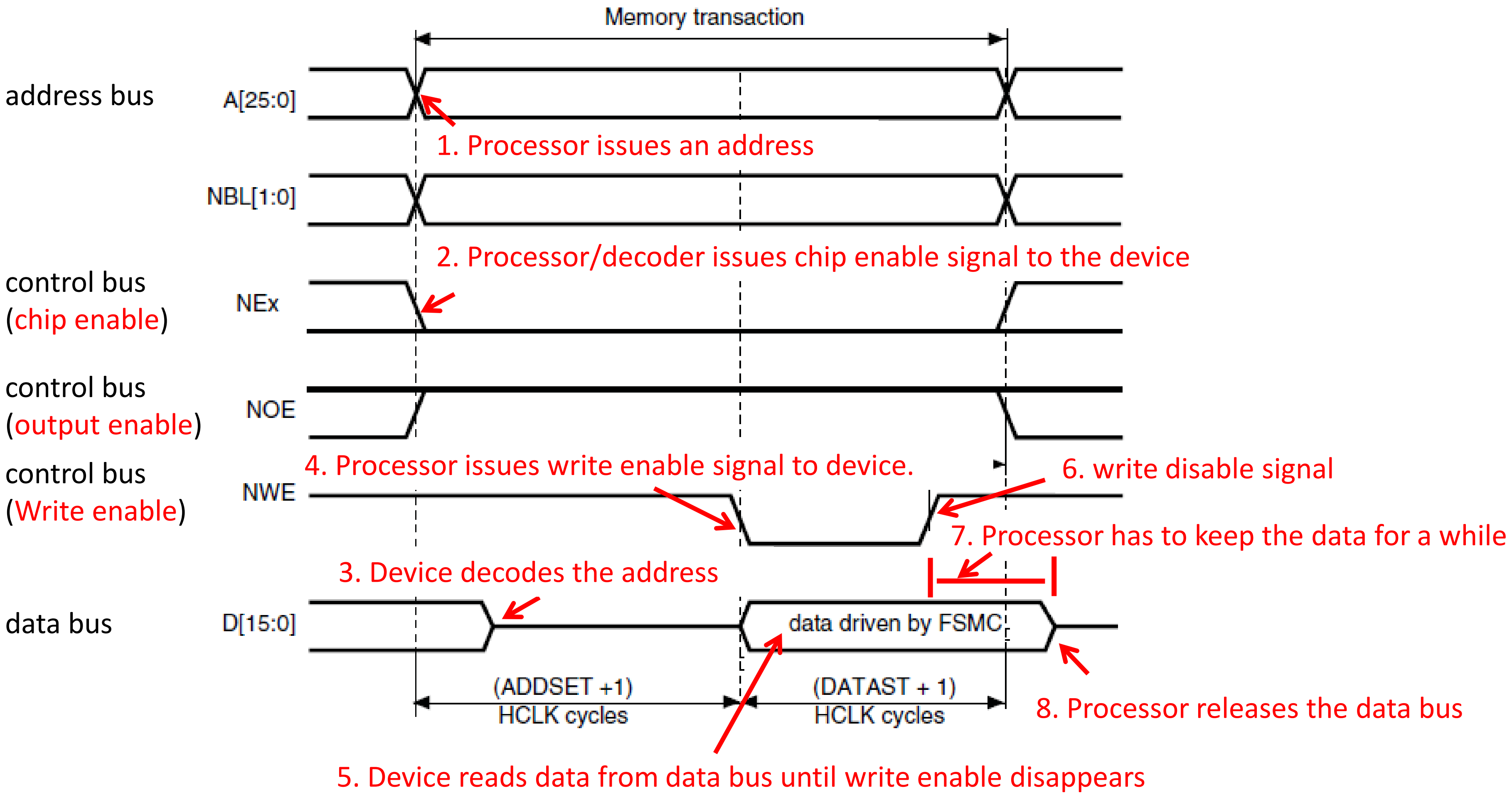
Memory Access – Read (Mode 1)

The Address bus is 26-bit in the figure, it will be introduced in lab-3.



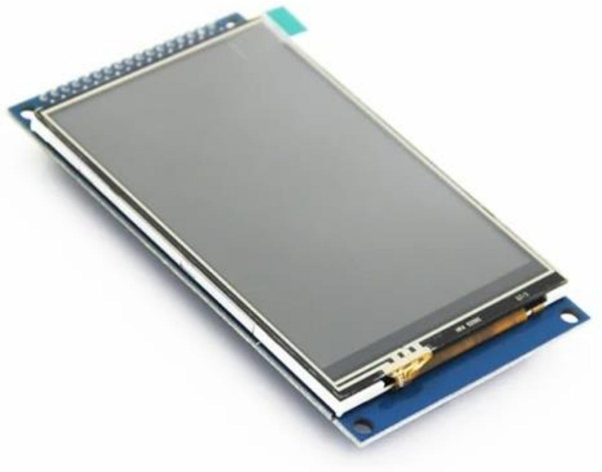
(Pictures from RM0008)

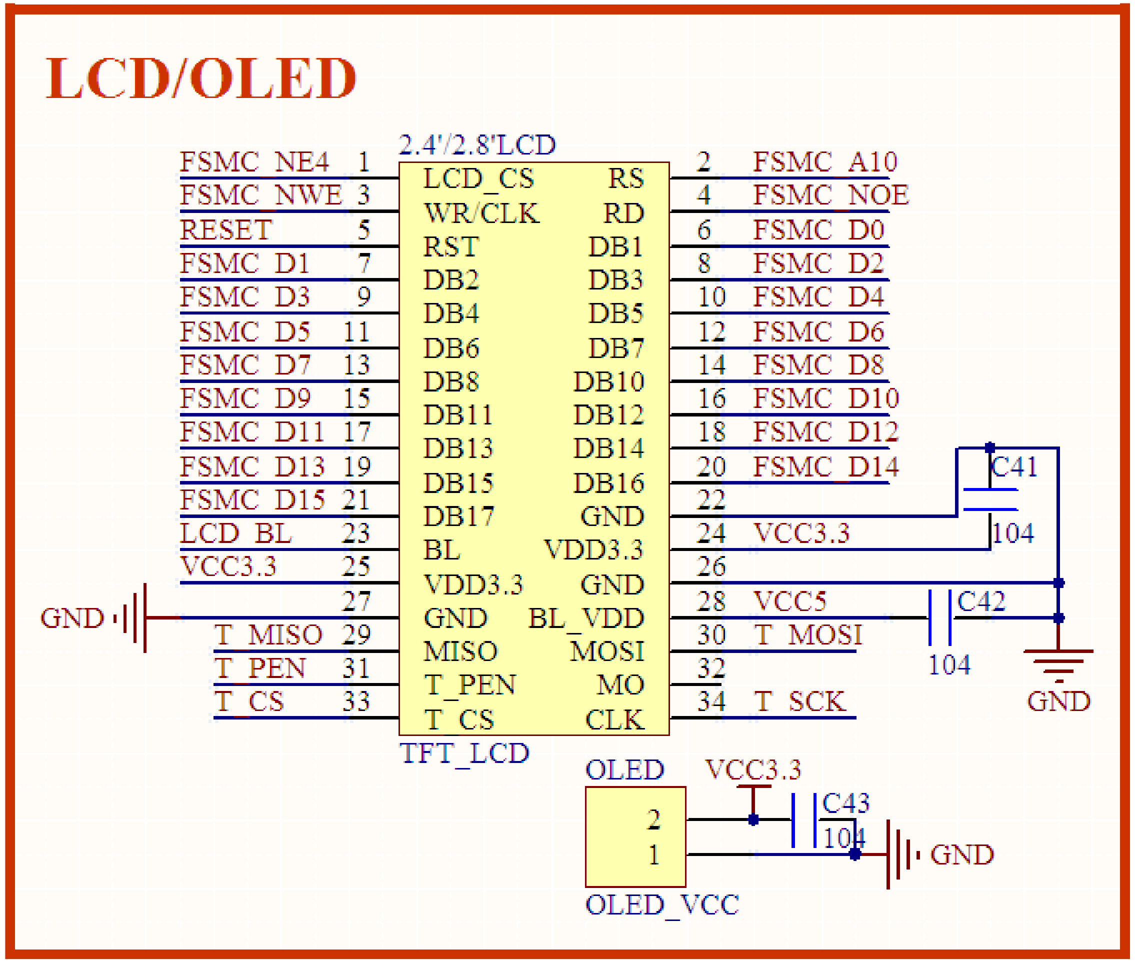
Memory Access – Write (Mode 1)



(Pictures from RM0008)

Example – LCD Control





Example – LCD Control

TFT-LCD pin	TFT-LCD function	STM32 pin AF	STM32 pin
LCD_BL	Backlight	PB0	PB0
LCD_CS	LCD chip select	FSMC_NE4	PG12
LCD_RS	0=command / 1=data selection	FSMC_A10	PG0
LCD_WR	LCD write	FSMC_NWE	PD5
LCD_RD	LCD read	FSMC_NOE	PD4
LCD_D[15:0]	LCD 16-bit data bus	FSMC_D15~FSMC_D0	PD14, PD15, PD0, PD1, PE7 ~ PE15, PD8 ~ PD10
LCD_RST	LCD reset	* Board hardware reset	

End