ELC4438: Embedded System Design

ARM Cortex-M Architecture II

Liang Dong

Electrical and Computer Engineering
Baylor University

Memory system

- The memory systems in microcontrollers often contain two or more types of memories:
  - flash memory for program code,
  - static RAM (SRAM) for data, and in some cases Electrical Erasable Read Only Memory (EEPROM).

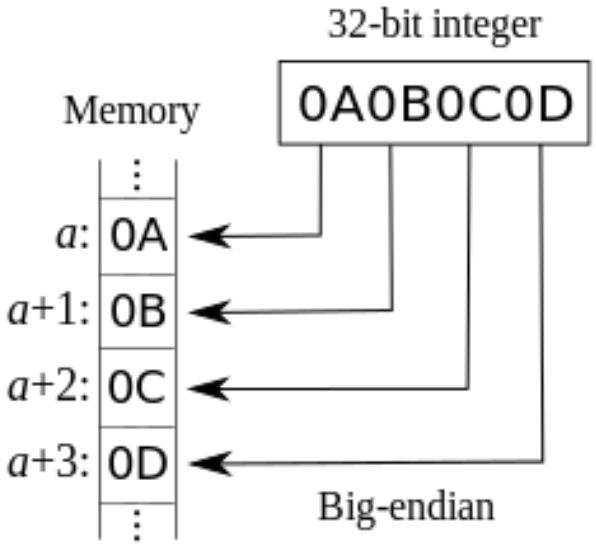
Memory system features

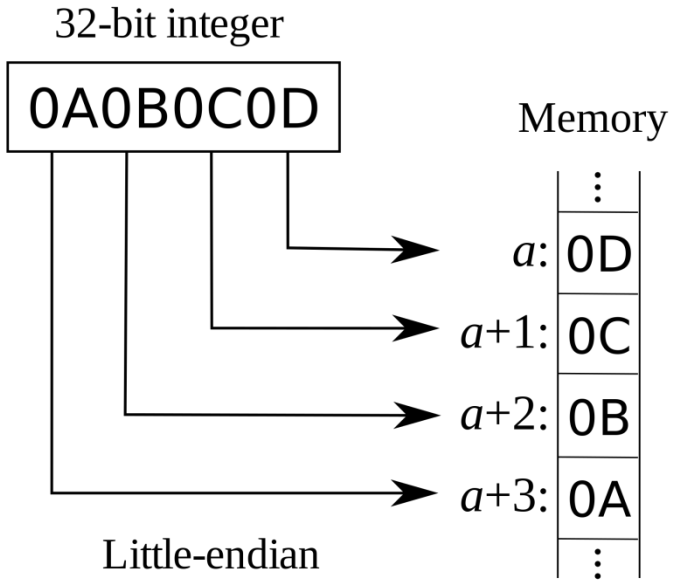
- 4GB linear address space
  - The Cortex-M3 and Cortex-M4 processors provide 32-bit buses using a generic bus protocol
  - allows connections to 32/16/8-bit memory devices with suitable memory interface controllers.

- Architecturally defined memory map
  - The 4GB memory space is divided into a number of regions for various predefined memory and peripheral uses
  - multiple bus interfaces to allow simultaneous access from the CODE

Memory system features

- Support for little endian and big endian memory systems





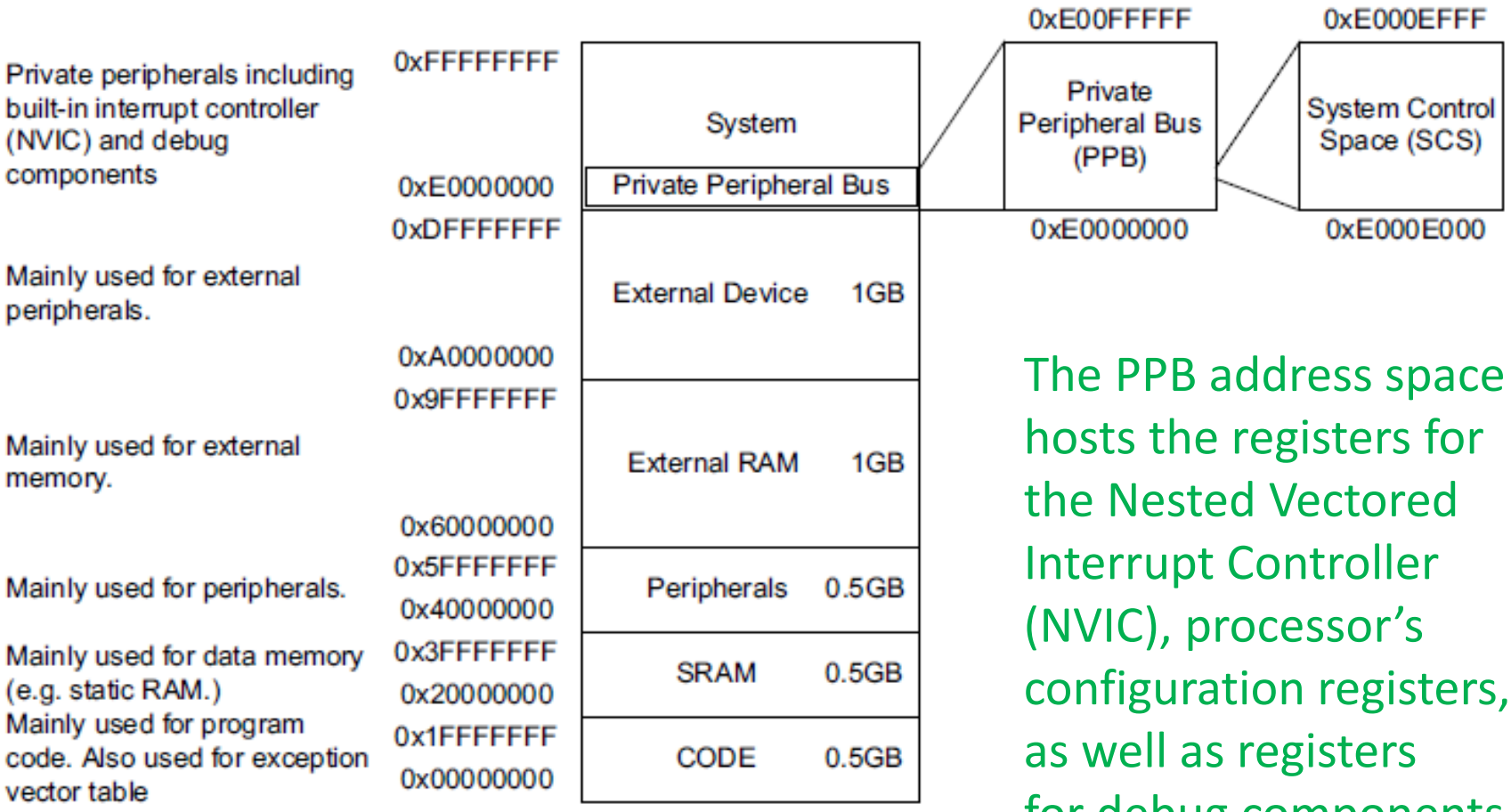
Memory system features

- Bit band accesses
  - Two 1MB regions in the memory map are bit addressable via two bit-band regions.
- Write buffer
- Memory Protection Unit
  - The MPU in the Cortex-M3 and Cortex-M4 processor supports eight programmable regions
- Unaligned transfer support

Memory map

- Program code accesses (e.g., CODE region)
- Data accesses (e.g., SRAM region)
- Peripherals (e.g., Peripheral region)
- Processor’s internal control and debug components (e.g., Private Peripheral Bus)

Memory map

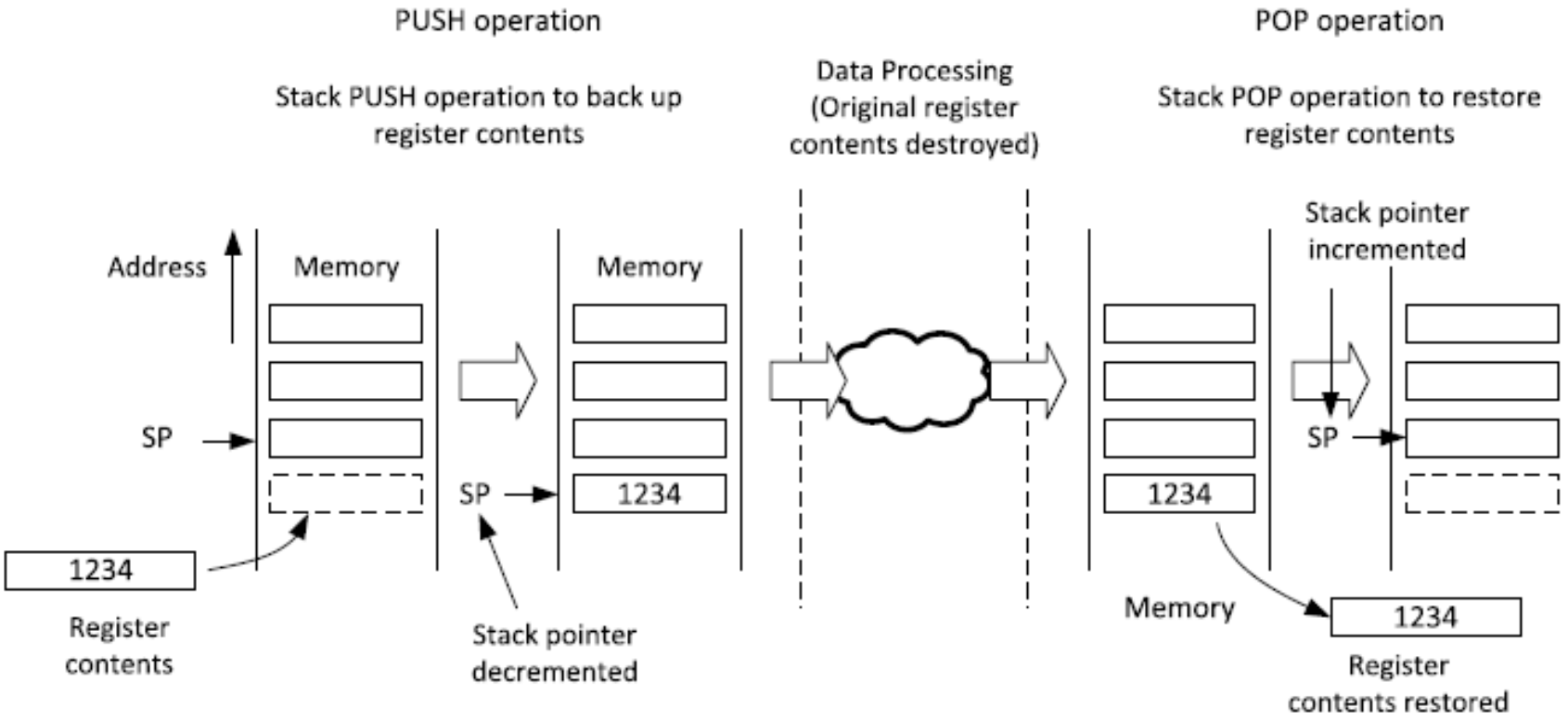


The PPB address space hosts the registers for the Nested Vectored Interrupt Controller (NVIC), processor's configuration registers, as well as registers for debug components.

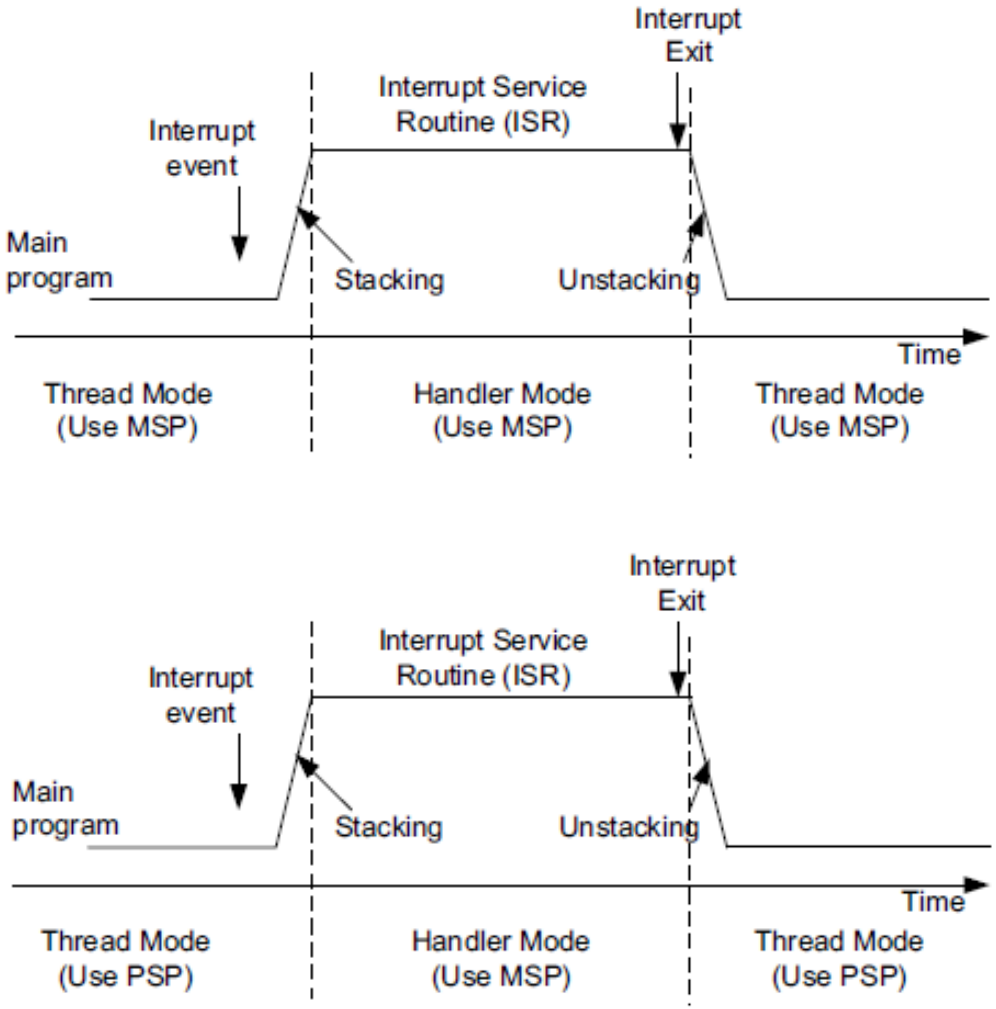
Stack memory

- Cortex-M processors need stack memory to operate and have stack pointers (R13).
- Allows a portion of memory to be used as Last-In-First-Out data storage buffer.
- The PUSH instruction to store data in stack and the POP instruction to retrieve data from stack

Full-descending stack



Two stack pointers



Exceptions and interrupts

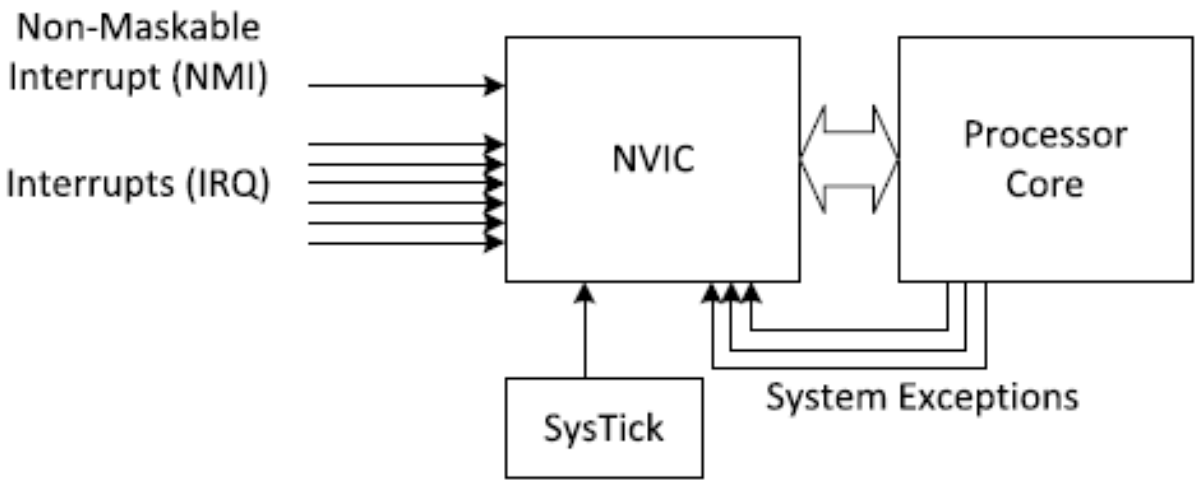
- Exceptions are events that cause changes to program flow.
  - suspends the current executing task and executes the exception handler
- Interrupts are usually generated from peripheral or external inputs, and in some cases they can be triggered by software.
- The exception handlers for interrupts - Interrupt Service Routines (ISR).

Exceptions and interrupts

- Usually IRQs are generated by on-chip peripherals or from external interrupt inputs though I/O ports.
- Inside the processor there is also a timer called SysTick, which can generate a periodic timer interrupt request.
- The processor itself is also a source of exception events.
  - Fault events
  - Software exceptions

Exceptions and interrupts

- Exceptions are processed by the NVIC.



Exceptions and interrupts

- Each exception source has an exception number.
- Exception numbers 1 to 15 are classified as system exceptions, and exceptions 16 and above are for interrupts.
- The design of the NVIC in the Cortex-M3 and Cortex-M4 processors can support up to 240 interrupt inputs. (in practice, 16 ~ 100)

Exceptions and interrupts

- Exception vectors are stored in a vector table
- The processor reads this table to determine the starting address of an exception handler
- The interrupt latency of the Cortex-M3 and Corex-M4 is very low - 12 clock cycles.

Nested vectored interrupt controller (NVIC)

- It is programmable and its registers are located in the System Control Space of the memory map.

- The NVIC handles the exceptions and interrupt configurations, prioritization, and interrupt masking.

Nested vectored interrupt controller (NVIC)

- Flexible exception and interrupt management
- Nested exception/interrupt support
- Vectored exception/interrupt entry
  - When an exception occurs, the processor will need to locate the starting point of the corresponding exception handler. The Cortex-M processors automatically locate the starting point of the exception handler from a vector table in the memory.
- Interrupt masking
  - the PRIMASK special register

Vector table

- The vector table is an array of word data inside the system memory, each representing the starting address of one exception type
- The vector table is relocatable
- It is controlled by a programmable register in the NVIC called the Vector Table Offset Register (VTOR).

Vector table

Exception Type	CMSIS Interrupt Number	Address Offset	Vectors
18 - 255	2 - 239	0x48 – 0x3FF	IRQ #2 - #239 [1]
17	1	0x44	IRQ #1 [1]
16	0	0x40	IRQ #0 [1]
15	-1	0x3C	SysTick [1]
14	-2	0x38	PendSV [1]
NA	NA	0x34	Reserved
12	-4	0x30	Debug Monitor [1]
11	-5	0x2C	SVC [1]
NA	NA	0x28	Reserved
NA	NA	0x24	Reserved
NA	NA	0x20	Reserved
NA	NA	0x1C	Reserved
6	-10	0x18	Usage fault [1]
4	-11	0x14	Bus Fault [1]
4	-12	0x10	MemManage Fault [1]
3	-13	0x0C	HardFault [1]
2	-14	0x08	NMI [1]
1	NA	0x04	Reset [1]
NA	NA	0x00	Initial value of MPS