第11次實習課

學生：林培瑋

2024 Advanced Mixed-Operation System (AMOS) Lab.

Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)

Hard fault

2024 Advanced Mixed-Operation System (AMOS) Lab.

Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)

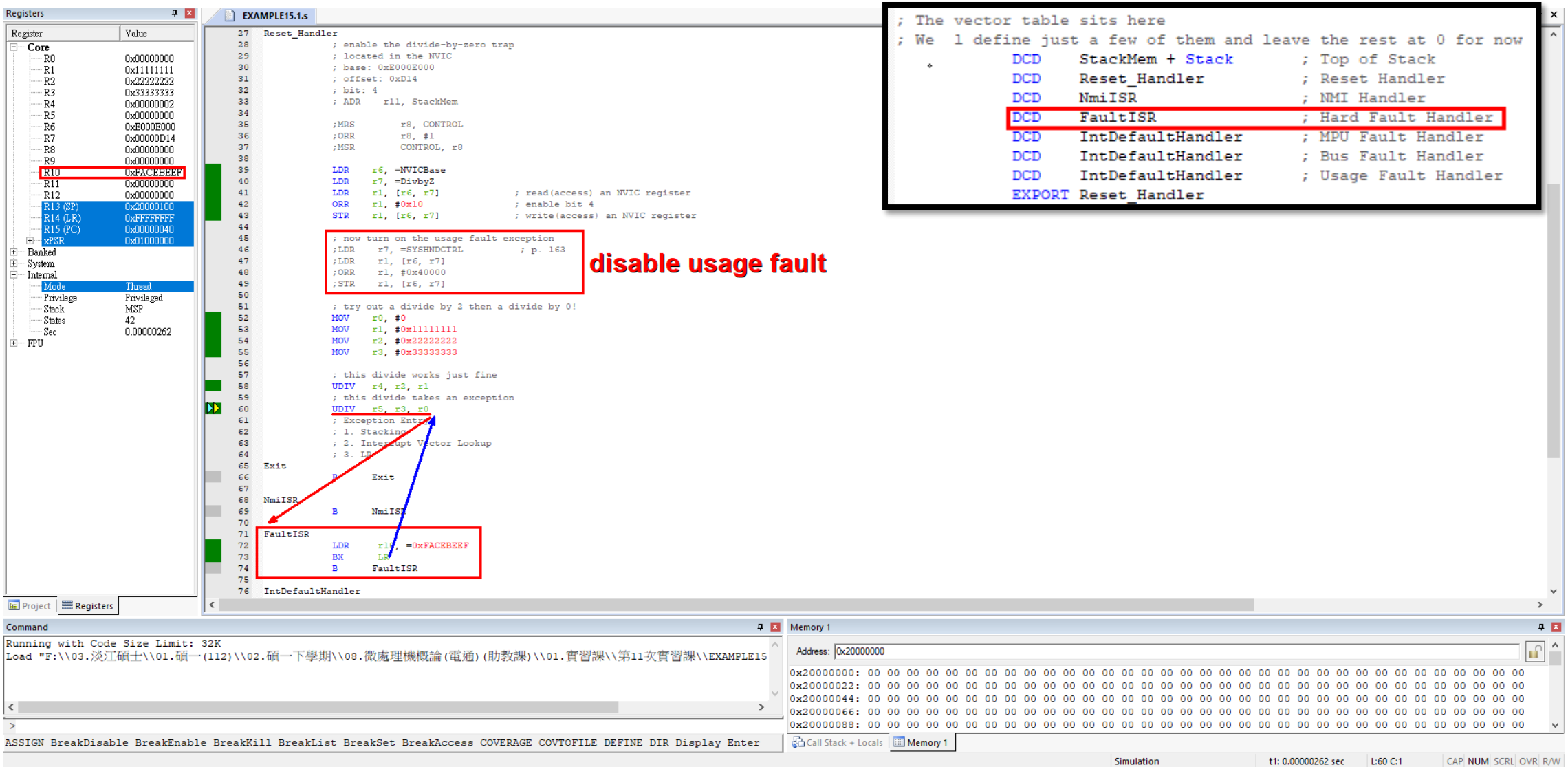
p.335 Hard fault & p.330 TABLE 15.1

A hard fault can occur when the processor sees an error during exception processing, or when another fault such as a usage fault is disabled.[1] In our example code, if we disable usage faults and then rerun the code, you will notice that the processor takes a hard fault when the UDIV instruction is attempted, rather than a usage fault. You can also see hard faults when there is an attempt to access the System Control Space in an unprivileged mode;[2] for example, if you attempt to write a value to one of the NVIC registers in Thread mode, the processor will take an exception.

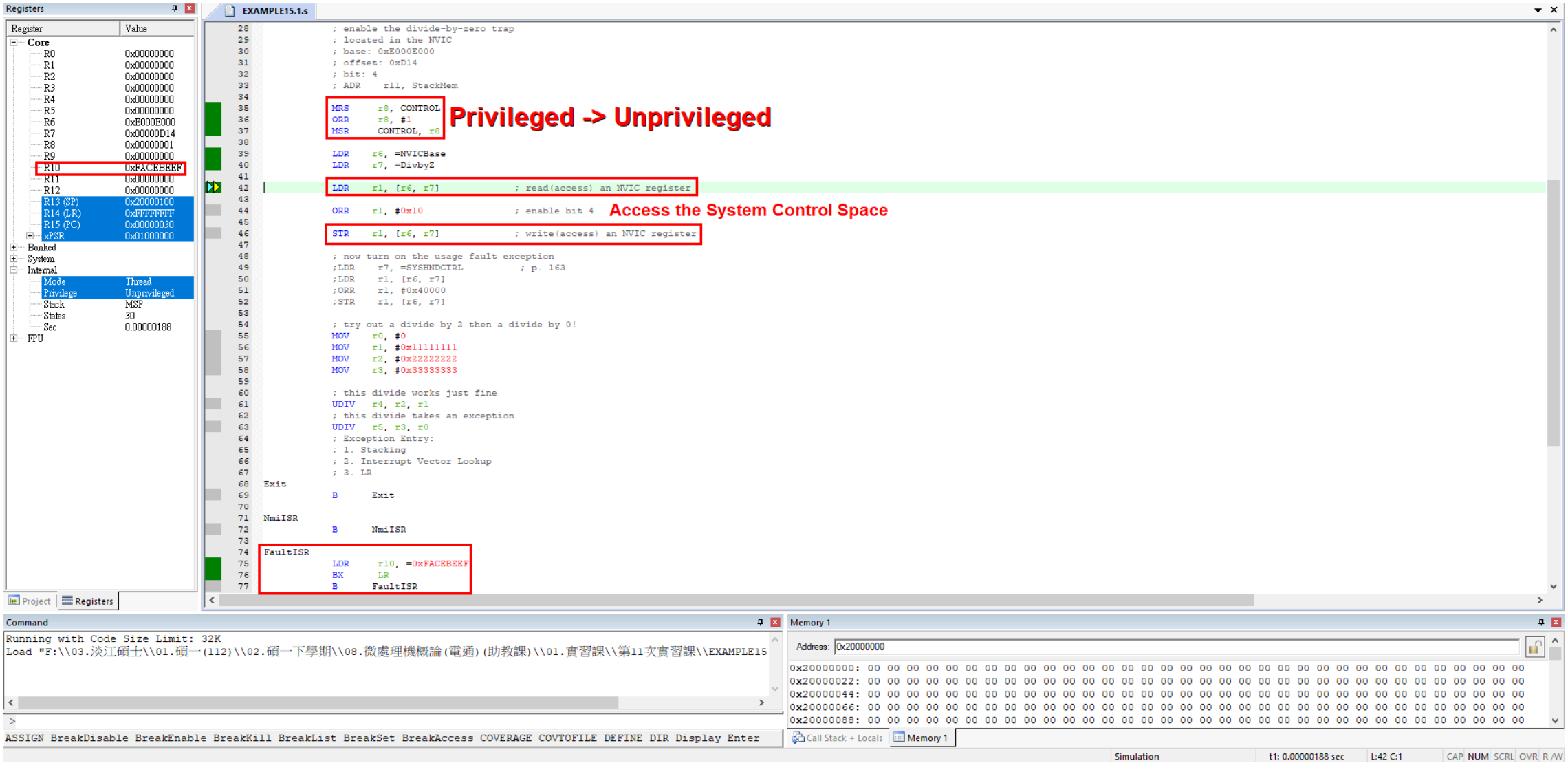
TABLE 15.1
Exception Types and Vector Table

Exception Type	Exception Number	Priority	Vector Address	Caused by...
—	—	—	0x00000000	Top of stack
Reset	1	– 3 (highest)	0x00000004	Reset
NMI	2	– 2	0x00000008	Non-maskable interrupt
Hard fault	3	– 1	0x0000000C	All fault conditions if the corresponding fault is not enabled
Mem mgmt fault	4	Programmable	0x00000010	MPU violation or attempted access to illegal locations
Bus fault	5	Programmable	0x00000014	Bus error, which occurs during AHB transactions when fetching instructions or data
Usage fault	6	Programmable	0x00000018	Undefined instructions, invalid state on instruction execution, and errors on exception return
—	7–10	—		Reserved
SVcall	11	Programmable	0x0000002C	Supervisor Call
Debug monitor	12	Programmable	0x00000030	Debug monitor requests such as watchpoints or breakpoints
—	13	—		Reserved
PendSV	14	Programmable	0x00000038	Pendable Service Call
SysTick	15	Programmable	0x0000003C	System Tick Timer
Interrupts	16 and above	Programmable	0x00000040 and above	Interrupts

1. Usage fault is disabled (Keil Tool)



2. Attempt to access the System Control Space in an unprivileged mode (Keil Tool)



Priority

2024 Advanced Mixed-Operation System (AMOS) Lab.

Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)

p.330 TABLE 15.1 Priority

FIFO queue(一般型, 先進先出)

CPU ← NVIC ⇐ peripheral chips

TABLE 15.1 若使用FIFO queue，重開機則會等待一段時間才去做。

Exception Types and Vector Table

(lower) priority value -> priority level (higher)

	Exception Type	Exception Number	Priority	Vector Address	Caused by…
	—	—	—	0x00000000	Top of stack
Fixed priority	Reset	1	– 3 (highest)	0x00000004	Reset
	NMI	2	– 2	0x00000008	Non-maskable interrupt 不可忽略
	Hard fault	3	– 1	0x0000000C	All fault conditions if the corresponding fault is not enabled
Programmable priority (changeable) (>=0)	Mem mgmt fault	4	Programmable	0x00000010	MPU violation or attempted access to illegal locations
	Bus fault	5	Programmable	0x00000014	Bus error, which occurs during AHB transactions when fetching instructions or data
	Usage fault	6	Programmable	0x00000018	Undefined instructions, invalid state on instruction execution, and errors on exception return
software, system exceptions	—	7–10	—		Reserved
	SVcall	11	Programmable	0x0000002C	Supervisor Call
	Debug monitor	12	Programmable	0x00000030	Debug monitor requests such as watchpoints or breakpoints
	—	13	—		Reserved
	PendSV	14	Programmable	0x00000038	Pendable Service Call
	SysTick	15	Programmable	0x0000003C	System Tick Timer
hardware interrupts	Interrupts	16 and above (16~255, 240 interrupts)	Programmable	0x00000040 and above	Interrupts

EX：install driver

get an interrupt number (exception number)

install program in main memory vector

inerrupt number

高優先權搶佔低優先權的位置

priority queue

1. who will be served first
2. who will be interrupted (preempted 搶佔)

Priority value of an interrupt is stored in a corresponding 8-bit Interrupt Priority Register (Priority Level Register)in NVIC.

00000000~11111111 256 interrupt priority values

16 priority values — **4 bits of Interrupt Priotity Register 0000~1111 (4 implemented bits, 4 not implemented bits-always 0)**

8 priority values — **3 bits of Interrupt Priotity Register 000~111 (3 implemented bits, 5 not implemented bits-always 0)**

MSB(Most Significant Bit) first implemented
LSB(Least Significant Bit) first implemented

MSB first vs. LSB first

LSB → 00000000~00000111
MSB → 00000000~11100000

如果從16 → 8，
考慮Modularity(有彈性)、Reconfiguration(可再造的)，
決定哪一個是最佳的。

p.330 TABLE 15.1 Priority

- **LSB**：
  - 16 priority values ➔ 4 bits
  - 0000**1011** (value 11, **lower** level)
  - 0000**0111** (value 7, **higher** level)

  ➔reconfigure to 8 priority values only ➔ 3 bits
  - 00000**011** (value 2, **higher** level) ➔ **priority inversion**
  - 00000**111** (value 7, **lower** level)
- **MSB**：
  - 16 priority values ➔ 4 bits
  - **1011**0000 (value 0xB0, **lower** level)
  - **0111**0000 (value 0x70, **higher** level)

  ➔reconfigure to 8 priority values only ➔ 3 bits
  - **101**00000 (value 0xA0, **lower** level) ➔ **No priority inversion**
  - **011**00000 (value 0x60, **higher** level)

https://developer.arm.com/documentation/dui0552/a/cortex-m3-peripherals/nested-vectored-interrupt-controller/interrupt-priority-registers

Q&A

Thanks for your attention !!