

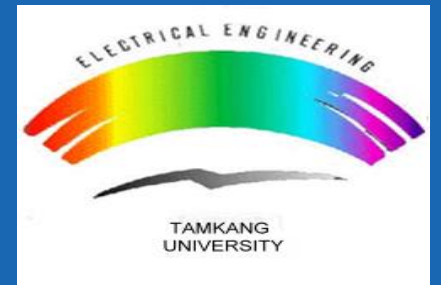
第08次實習課

學生：林培瑋

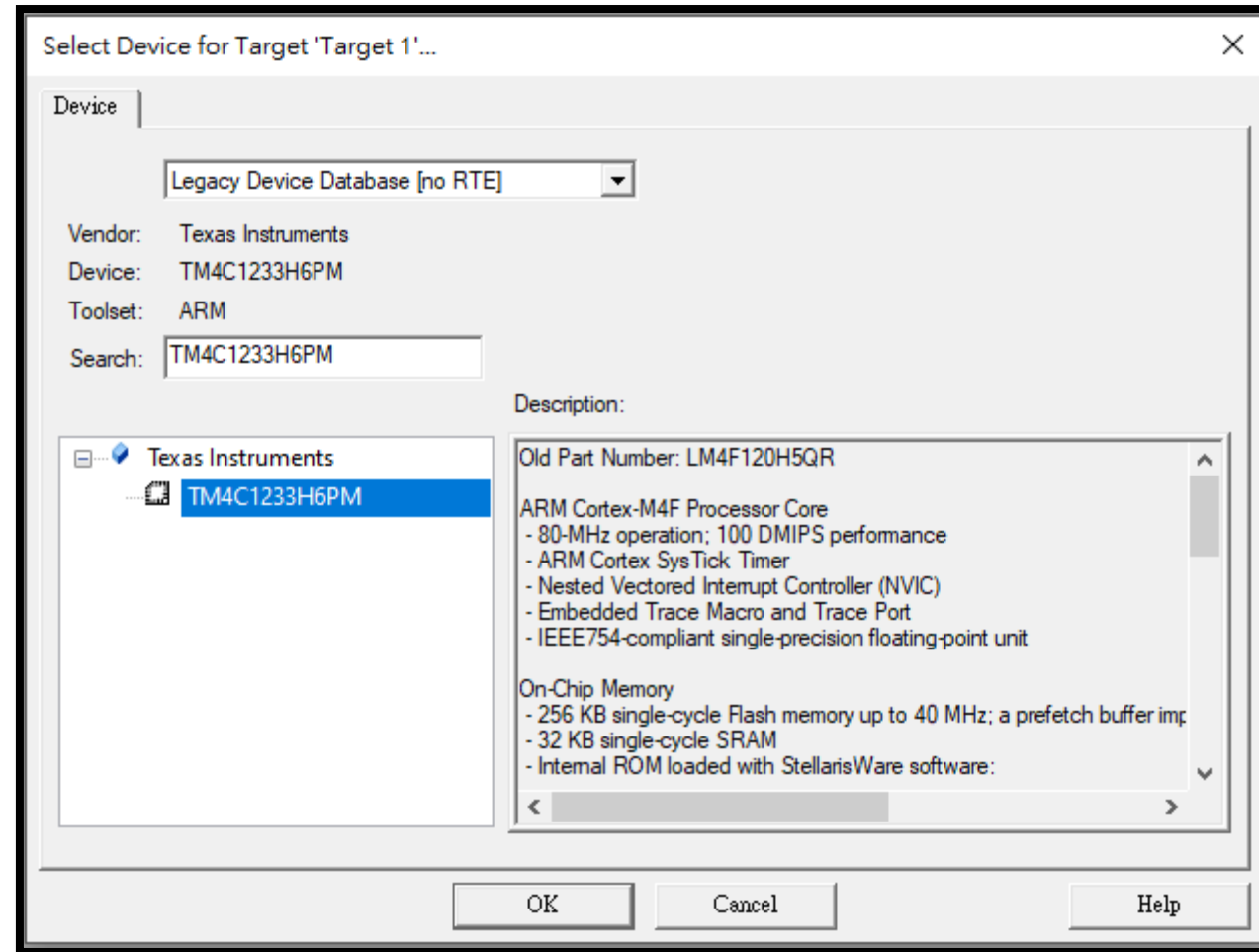
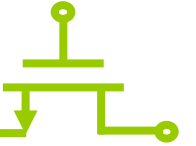
2024 Advanced Mixed-Operation System (AMOS) Lab.

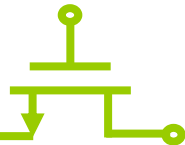


Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)



EXAMPLE 15.1





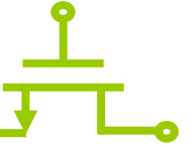
```

; enable the divide-by-zero trap
; located in the NVIC
; base: 0xE000E000
; offset: 0xD14
; bit: 4
LDR    r6, =NVICBase
LDR    r7, =DivbyZ
LDR    r1, [r6, r7]
ORR    r1, #0x10                ; enable bit 4
STR    r1, [r6, r7]
; now turn on the usage fault exception
LDR    r7, =SYSHNDCTRL          ; p. 163
LDR    r1, [r6, r7]
ORR    r1, #0x40000
STR    r1, [r6, r7]
    
```

Reset_Handler

```

; 啟用除零陷阱
; 在 NVIC 中設置
; 基址: 0xE000E000
; 偏移: 0xD14
; 位: 4
LDR    r6, =NVICBase           ; 將 NVIC 的基址載入 r6 中
LDR    r7, =DivbyZ             ; 將 DivbyZ 的地址載入 r7 中
LDR    r1, [r6, r7]           ; 將 NVICBase + DivbyZ 地址處的值讀入 r1 中
ORR    r1, #0x10               ; 啟用第 4 位
STR    r1, [r6, r7]           ; 將修改後的值寫入 NVICBase + DivbyZ 的地址處
; 現在啟用使用異常處理器
LDR    r7, =SYSHNDCTRL         ; 參見第 163 頁
LDR    r1, [r6, r7]           ; 將 SYSHNDCTRL 的地址載入 r7 中
ORR    r1, #0x40000           ; 設置第 18 位以啟用使用異常處理器
STR    r1, [r6, r7]           ; 將修改後的值寫入 SYSHNDCTRL 的地址處
    
```



Configuration Control Register

Read/write

0xE000ED14

0x00000000

Configuration Control Register

Configuration Control Register

Use the **Configuration Control Register** to:

- enable NMI, Hard Fault and FAULTMASK to ignore bus fault
- trap divide by zero, and unaligned accesses
- enable user access to the Software Trigger Exception Register
- control entry to Thread Mode.

The register address, access type, and Reset state are:

Address

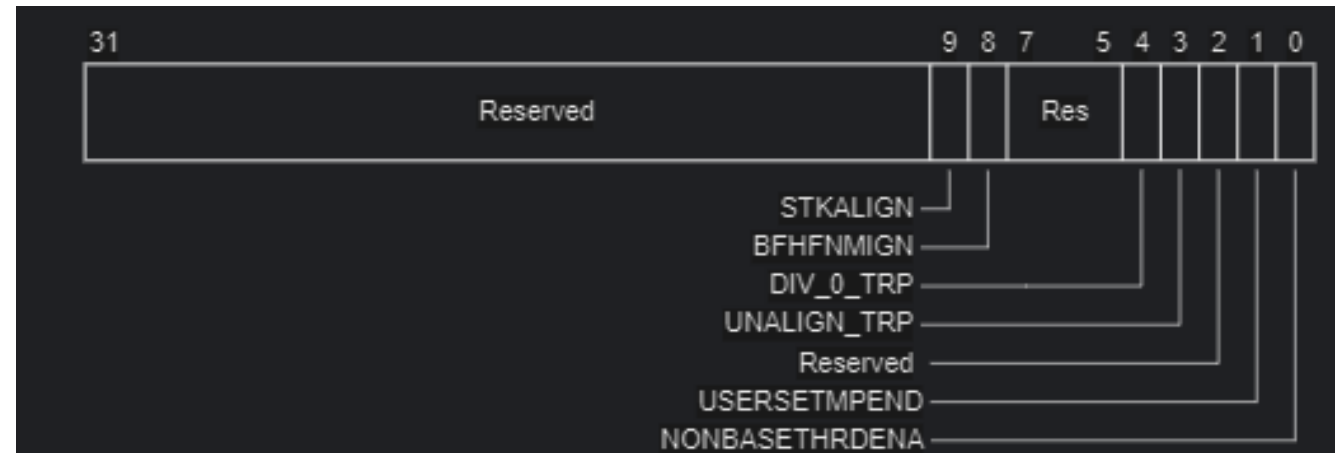
0xE000ED14

Access

Read/write

Reset state

0x00000000



[4] DIV_0_TRP

Trap on Divide by 0. This enables faulting/halting when an attempt is made to divide by 0. The relevant Usage Fault Status Register bit is DIVBYZERO, see [Usage Fault Status Register](#).



TABLE 15.1 Usage fault

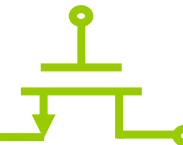


TABLE 15.1

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, <u>invalid state on instruction execution</u> , 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

參考NVIC技術手冊(0xE000ED24)



System Handler Control and State Register

Read/write

0xE000ED24

0x00000000

System Handler Control and State Register

System Handler Control and State Register

Use the **System Handler Control and State Register** to:

- enable or disable the system handlers
- determine the pending status of bus fault, mem manage fault, and SVC
- determine the active status of the system handlers.

If a fault condition occurs while its fault handler is disabled, the fault escalates to a Hard Fault.

The register address, access type, and Reset state are:

Address

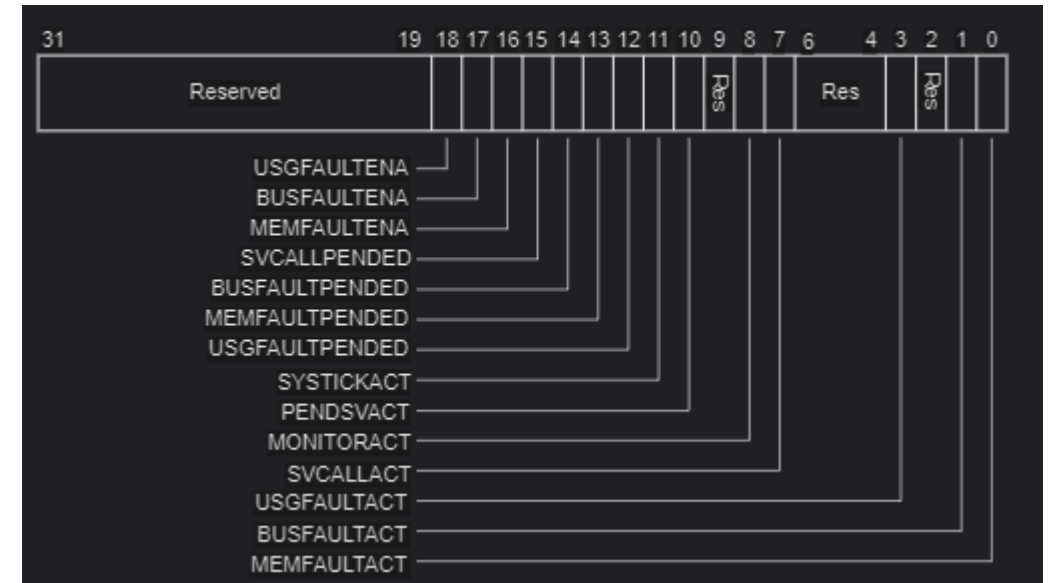
0xE000ED24

Access

Read/write

Reset state

0x00000000

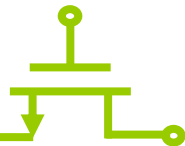


[18]

USGFAULTENA

Set to 0 to disable, else 1 for enabled.

P332 查找StackMem初始位置



Project: EXAMPLE15.1
Target 1
Source Group 1

```

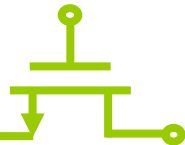
13          THUMB
14
15; The vector table sits here
16; We 1 define just a few of them and leave the rest at 0 for now
17          DCD      StackMem + Stack      ; Top of Stack
18          DCD      Reset_Handler         ; Reset Handler
19          DCD      NmiISR                 ; NMI Handler
20          DCD      FaultISR              ; Hard Fault Handler
21          DCD      IntDefaultHandler     ; MPU Fault Handler
22          DCD      IntDefaultHandler     ; Bus Fault Handler
23          DCD      IntDefaultHandler     ; Usage Fault Handler
24          EXPORT   Reset_Handler
25          ENTRY
26
27Reset_Handler
28          ; enable the divide-by-zero trap
29          ; located in the NVIC
30          ; base: 0xE000E000
31          ; offset: 0xD14
32          ; bit: 4
33          ADR      r11, StackMem
34          LDR      r6, =NVICBase
35          LDR      r7, =DivbyZ
36          LDR      r1, [r6, r7]
37          ORR      r1, #0x10              ; enable bit 4
38          STR      r1, [r6, r7]
39          ; now turn on the usage fault exception
40          LDR      r7, =SYSHNDCTRL       ; p. 163
41          LDR      r1, [r6, r7]
42          ORR      r1, #0x40000

```

memory. On the Tiva TM4C1233H6PM microcontroller, SRAM begins at address **0x20000000** and the stack has been defined to be 256 bytes (0x100) at the top of our program. If you look at memory starting just below 0x20000100, you will notice that the contents of registers r0 through r3, register r12, the Link Register, the PC, and the contents of the xPSR have been moved onto the stack, shown in Figure 15.5. Recall that the stack pointer indicates the address of the last full entry, so stacking would begin at address 0x200000FC.

Build Output
EXAMPLE15.1.s(33): error: A1898E: Target cannot be relocated. No suitable relocation exists for this instruction
".\Objects\EXAMPLE15.1" - 1 Error(s), 0 Warning(s).
Target not created.
Build Time Elapsed: 00:00:00

EXAMPLE15.1-Keil Tool-StackMem



Registers

Register	Value
R0	0x00000000
R1	0x11111111
R2	0x22222222
R3	0x33333333
R4	0x00000002
R5	0x00000000
R6	0xE000E000
R7	0x0000D2A
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x200000E0
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000056
xPSR	0x01000006

Banked
System
Internal
Mode Handler
Privilege Privileged
Stack MSP
States 35
Sec 0.00000219
FPU

Disassembly

```

67: LDR r7, =Usagefault
0x00000056 F640572A MOVW r7, #0xD2A
68: LDRH r1, [r6, r7]
0x0000005A SBF1 LDRH r1, [r6, r7]
69: TEQ r1, #0x200
    
```

EXAMPLE15.1.s

```

49 ; this divide works just fine
50 UDIV r4, r2, r1
51 ; this divide takes an exception
52 UDIV r5, r3, r0
53 ; Exception Entry:
54 ; 1. Stacking
55 ; 2. Interrupt Vector Lookup
56 Exit
57 B Exit
58
59 NmiISR
60 B NmiISR
61
62 FaultISR
63 B FaultISR
64
65 IntDefaultHandler
66 ; let read the Usage Fault Status Register
67 LDR r7, =Usagefault
68 LDRH r1, [r6, r7]
    
```

Memory 1

Address: 0x20000000

0x20000070:	00 00
0x2000008C:	00 00
0x200000A8:	00 00
0x200000C4:	00 00
0x200000E0:	00 00 00 00 11 11 11 11 22 22 22 22 33 33 33 33 00 00 FF FF FF FF 4C 00 00 00 00 00 00 00 00 00 00
0x200000FC:	00 00 00 01 00
0x20000118:	00 00

StackMem

Low address

Full Descending

256-byte stack

High Address

Command

Running with Code Size Limit: 32K

Load "E:\03.淡江碩士\01.碩一(112)\02.碩一下學期\08.微處理機概論(電通)(助教課)\01.實習課\第0

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE

Simulation

t1: 0.00000219 sec L:67 C:1 CAP NUM SCRL OVR R/W

R0	0x00000000
R1	0x11111111
R2	0x22222222
R3	0x33333333
R4	0x00000002
R5	0x00000000
R6	0xE000E000
R7	0x00000D24
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000

New SP,
8-byte aligned

AAPCS : Arm Architecture Program Call Standard
Caller Saved registers
Callee Saved registers

P336 TABLE15.4對照InDefaultHandler

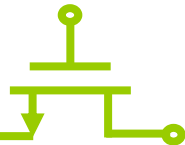


TABLE 15.4

Usage Fault Status Register (Offset 0xD2A)

Bit	Name	Reset Value	Description
9	DIVBYZERO	0	Indicates a divide by zero has occurred (only if DIV_0_TRP is also set)
8	UNALIGNED	0	An unaligned access fault has occurred
7:4	—	—	—
3	NOCP	0	Indicates a coprocessor instruction was attempted
2	INVPC	0	An invalid EXC_RETURN value was used in an exception
1	INVSTATE	0	An attempt was made to switch to an invalid state
0	UNDEFINSTR	0	Processor tried to execute an undefined instruction

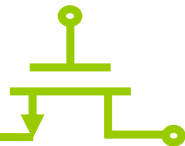
```

IntDefaultHandler
; let    read the Usage Fault Status Register
LDR     r7, =Usagefault
LDRH    r1, [r6, r7]
TST     r1, #0x200
IT      NE
; IT:If NE Then
; ITT
; ITTE
; ITTEE:If Then 2 Else 2
LDRNE   r9, =0xDEADDEAD
; MOVNE r10, r9
; LDRNE r11, =FACEBEEF
; MOVEQ r12, r11
    
```

讓Assembler先知道後面的組織，
增加程式執行效率。(可有可無)

T：正面；E：負面
最多接4個

EXAMPLE 15.1-Enable the divide-by-zero trap



Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000100
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000001C
xPSR	0x01000000

- Banked
- System
- Internal
 - Mode
 - Privilege
 - Stack
 - States
 - Sec
- FPU

```

34:      LDR    r6, =NVICBase
0x0000001C F04F26E0  MOV    r6, #0xE000E000
35:      LDR    r7, =DivbyZ
0x00000020 F6405714  MOVW   r7, #0xD14
36:      LDR    r1, [r6, r7]

```

EXAMPLE15.1.s

```

28      ; enable the divide-by-zero trap
29      ; located in the NVIC
30      ; base: 0xE000E000
31      ; offset: 0xD14
32      ; bit: 4
33      ; ADR    r11, StackMem
34      LDR    r6, =NVICBase
35      LDR    r7, =DivbyZ
36      LDR    r1, [r6, r7]
37      ORR    r1, #0x10      ; enable bit 4
38      STR    r1, [r6, r7]
39      ; now turn on the usage fault exception
40      LDR    r7, =SYSHNDCTRL ; p. 163
41      LDR    r1, [r6, r7]
42      ORR    r1, #0x40000
43      STR    r1, [r6, r7]
44      ; try out a divide by 2 then a divide by 0!
45      MOV    r0, #0
46      MOV    r1, #0x11111111
47      MOV    r2, #0x22222222

```

Project

Registers

Command

Running with Code Size Limit: 32K
Load "E:\03.淡江碩士\01.碩一(112)\02.碩一下學期\08.微處理機概論(電通)(助教課)\01.實習課\第0

Memory 1

Address: 0xE000ED14

0xE000ED14:	00 02 00 00	00 00
0xE000ED30:	00 00 00 00	00 00
0xE000ED4C:	00 00 00 00	00 00
0xE000ED68:	31 20 23 20	31 11 11 01 02 01 31 01 00
0xE000ED84:	00 00 00 00	00 00
0xE000EDA0:	00 00 00 00	00 00
0xE000EDBC:	00 00 00 00	00 00

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE

Call Stack + Locals

Memory 1

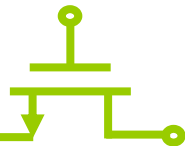
Simulation

t1: 0.00000000 sec

L:34 C:1

CAP NUM SCRL OVR R/W

EXAMPLE 15.1-Enable the divide-by-zero trap



Register	Value
R0	0x00000000
R1	0x00000210
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0xE000E000
R7	0x0000D14
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000100
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000002C
xPSR	0x01000000

Banked

System

Internal

Mode

Privilege

Stack

States

Sec

Thread

Privileged

MSP

7

0.00000044

FPU

Disassembly

```

40:      LDR    r7, =SYSHNDCTRL      ; p. 163
0x0000002C F6405724  MOVW    r7, #0xD24
41:      LDR    r1, [r6, r7]
0x00000030 59F1      LDR    r1, [r6, r7]
42:      ORR    r1, #0x40000

```

EXAMPLE15.1.s

```

34      LDR    r6, =NVICBase
35      LDR    r7, =DivbyZ
36      LDR    r1, [r6, r7]
37      ORR    r1, #0x10             ; enable bit 4
38      STR    r1, [r6, r7]
39      ; now turn on the usage fault exception
40      LDR    r7, =SYSHNDCTRL      ; p. 163
41      LDR    r1, [r6, r7]
42      ORR    r1, #0x40000
43      STR    r1, [r6, r7]
44      ; try out a divide by 2 then a divide by 0!
45      MOV    r0, #0
46      MOV    r1, #0x11111111
47      MOV    r2, #0x22222222
48      MOV    r3, #0x33333333
49      ; this divide works just fine
50      UDIV   r4, r2, r1
51      ; this divide takes an exception
52      UDIV   r5, r3, r0
53      ; Exception Entry:

```

Command

Running with Code Size Limit: 32K
Load "E:\03.淡江碩士\01.碩一(112)\02.碩一下學期\08.微處理機概論(電通)(助教課)\01.實習課\第0

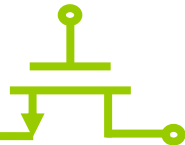
Memory 1

Address: 0xE000ED14
0xE000ED14: 10 02 00
0xE000ED30: 00
0xE000ED4C: 00 10 11 14 01 00 20 11 02
0xE000ED68: 31 20 23 20 31 11 11 01 02 01 31 01 00
0xE000ED84: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xE000EDA0: 00
0xE000EDBC: 00

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE

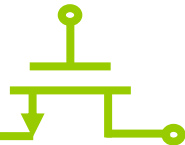
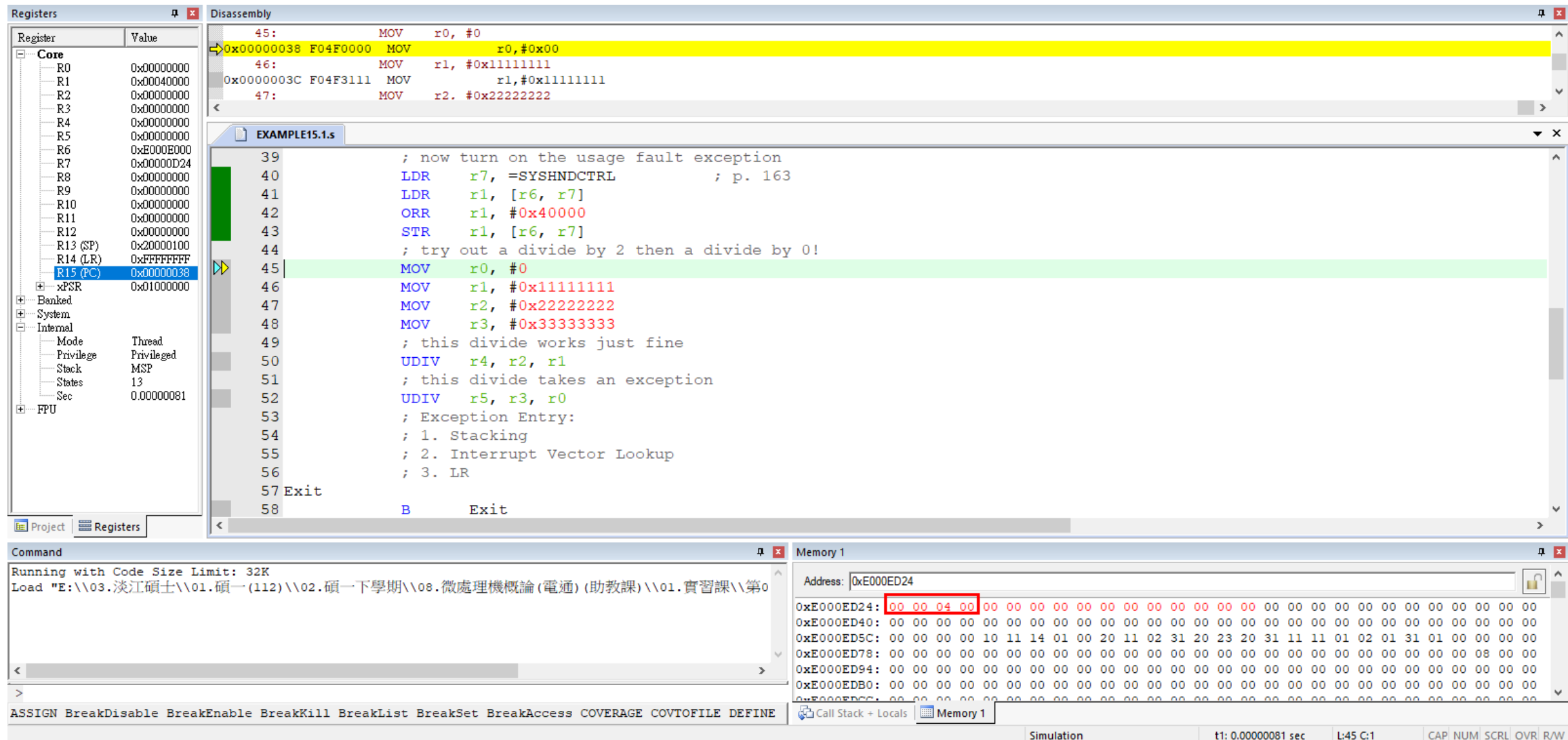
Call Stack + Locals Memory 1

Simulation t1: 0.00000044 sec L:40 C:1 CAP NUM SCRL OVR R/W



P. W. LIN

EXAMPLE 15.1-Turn on the usage fault exception

The screenshot displays the Keil uVision IDE interface during a simulation. The main window shows the disassembly of a program named 'EXAMPLE15.1.s'. The assembly code is as follows:

```

45:      MOV     r0, #0
0x00000038 F04F0000 MOV     r0, #0x00
46:      MOV     r1, #0x11111111
0x0000003C F04F3111 MOV     r1, #0x11111111
47:      MOV     r2, #0x22222222

```

The program then enters a loop where it attempts to divide by zero, which triggers a usage fault exception:

```

39      ; now turn on the usage fault exception
40      LDR     r7, =SYSHNDCTRL      ; p. 163
41      LDR     r1, [r6, r7]
42      ORR     r1, #0x40000
43      STR     r1, [r6, r7]
44      ; try out a divide by 2 then a divide by 0!
45      MOV     r0, #0
46      MOV     r1, #0x11111111
47      MOV     r2, #0x22222222
48      MOV     r3, #0x33333333
49      ; this divide works just fine
50      UDIV    r4, r2, r1
51      ; this divide takes an exception
52      UDIV    r5, r3, r0
53      ; Exception Entry:
54      ; 1. Stacking
55      ; 2. Interrupt Vector Lookup
56      ; 3. LR
57      Exit
58      B       Exit

```

The 'Registers' window on the left shows the state of the processor registers. The 'Command' window at the bottom shows the command 'Load "E:\03.淡江碩士\01.碩一(112)\02.碩一下學期\08.微處理機概論(電通)(助教課)\01.實習課\第0'.

The 'Memory' window at the bottom right shows the memory dump starting at address 0xE000ED24. The first four bytes of the memory dump are highlighted in red, indicating the exception entry point.

EXAMPLE 15.1-Final



Registers

Register	Value
R0	0x00000000
R1	0x00000200
R2	0x22222222
R3	0x33333333
R4	0x00000002
R5	0x00000000
R6	0xE000E000
R7	0x0000D2A
R8	0x00000000
R9	0xDEADDEAD
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x200000E0
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000066
xPSR	0x01000006

Disassembly

84: B done

0x00000066 E7FE B 0x00000066

0x00000068 DEAD DCW 0xDEAD

0x0000006A DEAD DCW 0xDEAD

0x0000006C 0000 MOVS r0,r0

EXAMPLE15.1.s

```

71      IT      NE
72      ; IT:If NE Then
73      ; ITT
74      ; ITTE
75      ; ITTEE:If Then 2 Else 2
76      LDRNE   r9, =0xDEADDEAD
77      ; MOVNE r10, r9
78      ; LDRNE r11, =FACEBEEF
79      ; MOVEQ r12, r11
80
81      ; r1 should have bit 9 set indicating
82      ; a divide-by-zero has taken place
83 done
84      B       done
85
86      ALIGN
87
88      END
89

```

Command

Running with Code Size Limit: 32K

Load "E:\03.淡江碩士\01.碩一(112)\02.碩一下學期\08.微處理機概論(電通)(助教課)\01.實習課\第0

Memory 1

Address: 0xE000ED24

0xE000ED24:	08 00 04 00 00 00 00 02 00
0xE000ED40:	00 00
0xE000ED5C:	00 00 00 00 10 11 14 01 00 20 11 02 31 20 23 20 31 11 11 01 02 01 31 01 00 00 00 00 00 00 00 00
0xE000ED78:	00 08 00 00 00
0xE000ED94:	00 00
0xE000EDB0:	00 00
0xE000EDCC:	00 00

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE

Simulation

t1: 0.00000262 sec

L:84 C:1

CAP NUM SCRL OVR R/W

EXAMPLE 15.3 : Exception Entry→1. Stacking

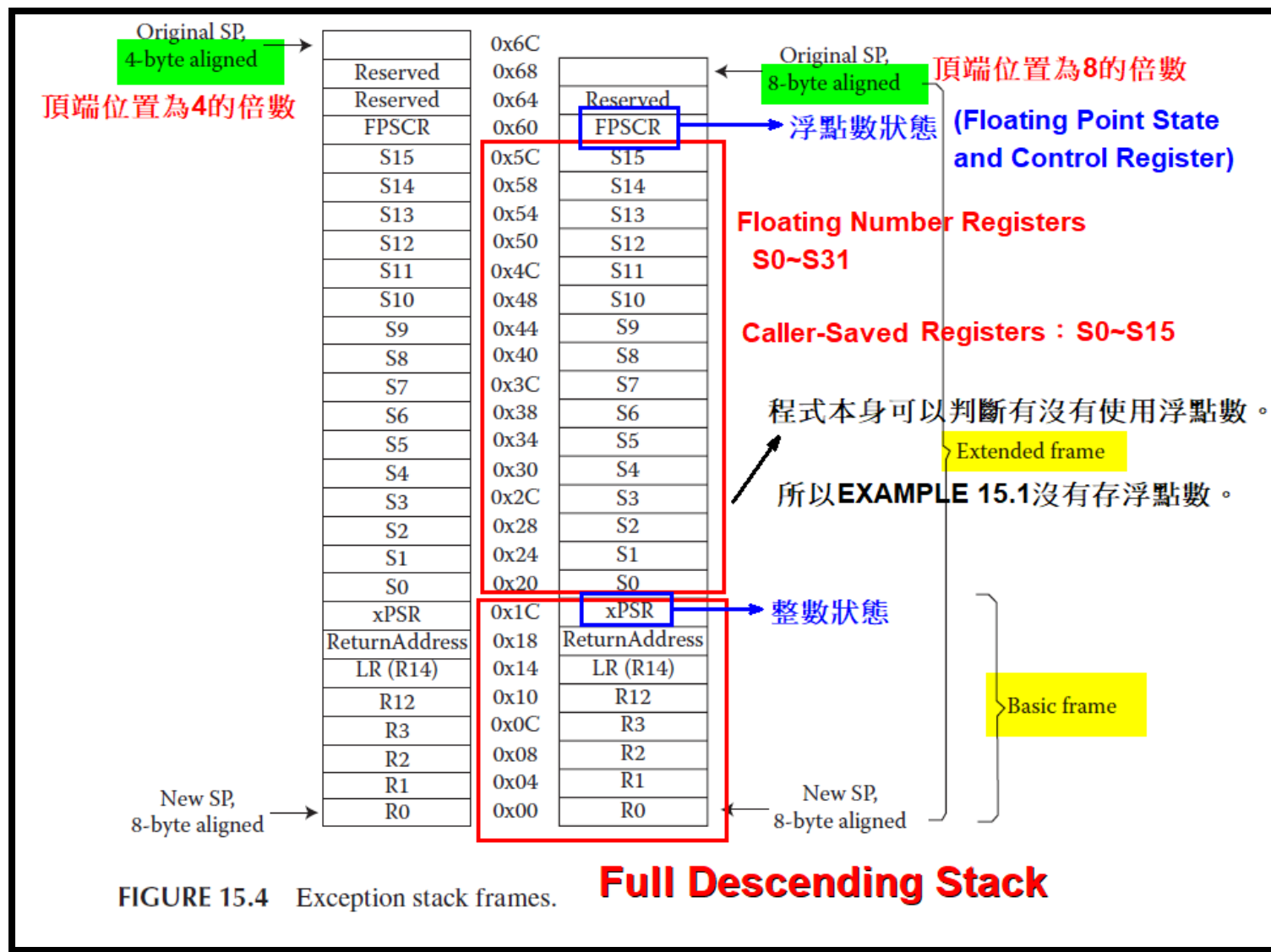
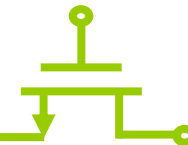
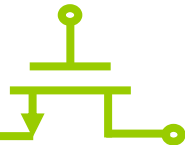


FIGURE 15.4 Exception stack frames.

EXAMPLE 15.2 Switch to user Thread mode



IntDefaultHandler

; let's read the Usage Fault Status Register

LDR r7, =Usagefault

LDRH r1, [r6, r7]

TEQ r1, #0x200

IT NE

LDRNE r9, =0xDEADDEAD

; r1 should have bit 9 set indicating

; a divide-by-zero has taken place

; switch to user Thread mode

MRS (從S到R) r8, CONTROL

ORR r8, r8, #1

MSR (從R到S) CONTROL, r8

BX LR

ALIGN

原本使用MOV

因為有使用到系統暫存器，所以用MRS、MSR。

Register(一般暫存器)：r8...

Special register(系統暫存器)：CONTROL...

回到原來被中斷的地方
但意義與副函式不同。

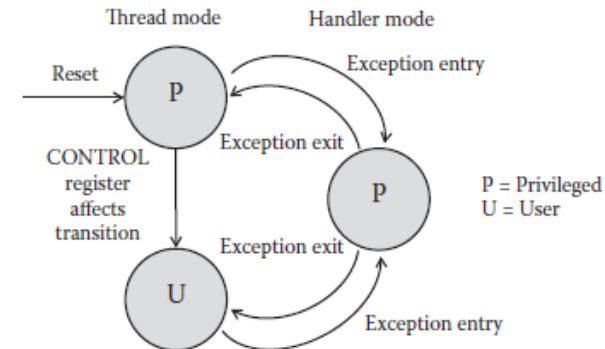


FIGURE 15.1 Cortex-M4 operation modes.

31	3	2	1	0
Reserved	FPCA	ASP	TMPL	

FPCA - Floating-point context active

1 - Preserve floating-point state when processing exception

0 - No floating-point context active

ASP - Active stack pointer

1 - PSP

0 - MSP

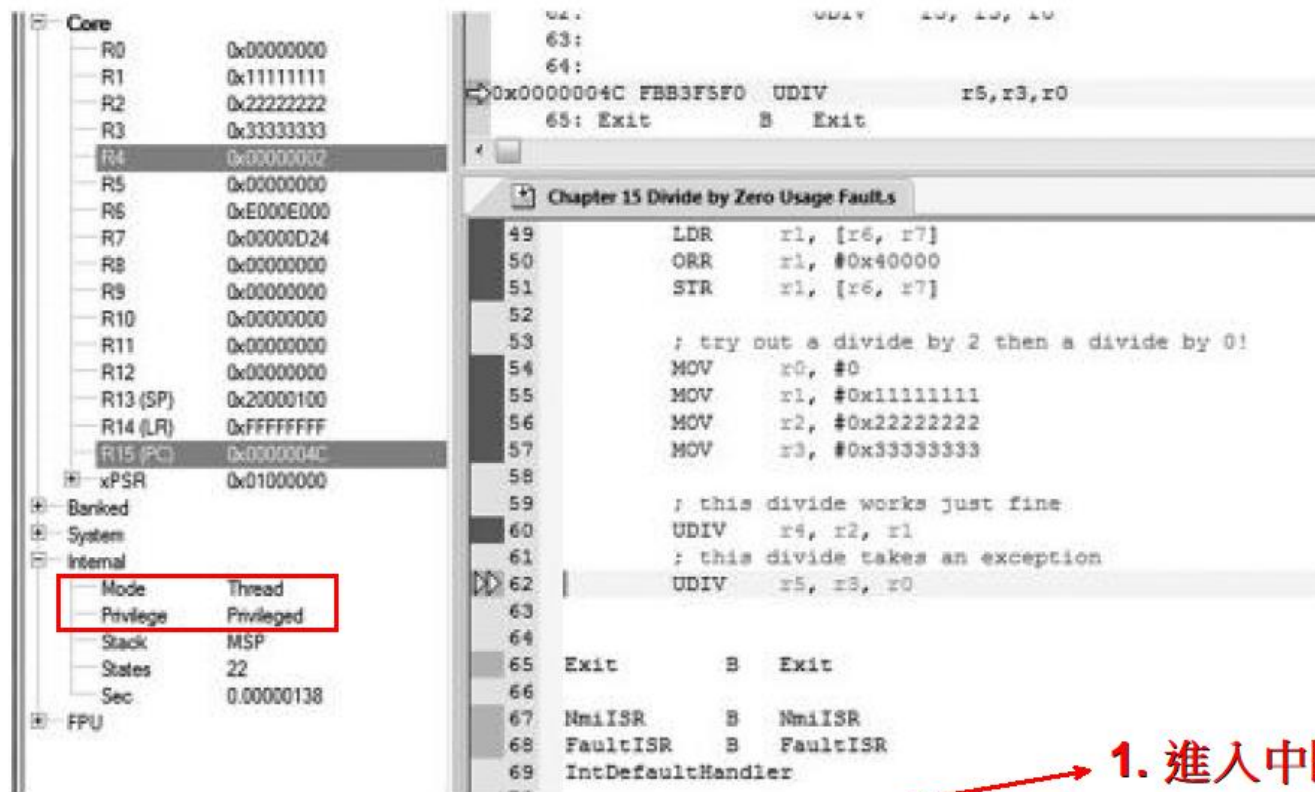
TMPL - Thread mode privilege level

1 - Unprivileged

0 - Privileged

FIGURE 15.2 CONTROL Register on the Cortex-M4.

觀察Keil Tool上Mode、Privilege變化



觀察Mode、Privileged變化。

1. 進入中斷程式後
 2. 執行完EXAMPLE15.2回到主程式
- (有可能會考)

FIGURE 15.3 Cortex-M4 operating in privileged Thread mode.

Q&A

Thanks for your attention !!