PSI3441 – Arquitetura de Sistemas Embarcados

Interrupções

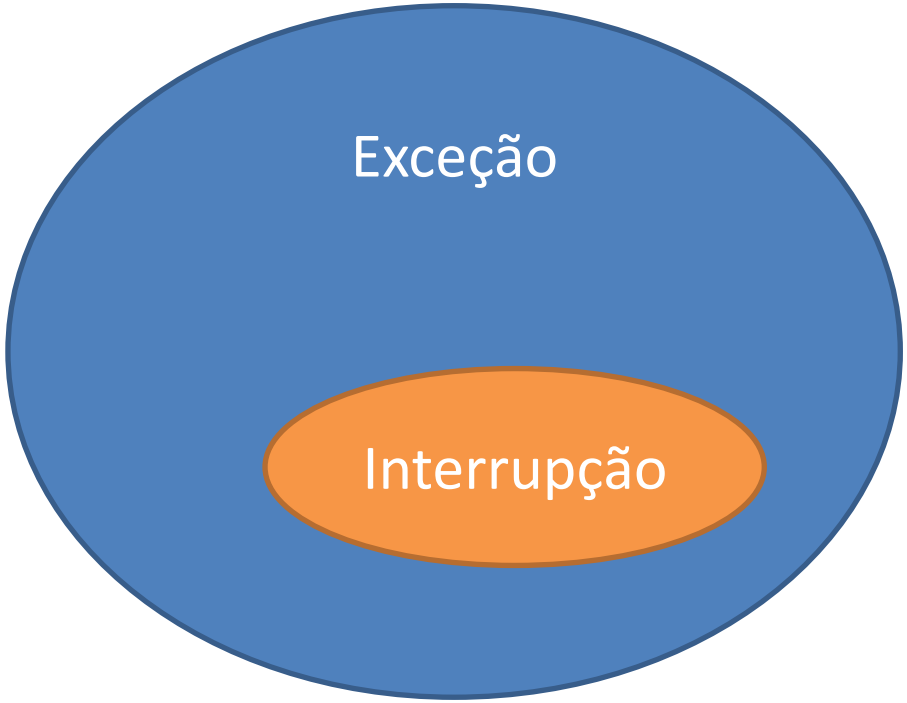
Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – grehder@lme.usp.br

2018

Exceção – Interrupção

- Exceção: Qualquer evento que altera o fluxo normal de execução de instruções.



Exceção: eventos anormais:
- Instruções inválidas
- Acesso ilegal

Interrupção: Sinal de hardware
- Sinais externos
- Flags de periféricos

Pooling vs. Interrupção

- Pooling

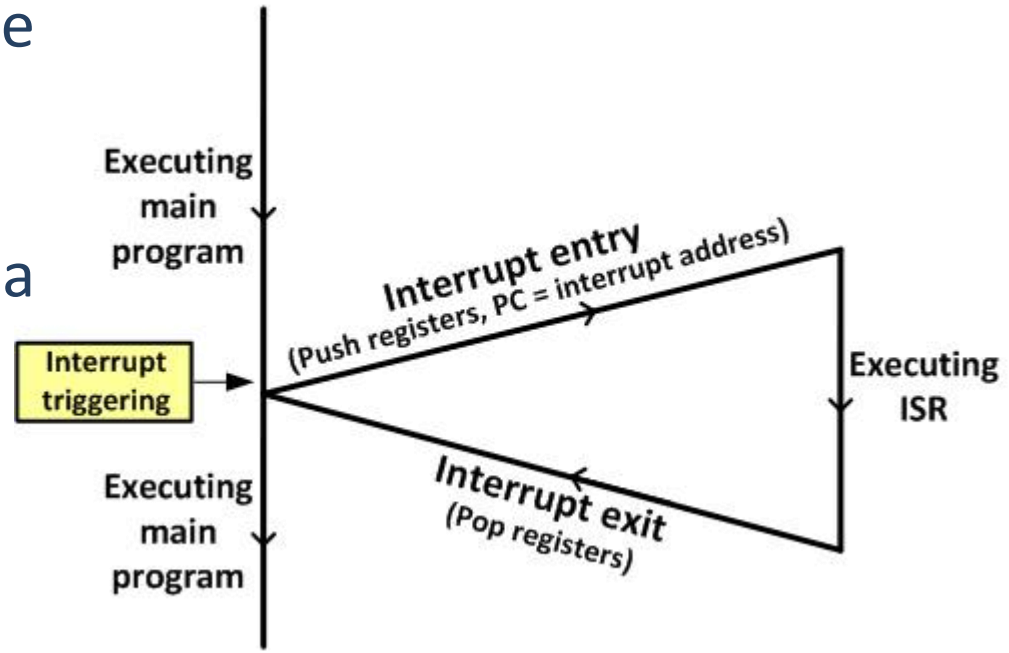
```
// Monitora continuamente o flag do registrador.

//No main
for(;;)
{
if(SysTick_PDD_ReadControlStatusReg(SysTick_DEVICE)&0x10000)
{
Bit1_NegVal();
}
}
```

- Interrupção

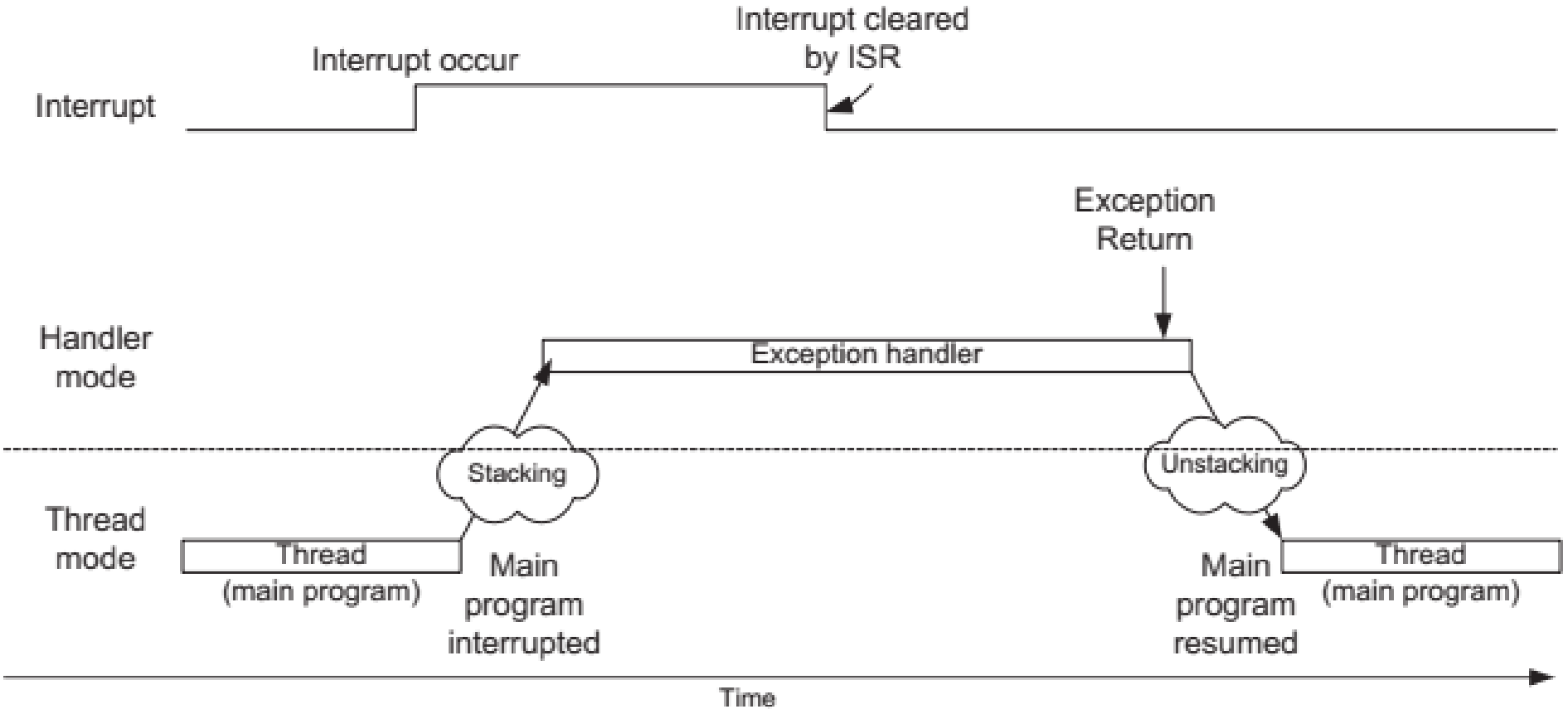
```
//Espera uma interrupção para executar a função.

//No main
for(;;)
{
}

//No Events (Interrupt service routine (ISR))
void TI1_OnInterrupt(LDD_TUserData *UserDataPtr)
{
Bit1_NegVal();
}
```

O que acontece quando ocorre uma interrupção

- Periférico gera um pedido de interrupção (interrrupt request – IRQ)
- O processador detecta e aceita o IRQ. O programa que está rodando é suspenso e a CPU salva no stack estado atual (contadores, registradores, flags, etc);
- CPU busca o endereço das rotinas de tratamento de interrupção (Interrupt Service Routine - ISR) em uma tabela (Vector Table);
- CPU vai para o endereço do ISR e executa seu código (função);
- CPU retorna ao estado anteriormente salvo e continua a execução do programa anterior.



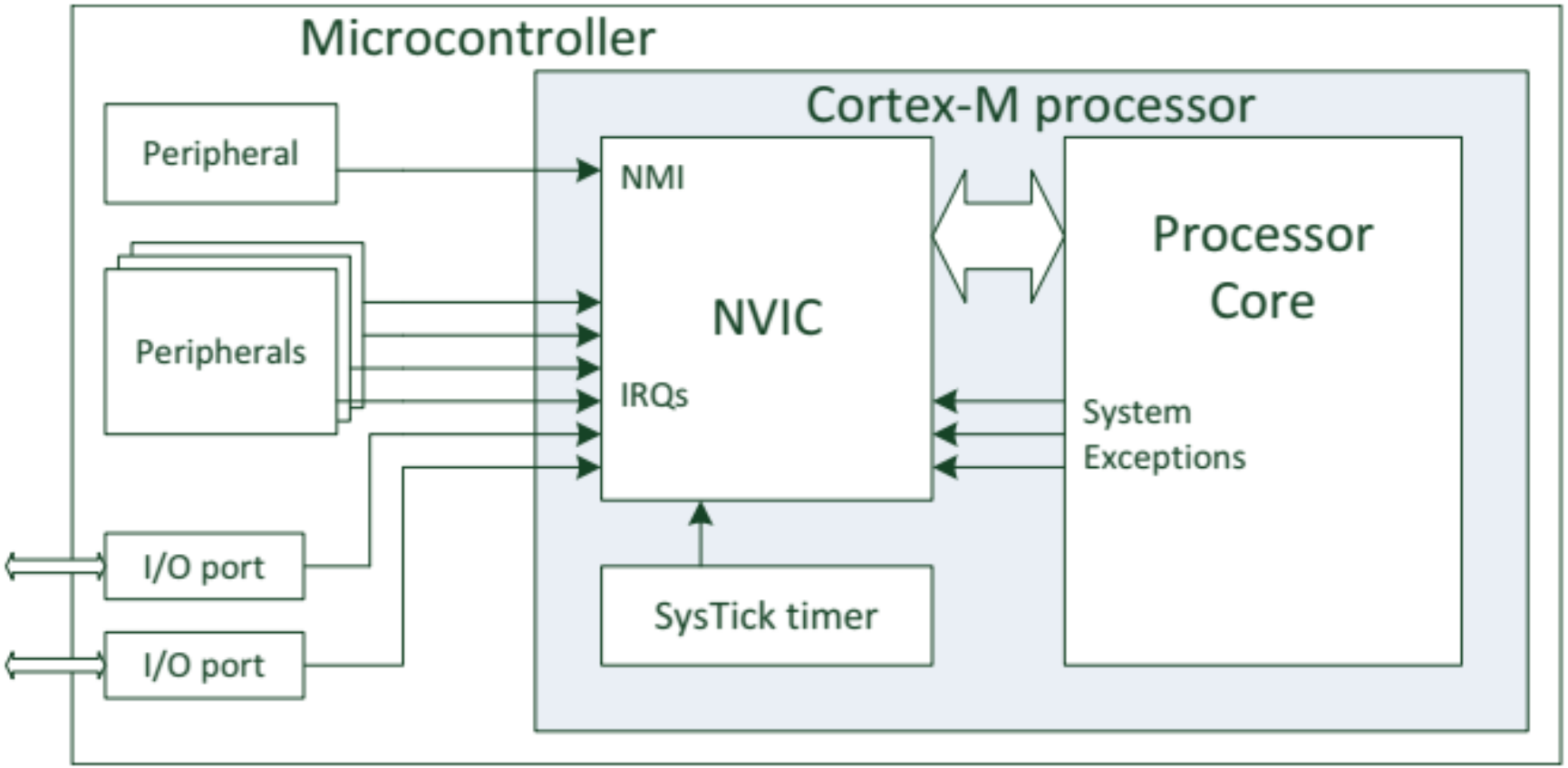
Exceção/Interrupção execução



Definições

- *Interrupt Requests* (IRQs) – Interrupções associadas a periféricos externos, ex. GPIOs. O Cortex M0+ possuí 32 IRQs.
- *Non-Maskable Interrupt* (NMI) – IRQs de alta prioridade, ex. *Watchdog*.
- *Handlers* – função que é executada quando uma exceção ocorre. Caso seja uma interrupção também é conhecido como *interrupt handler* ou *Interrupt Service Routine* (ISR).
- *Nested Interrupts* – por causa dos diferentes níveis de prioridade, uma interrupção pode ocorrer quando um *handler* está sendo executado. Isto é um neste*d interrupt*.

Nested Vectored Interrupt Controller (NVIC)



NVIC – Nested Vectored Interrupt Controller: Controla todos as interrupções externas

Vector Table

Non-Maskable Interrupt **–** não pode ser desabilitado. Falta de energia e watchdog.

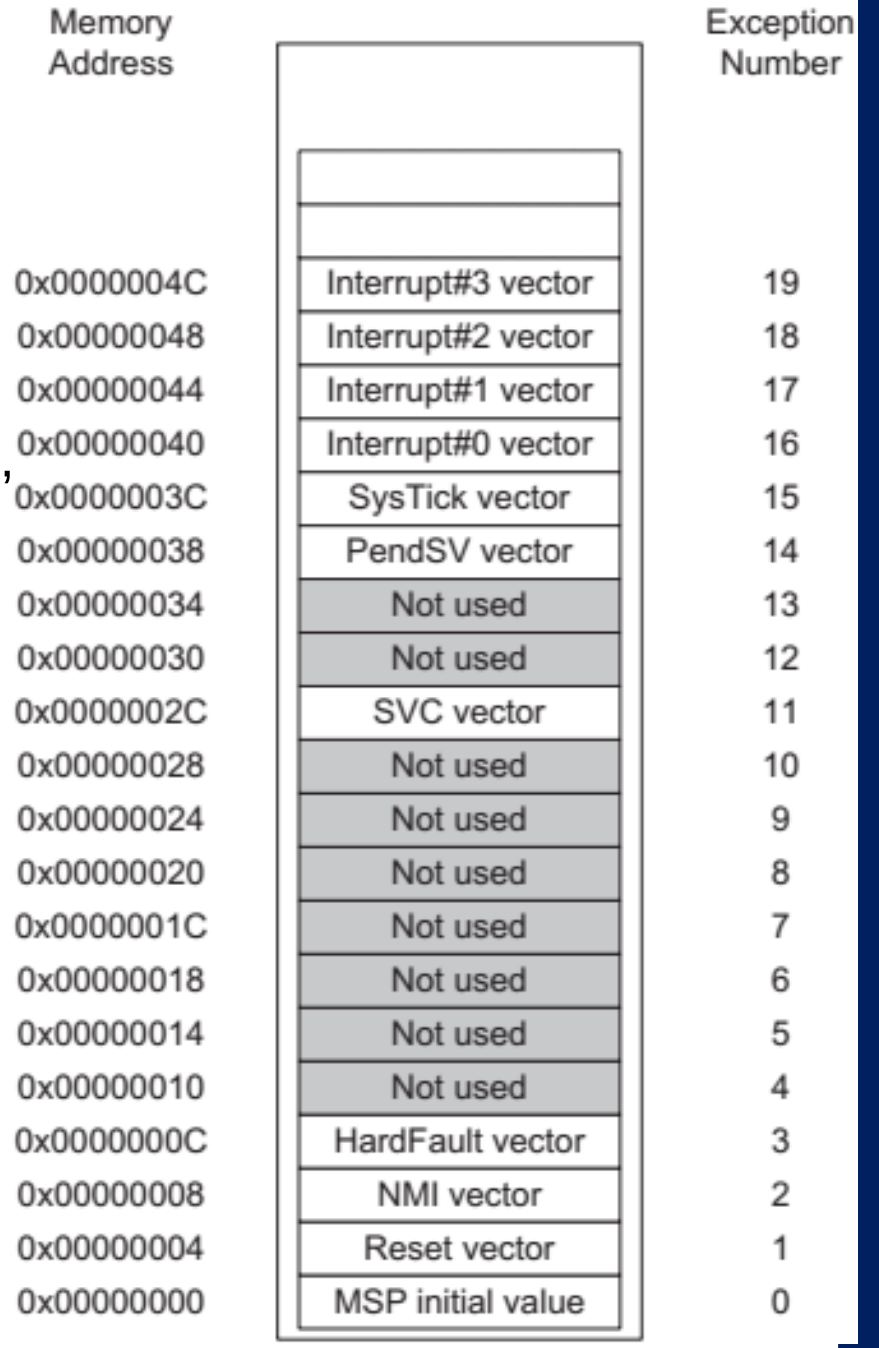
HardFault – instruções inválidas, erros no bus, na memória ou operações ilegais

SVCall (Supervisor Call) – modo de supervisão usado em sistemas operacionais

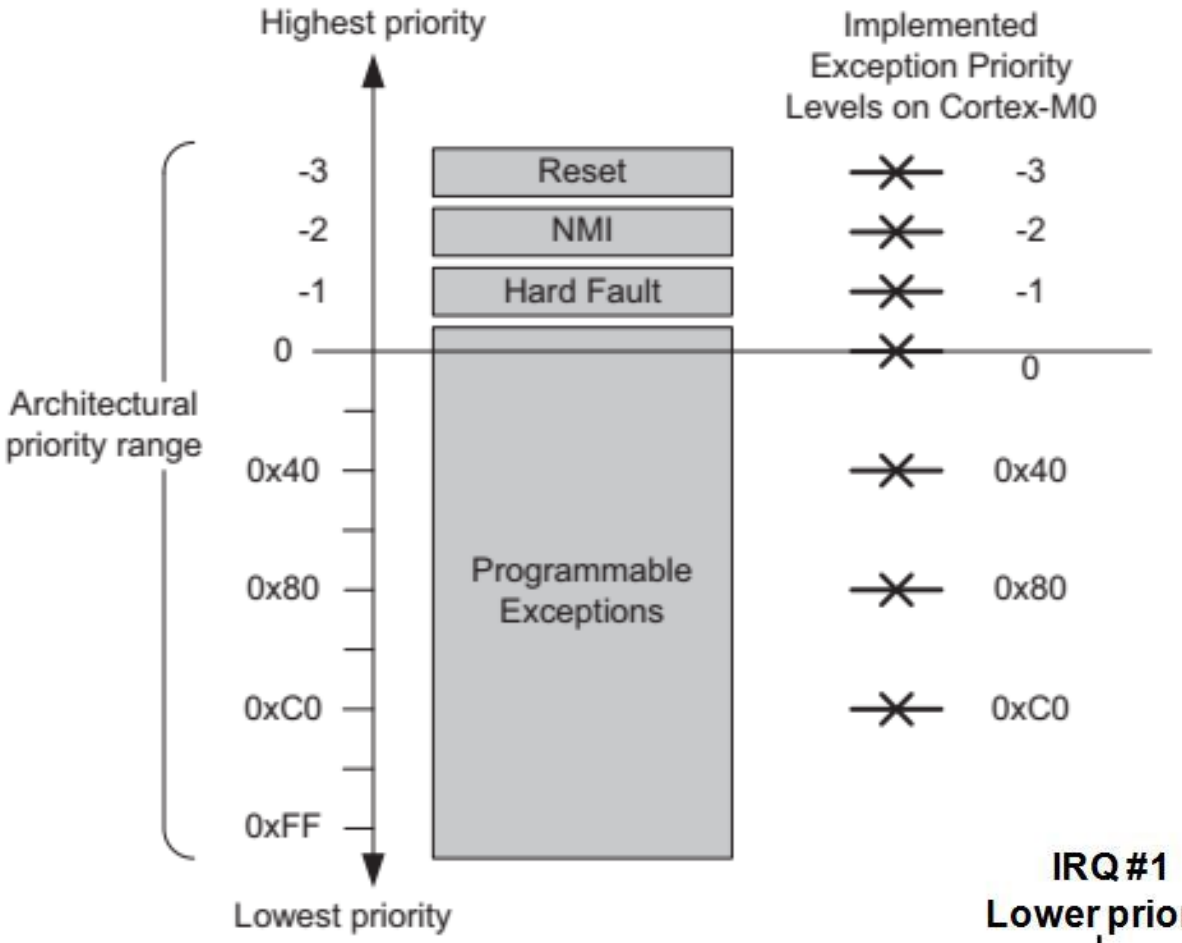
Pendable Service Call – usado por sistemas operacionais

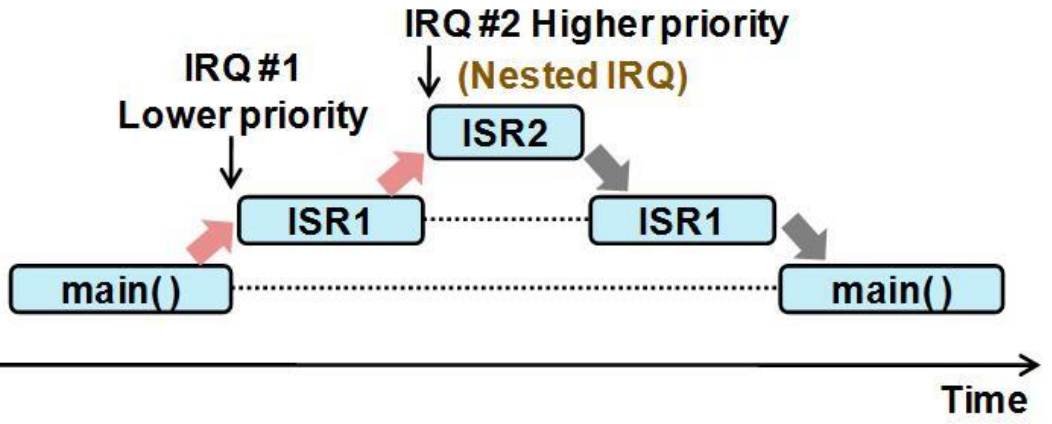
System Tick Timer – interrupção do timer

Interrupts – interrupção de outros periféricos (depende do fabricante)



Prioridade


Highest priority
Implemented
Exception Priority
Levels on Cortex-M0
-3
-2
-1
0
0x40
0x80
0xC0
0xFF
Reset
NMI
Hard Fault
Programmable
Exceptions
-3
-2
-1
0
0x40
0x80
0xC0
Architectural
priority range
Lowest priority


IRQ #2 Higher priority
(Nested IRQ)
IRQ #1
Lower priority
ISR2
ISR1
ISR1
main()
main()
Time

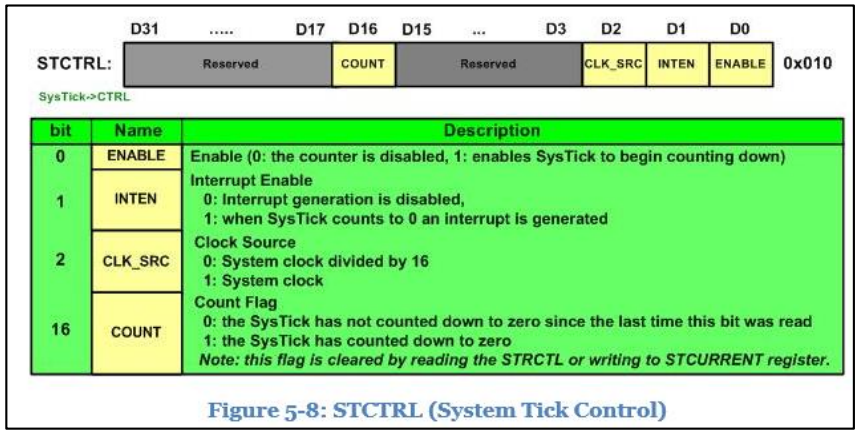
Vector Table

Core

SysTick

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_0000	0	—	—	ARM core	Initial Stack Pointer
0x0000_0004	1	—	—	ARM core	Initial Program Counter
0x0000_0008	2	—	—	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	—	—	ARM core	Hard Fault
0x0000_0010	4	—	—	—	—
0x0000_0014	5	—	—	—	—
0x0000_0018	6	—	—	—	—
0x0000_001C	7	—	—	—	—
0x0000_0020	8	—	—	—	—
0x0000_0024	9	—	—	—	—
0x0000_0028	10	—	—	—	—
0x0000_002C	11	—	—	ARM core	Supervisor call (SVCall)
0x0000_0030	12	—	—	—	—
0x0000_0034	13	—	—	—	—
0x0000_0038	14	—	—	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	—	—	ARM core	System tick timer (SysTick)
Non-Core Vectors					
0x0000_0040	16	0	0	DMA	DMA channel 0 transfer complete and error
0x0000_0044	17	1	0	DMA	DMA channel 1 transfer complete and error
0x0000_0048	18	2	0	DMA	DMA channel 2 transfer complete and error
0x0000_004C	19	3	0	DMA	DMA channel 3 transfer complete and error
0x0000_0050	20	4	1	—	—
0x0000_0054	21	5	1	FTFA	Command complete and read collision
0x0000_0058	22	6	1	PMC	Low-voltage detect, low-voltage warning
0x0000_005C	23	7	1	LLWU	Low Leakage Wakeup
⋮					
0x0000_00A4	41	25	6	DAC0	
0x0000_00A8	42	26	6	TSI0	
0x0000_00AC	43	27	6	MCG	
0x0000_00B0	44	28	7	LPTMR0	
0x0000_00B4	45	29	7	—	
0x0000_00B8	46	30	7	Port control module	Pin detect (Port A)
0x0000_00BC	47	31	7	Port control module	Pin detect (Port D)

Timer



bit	Name	Description
0	ENABLE	Enable (0: the counter is disabled, 1: enables SysTick to begin counting down)
1	INTEN	Interrupt Enable 0: Interrupt generation is disabled, 1: when SysTick counts to 0 an interrupt is generated
2	CLK_SRC	Clock Source 0: System clock divided by 16 1: System clock
16	COUNT	Count Flag 0: the SysTick has not counted down to zero since the last time this bit was read 1: the SysTick has counted down to zero *Note: this flag is cleared by reading the STRCTL or writing to STCURRENT register.*

Figure 5-8: STCTRL (System Tick Control)

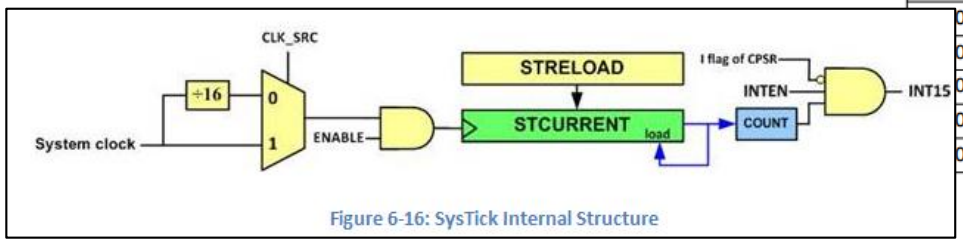
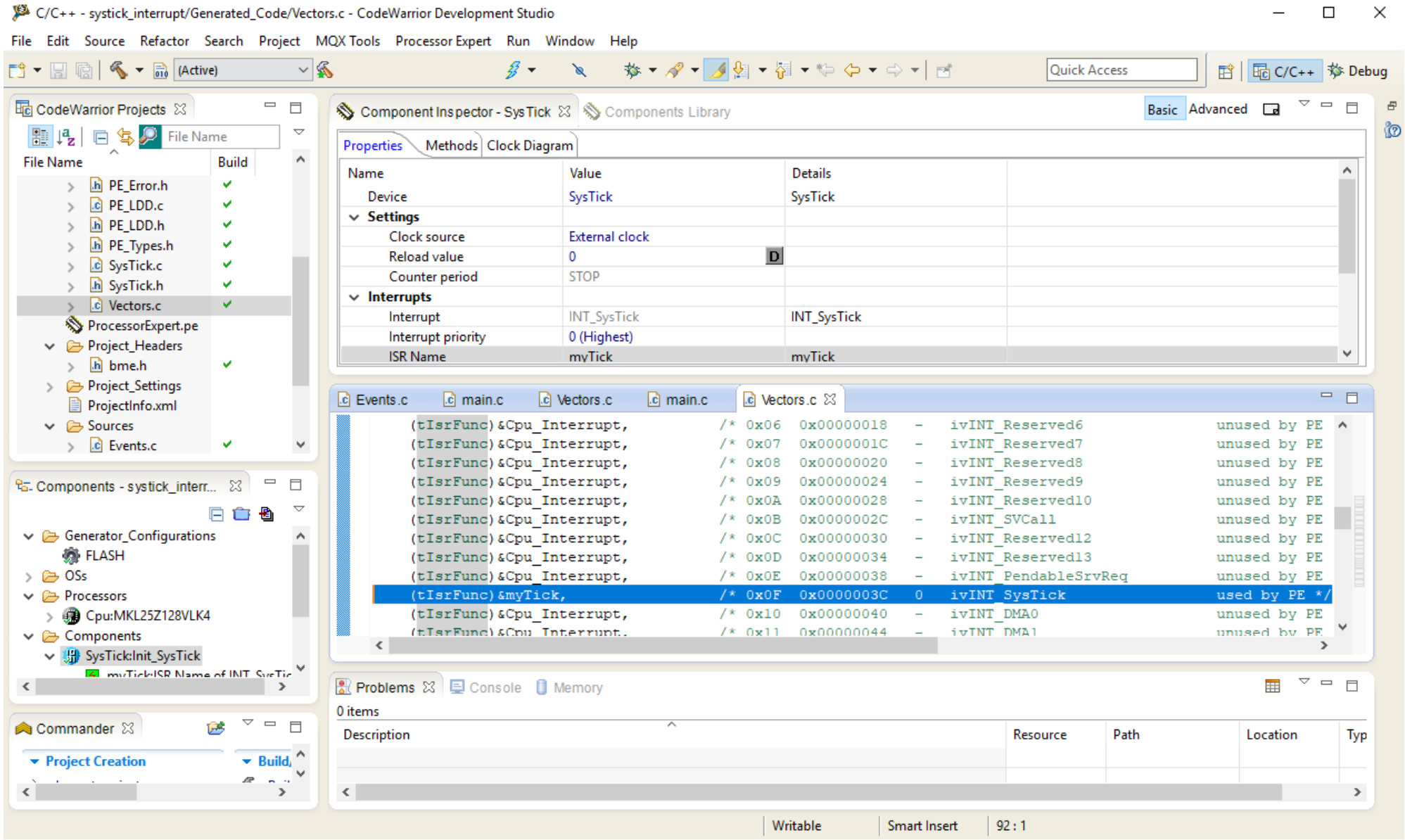


Figure 6-16: SysTick Internal Structure

SysTick – Interrupt Handler



referência

- Yiu, Joseph. *The Definitive Guide to Arm® Cortex®-m0 and Cortex-m0+ Processors*. Academic Press, 2015.