

Agenda

- Introducción
- Presentación de la Familia Cortex M
- Arquitectura de los Cortex M
- Repertorio de Instrucciones y Ejemplos
- Sistema de Memoria
- **Excepciones, Interrupciones y el NVIC**
- La familia NXP LPCXXXX

Excepciones e Interrupciones

¿Por qué excepciones?



Excepciones e interrupciones

3

3

Excepciones

- Es un concepto más amplio que las interrupciones pues no necesariamente se producen por acciones externas.
- Las interrupciones son un subgrupo de las excepciones y se procesan similarmente

Excepciones e interrupciones

4

4

Excepciones

- **Reset**
- **NMI**
- **Faults**
 - Hard Fault
 - Memory Manage
 - Bus Fault
 - Usage Fault
- **SVCall**
- **Debug Monitor**
- **PendSV**
- **SysTick Interrupt**
- **External Interrupt**

- **Hard Fault** : default Fault or any fault unable to activate
- **Memory Manage** : MPU violations
- **Bus Fault** : prefetch and memory access violations
- **Usage Fault** : undef instructions, divide by zero, etc.

Excepciones

Exception Number	Exception Type	Priority (Default to 0 if Programmable)	Description
0	NA	NA	No exception running
1	Reset	− 3 (Highest)	Reset
2	NMI	− 2	Nonmaskable interrupt (external NMI input)
3	Hard fault	− 1	All fault conditions, if the corresponding fault handler is not enabled
4	MemManage fault	Programmable	Memory management fault, MPU violation or access to illegal locations
5	Bus fault	Programmable	Bus error (Prefetch Abort or Data Abort)
6	Usage fault	Programmable	Exceptions due to program error
7-10	Reserved	NA	Reserved
11	SVCall	Programmable	System service call
12	Debug monitor	Programmable	Debug monitor (break points, watchpoints, or external debug request)
13	Reserved	NA	Reserved
14	PendSV	Programmable	Pendable request for system device
15	SYSTICK	Programmable	System tick timer
16	IRQ #0	Programmable	External interrupt #0
17	IRQ #1	Programmable	External interrupt #1
...
255	IRQ #239	Programmable	External interrupt #239

Externas

Excepciones

```
; Vector Table Mapped to Address 0 at Reset

AREA RESET, DATA, READONLY
EXPORT __Vectors

__Vectors
DCD __initial_sp           ; Top of Stack
DCD Reset_Handler         ; Reset Handler
DCD NMI_Handler           ; NMI Handler
DCD HardFault_Handler     ; Hard Fault Handler
DCD MemManage_Handler     ; MPU Fault Handler
DCD BusFault_Handler      ; Bus Fault Handler
DCD UsageFault_Handler    ; Usage Fault Handler
DCD 0                     ; Reserved
DCD 0                     ; Reserved
DCD 0                     ; Reserved
DCD 0                     ; Reserved
DCD SVC_Handler           ; SVC Call Handler
DCD DebugMon_Handler      ; Debug Monitor Handler
DCD 0                     ; Reserved
DCD PendSV_Handler        ; PendSV Handler
DCD SysTick_Handler       ; SysTick Handler
```

Excepciones e interrupciones

7

7

Excepciones

```
; External Interrupts
DCD WDT_IRQHandler         ; 16: Watchdog Timer
DCD TIMER0_IRQHandler     ; 17: Timer0
DCD TIMER1_IRQHandler     ; 18: Timer1
DCD TIMER2_IRQHandler     ; 19: Timer2
DCD TIMER3_IRQHandler     ; 20: Timer3
DCD UART0_IRQHandler      ; 21: UART0
DCD UART1_IRQHandler      ; 22: UART1
DCD UART2_IRQHandler      ; 23: UART2
DCD UART3_IRQHandler      ; 24: UART3
DCD PWM1_IRQHandler       ; 25: PWM1
DCD I2C0_IRQHandler       ; 26: I2C0
DCD I2C1_IRQHandler       ; 27: I2C1
DCD I2C2_IRQHandler       ; 28: I2C2
DCD SPI_IRQHandler        ; 29: SPI
DCD SSP0_IRQHandler       ; 30: SSP0
DCD SSP1_IRQHandler       ; 31: SSP1
DCD PLL0_IRQHandler       ; 32: PLL0 Lock (Main PLL)
DCD RTC_IRQHandler        ; 33: Real Time Clock
```

Excepciones e interrupciones

8

8

Excepciones

```
DCD EINT0_IRQHandler ; 34: External Interrupt 0
DCD EINT1_IRQHandler ; 35: External Interrupt 1
DCD EINT2_IRQHandler ; 36: External Interrupt 2
DCD EINT3_IRQHandler ; 37: External Interrupt 3
DCD ADC_IRQHandler ; 38: A/D Converter
DCD BOD_IRQHandler ; 39: Brown-Out Detect
DCD USB_IRQHandler ; 40: USB
DCD CAN_IRQHandler ; 41: CAN
DCD DMA_IRQHandler ; 42: General Purpose DMA
DCD I2S_IRQHandler ; 43: I2S
DCD ENET_IRQHandler ; 44: Ethernet
DCD RIT_IRQHandler ; 45: Repetitive Interrupt Timer
DCD MCPWM_IRQHandler ; 46: Motor Control PWM
DCD QEI_IRQHandler ; 47: Quadrature Encoder Interface
DCD PLL1_IRQHandler ; 48: PLL1 Lock (USB PLL)
DCD USBActivity_IRQHandler ; 49: USB Activity interrupt to wakeup
DCD CANActivity_IRQHandler ; 50: CAN Activity interrupt to wakeup
```

Excepciones e interrupciones

9

9

¿WEAK?

```
; Reset Handler

Reset_Handler PROC
EXPORT Reset_Handler [WEAK]
IMPORT SystemInit
IMPORT __main
LDR R0, =SystemInit
BLX R0
LDR R0, =__main
BX R0
ENDP

; Dummy Exception Handlers (infinite loops which can be modified)

NMI_Handler PROC
EXPORT NMI_Handler [WEAK]
B .
ENDP

HardFault_Handler\
PROC
EXPORT HardFault_Handler [WEAK]
B .
ENDP

MemManage_Handler\
PROC
EXPORT MemManage_Handler [WEAK]
B .
ENDP
```

Excepciones e interrupciones

10

10

IRQHandler

```
Default_Handler PROC

EXPORT WDT_IRQHandler           [WEAK]
EXPORT TIMER0_IRQHandler       [WEAK]
EXPORT TIMER1_IRQHandler       [WEAK]
EXPORT TIMER2_IRQHandler       [WEAK]
EXPORT TIMER3_IRQHandler       [WEAK]
EXPORT UART0_IRQHandler        [WEAK]
EXPORT UART1_IRQHandler        [WEAK]
EXPORT UART2_IRQHandler        [WEAK]
EXPORT UART3_IRQHandler        [WEAK]
EXPORT PWM1_IRQHandler         [WEAK]
EXPORT I2C0_IRQHandler         [WEAK]
EXPORT I2C1_IRQHandler         [WEAK]
EXPORT I2C2_IRQHandler         [WEAK]
EXPORT SPI_IRQHandler          [WEAK]
EXPORT SSP0_IRQHandler         [WEAK]
EXPORT SSP1_IRQHandler         [WEAK]
EXPORT PLL0_IRQHandler         [WEAK]
EXPORT RTC_IRQHandler          [WEAK]
```

Excepciones e interrupciones

11

11

Vectores

```
EXPORT CANActivity_IRQHandler [WEAK]

WDT_IRQHandler
TIMER0_IRQHandler
TIMER1_IRQHandler
TIMER2_IRQHandler
TIMER3_IRQHandler
UART0_IRQHandler
UART1_IRQHandler
UART2_IRQHandler
UART3_IRQHandler
PWM1_IRQHandler
I2C0_IRQHandler
I2C1_IRQHandler
I2C2_IRQHandler
SPI_IRQHandler
SSP0_IRQHandler
SSP1_IRQHandler
PLL0_IRQHandler
RTC_IRQHandler
EINT0_IRQHandler
EINT1_IRQHandler
```

Excepciones e interrupciones

12

12

Cerca del Final...

```

B      .

ENDP

ALIGN

; User Initial Stack & Heap

IF      :DEF:__MICROLIB

EXPORT  __initial_sp
EXPORT  __heap_base
EXPORT  __heap_limit

ELSE

IMPORT  __use_two_region_memory
EXPORT  user_initial_stackheap


```

Excepciones e interrupciones

13

13

Final....

```

__user_initial_stackheap

LDR     R0, = Heap_Mem
LDR     R1, =(Stack_Mem + Stack_Size)
LDR     R2, =(Heap_Mem +  Heap_Size)
LDR     R3, = Stack_Mem
BX      LR

ALIGN

ENDIF

END


```

Excepciones e interrupciones

14

14

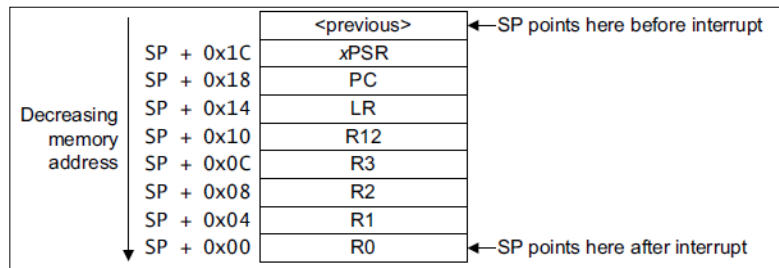
NVIC

- Se introduce un controlador de interrupciones vectorizadas y anidadas
- Permite el cambio dinámico de prioridades de interrupción.
- Reducción del tiempo de latencia de interrupción
- Enmascaramiento individual de interrupciones
- System Tick (Sys Tick), que es un contador descendente de 24 bits imaginado para trabajar con un RTOS u otras tareas similarmente planificadas

Manejo de interrupciones microcodificado

- **Entrada:**
 - Se pushean automáticamente a la pila los registros R0–R3, R12, LR, PSR, y el PC
 - En paralelo se hace la pre-fetch de la ISR en el bus de instrucciones.
 - ISR esta lista para comenzar tan pronto termine las operaciones de PUSH.
 - La llegada tardía de otra interrupción reiniciará el prefetch de la ISR pero no necesita repetir el salvado de registros.
- **Salida:**
 - El estado del procesador es recuperado automáticamente de la pila.
 - En paralelo, la instrucción interrumpida es prefetched, lista para reiniciar la ejecución apenas se completen los POPs de la pila.
 - Los POPs de la pila pueden ser interrumpidos, permitiendo nuevas ISR ser inmediatamente atendidas sin la sobrecarga temporal de guardar registros

Las excepciones y el stack



17

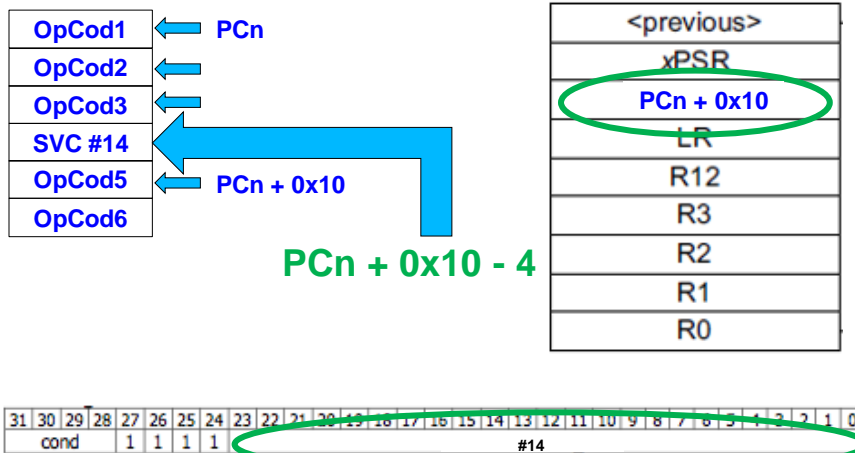
SVC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cond				1	1	1	1	immediate_24																							

Teniendo en cuenta que los 24 bits menos significativos no son atendidos por el procesador, ¿Cómo hace el user thread para pedir al handler su intervención para diversas acciones?

18

Como utilizar SVC para hacer un pedido del modo user thread al handler

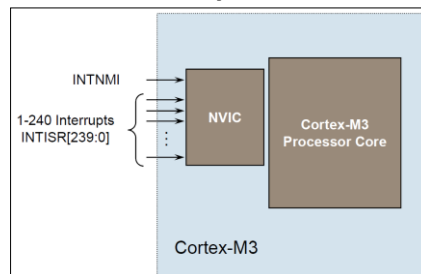


Excepciones e interrupciones

19

19

Interrupciones



- Una interrupción no enmascarable (INTNMI)
 - 1-240 interrupciones priorizables
 - Las interrupciones pueden ser enmascaradas
- Puede seleccionarse el número de interrupciones activas
- El controlador de interrupciones anidadas (NVIC) está fuertemente asociado al núcleo del procesador
- Las entradas de interrupción son activas ALTA

Excepciones e interrupciones

20

20

Terminología (a ser ampliada)

- Determinístico: Que es predecible. Dado un conjunto de entradas, siempre producirá las mismas salidas pasando por la misma secuencia de estados.
- Prevaciado (preemption) Habilidad de un sistema operativo o programa similar de detener la ejecución de una tarea programada a favor de una tarea de mayor prioridad.
- Tail chaining: Forma de encolar los retornos de interrupciones anidadas, de forma de optimizar el tiempo total de ejecución.
- Latencia de interrupciones: Demora experimentada desde que se genera una interrupción hasta que es atendida

Excepciones e interrupciones

21

21

Interrupciones

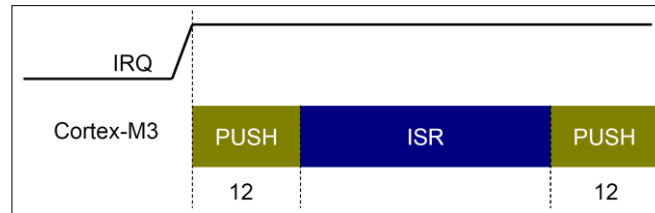
- Latencia de interrupciones determinística
- Características avanzadas
 - Prioridad de pre-vaciado
 - Tail chaining (encadenado de colas)

Excepciones e interrupciones

22

22

Latencia de Interrupciones del NVIC



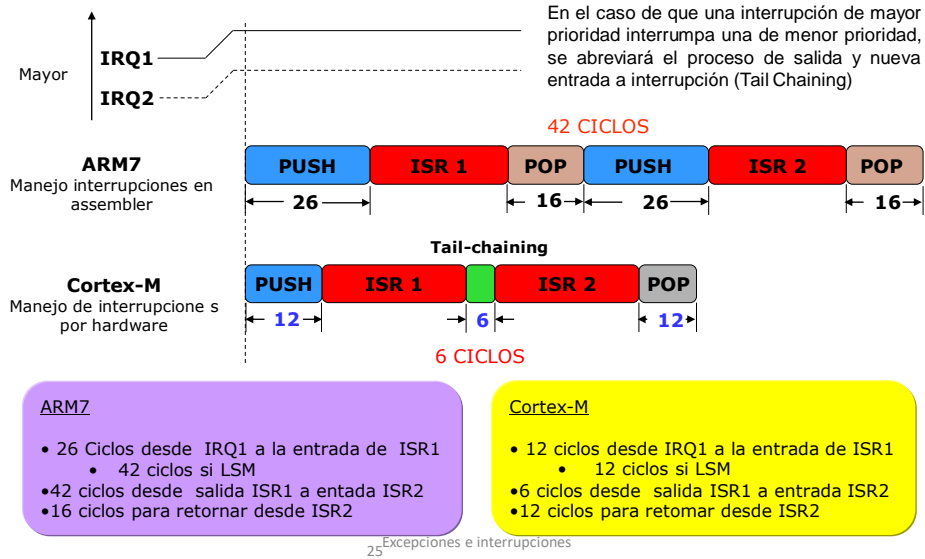
Latencia de interrupciones determinística

- Cortex-M tiene una latencia de interrupción de 12 ciclos y 12 ciclos para retornar de la ISR
- ARM7 tiene una latencia de interrupción entre 26 y 42 ciclos y 16 de retorno. NO ES DETERMINÍSTICO

Tail Chaining

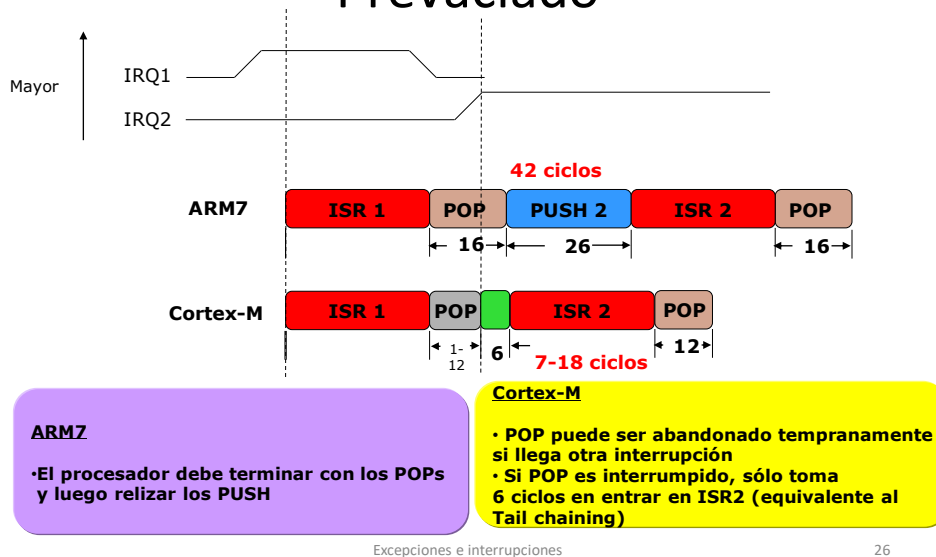
- En el caso de que una interrupción de mayor prioridad interrumpa una de menor prioridad, se abreviará el proceso de salida y nueva entrada a interrupción (Tail Chaining)

Tail Chaining



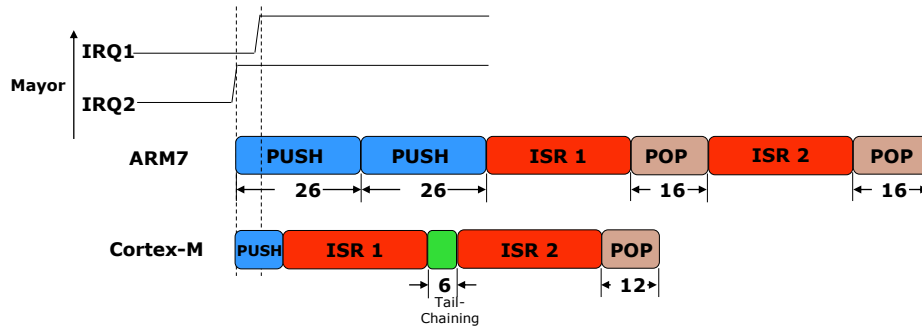
25

Latencia de interrupciones - Prevaciado



26

Latencia de interrupciones – Llegada tardía



ARM7

- 26 ciclos Para entrar a ISR2
- Inmediatamente pre-vaciado por IRQ1 y lleva 26 ciclos más entrar a ISR 1.
- ISR 1 se completa y lleva 16 ciclos Retornar a ISR 2.

Cortex-M

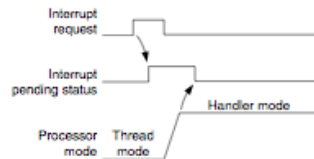
- Se interrumpen los push a ISR 2
- Stacking continúa pero la dirección del nuevo vector se busca en paralelo
- 6 ciclos más tarde entra ISR1.
- Tail-chain en ISR 2

Excepciones e interrupciones

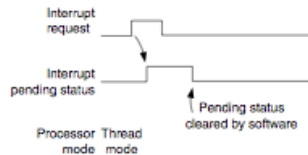
27

27

Manipulación de interrupciones en NVIC



Interrupción pendiente



Interrupción limpiada previa a la
Atención por el procesador

Excepciones e interrupciones

28

28

Tabla de vectores Inicial

Exception Type	Address Offset	Exception Vector
18-255	0x48-0x3FF	IRQ #2-239
17	0x44	IRQ #1
16	0x40	IRQ #0
15	0x3C	SYSTICK
14	0x38	PendSV
13	0x34	Reserved
12	0x30	Debug Monitor
11	0x2C	SVC
7-10	0x1C-0x28	Reserved
6	0x18	Usage fault
5	0x14	Bus fault
4	0x10	MemManage fault
3	0x0C	Hard fault
2	0x08	NMI
1	0x04	Reset
0	0x00	Starting value of the MSP

Tabla de vectores Inicial

Exception number	IRQ number	Offset	Vector
16+n	n	0x0040+4n	IRQn
.	.	.	.
.	.	.	.
.	.	.	.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
		0x0038	
13			Reserved
12			Reserved for Debug
11	-5		SVCall
10		0x002C	
9			
8			Reserved
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

Regreso de excepción

EXC_RETURN	Description
0xFFFFFFFF1	Return to Handler mode. Exception return gets state from the main stack. Execution uses MSP after return.
0xFFFFFFFF9	Return to Thread mode. Exception Return get state from the main stack. Execution uses MSP after return.
0xFFFFFFF9D	Return to Thread mode. Exception return gets state from the process stack. Execution uses PSP after return.
All other values	Reserved.

Desde el Reset.....

