

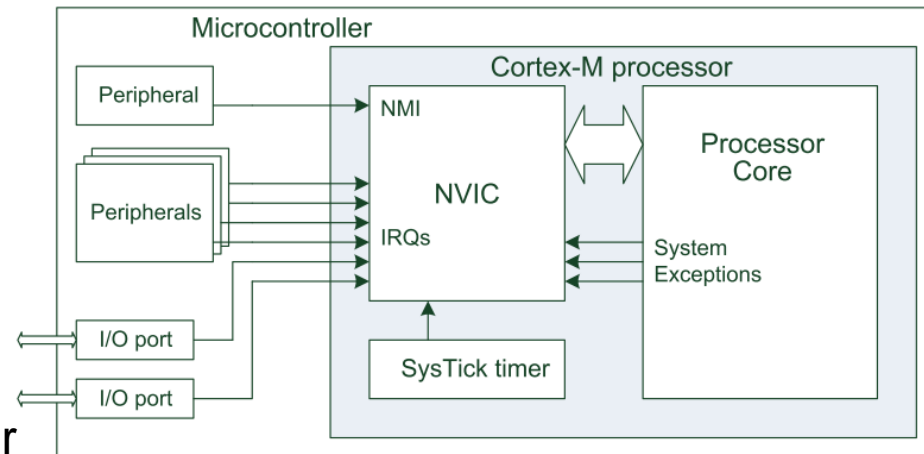
STM32F429 - system przerwain

Sygnały zgłoszenia przerwain:

- podawane wprost na odpowiednie wejście procesora/mikrokontrolera,
- generowane przez urządzenia peryferyjne,
- generowane przez bloki procesora np. (wyjątki: FPU x87, MMU - błędy strony),
- odpowiednią instrukcją (np. int 0x80, syscall w x86-64).

Gdy priorytet odebranego przerwain jest > od aktualnie wykonywanego programu:

- następuje **przerwain aktualnie wykonywanego programu**,
- **skok do procedury obsługi przerwain (ISR – Interrupt Service Routine)**.



NVIC – Nested Vectored Interrupt Controller

STM32F429 - system przerwania

- Zgłoszenie konieczności obsługi przerwania może nadejść w dowolnym momencie.
- Obsługa przerwania jest więc sterowaniem przebiegiem programu w losowych, nieprzewidywalnych momentach – cały system (sprzęt i oprogramowanie) musi być odpowiednio przygotowany (możliwość kontynuacji przerwanej zadania).
- Każde przerwanie ma swój priorytet (domyślny, większość można modyfikować).
- W przypadku nadejścia wielu zgłoszeń jednocześnie* o kolejności ich obsługi decyduje priorytet przerwania (*szczegóły dalej...).
- Program użytkownika (np. main, „thread” w Cortex-M4) ma z reguły priorytet najniższy (tzn. będzie w czasie swojego działania wielokrotnie przerywany).
- Priorytetu programu użytkownika nie można zmienić. W krytycznej sytuacji, jeżeli mamy techniczną możliwość, przerwania można wyłączyć.

STM32F429 - tablica wektorów przerwań

zawiera **adresy** procedur obsługi przerwań i zajmuje początek przestrzeni adresowej (opcjonalnie można ją relokować).

W STM32F4xx może być ok. 100 źródeł przerwań, zależnie od wersji układu...

Position	Priority	Type of priority	Acronym	Description	Address
	-	-	-	Reserved	0x0000 0000
	-3	fixed	Reset	Reset	0x0000 0004
	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000 0008
	-1	fixed	HardFault	All class of fault	0x0000 000C
	0	settable	MemManage	Memory management	0x0000 0010
	1	settable	BusFault	Pre-fetch fault, memory access fault	0x0000 0014
	2	settable	UsageFault	Undefined instruction or illegal state	0x0000 0018
	-	-	-	Reserved	0x0000 001C - 0x0000 002B
	3	settable	SVCcall	System service call via SWI instruction	0x0000 002C
	4	settable	Debug Monitor	Debug Monitor	0x0000 0030
	-	-	-	Reserved	0x0000 0034
	5	settable	PendSV	Pendable request for system service	0x0000 0038
	6	settable	SysTick	System tick timer	0x0000 003C
0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
1	8	settable	PVD	PVD through EXTI line detection interrupt	0x0000 0044
2	9	settable	TAMP_STAMP	Tamper and TimeStamp interrupts through the EXTI line	0x0000 0048
3	10	settable	RTC_WKUP	RTC Wakeup interrupt through the EXTI line	0x0000 004C
4	11	settable	FLASH	Flash global interrupt	0x0000 0050
5	12	settable	RCC	RCC global interrupt	0x0000 0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000 0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000 0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000 0068
11	18	settable	DMA1_Stream0	DMA1 Stream0 global interrupt	0x0000 006C

← 4 pierwsze bajty zajmuje adres początkowy wierzchołka stosu.

Fragment tablicy wektorów przerwań (rozdział 12. Reference Manual).

STM32F429 - system przerwań

przykłady wyjątków „systemowych”

Memory Management Fault Status Register (Offset 0xD28)

Bit	Name	Reset Value	Description
7	MMARVALID	0	Indicates the Memory Management Address register is valid
6:5	—	—	—
4	MSTKERR	0	Stacking error
3	MUNSTKERR	0	Unstacking error
2	—	—	—
1	DACCVIOL	0	Data access violation
0	IACCVIOL	0	Instruction access violation

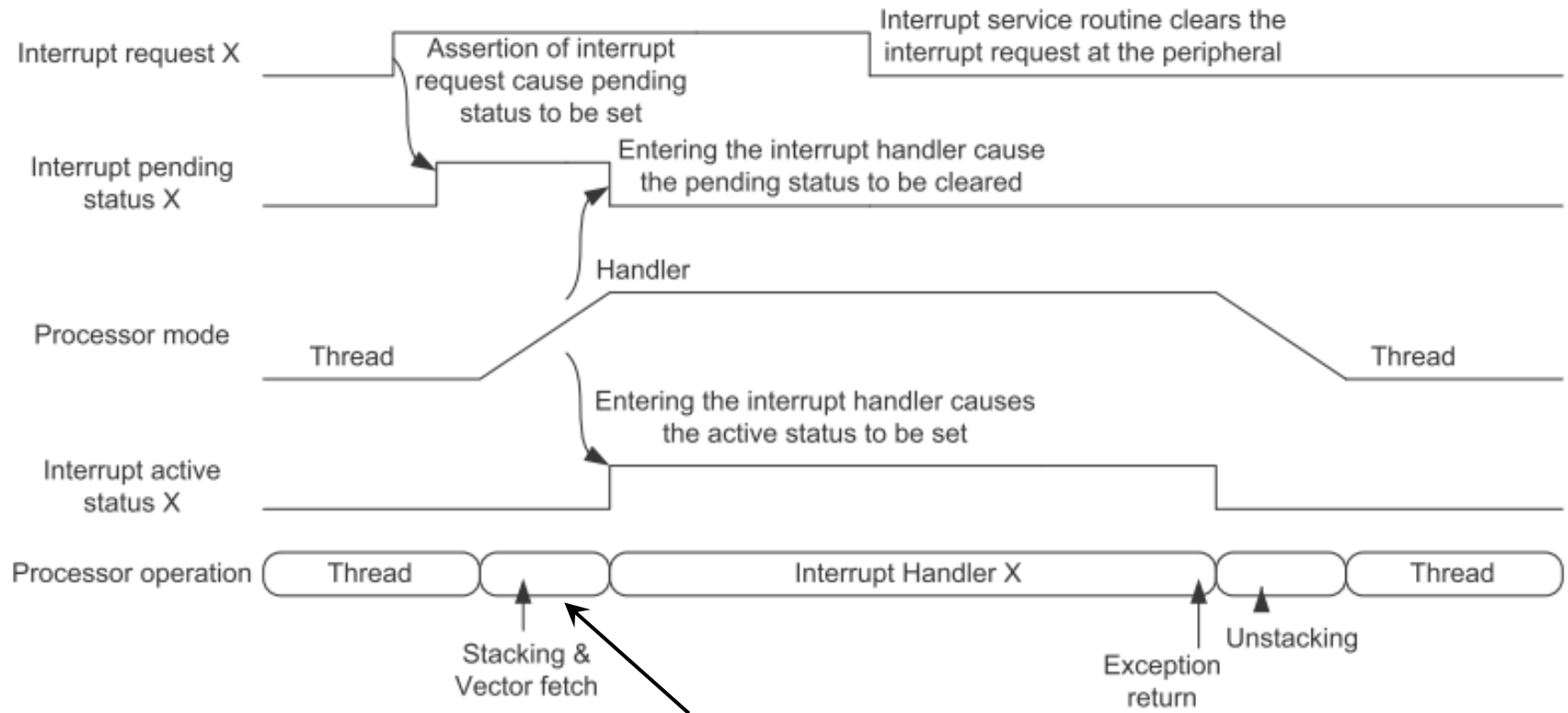
Usage Fault Status Register (Offset 0xD2A)

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

STM32F429 - system przerw

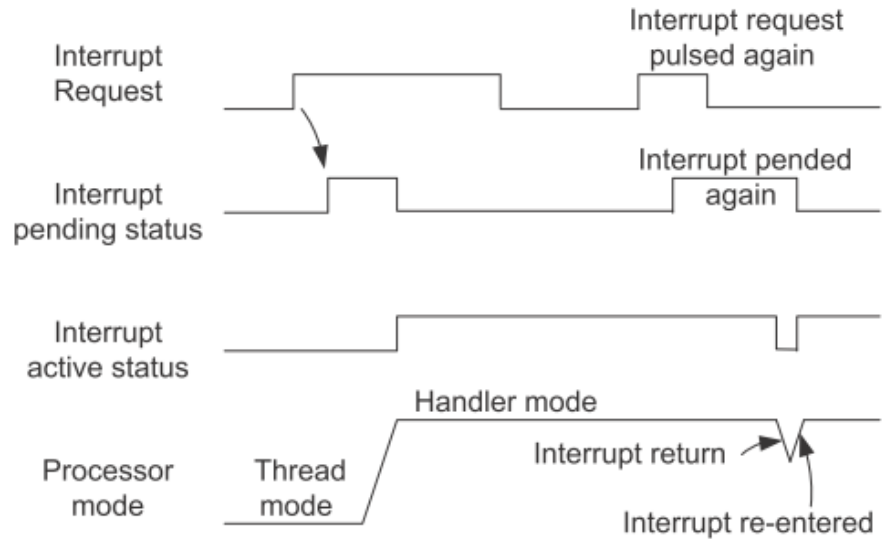
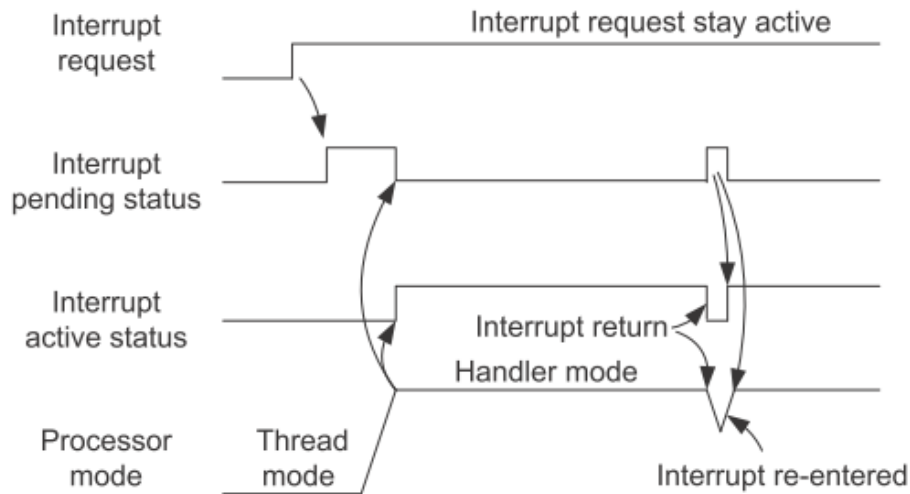
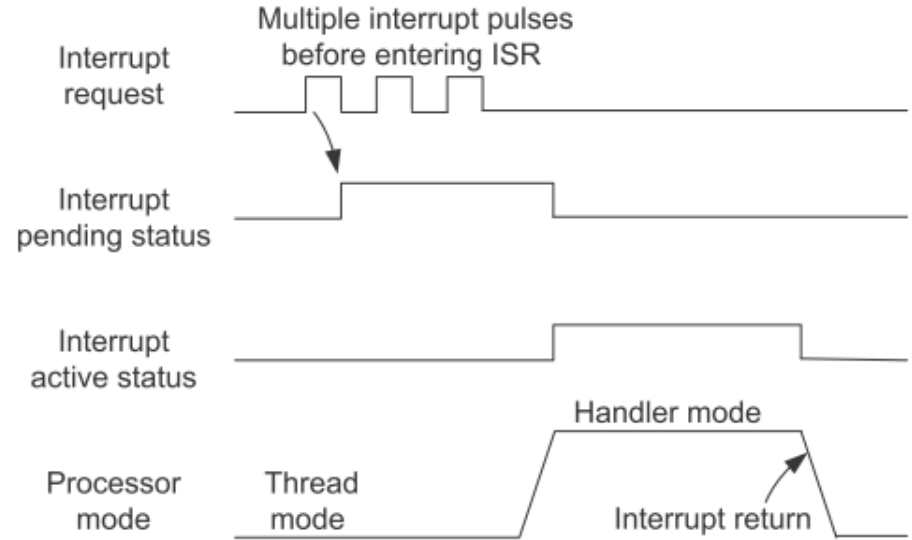
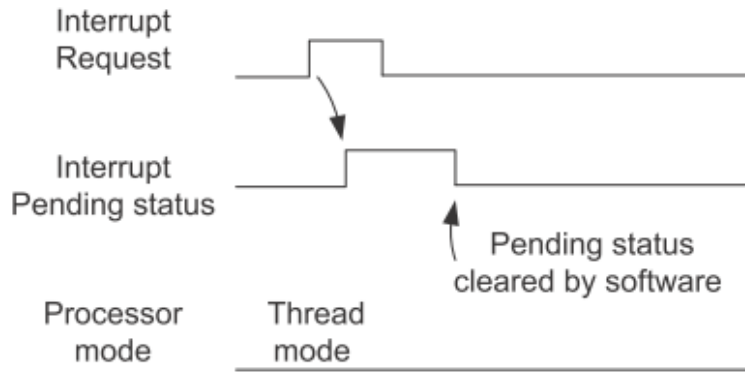
Procedura ISR zostanie wykonana gdy:

- dane przerwanie jest włączone,
- zostanie przyjęte i oczekuje na obsługę (ma „pending status”),
- przerwanie ma wyższy priorytet niż program aktualnie wykonywany.

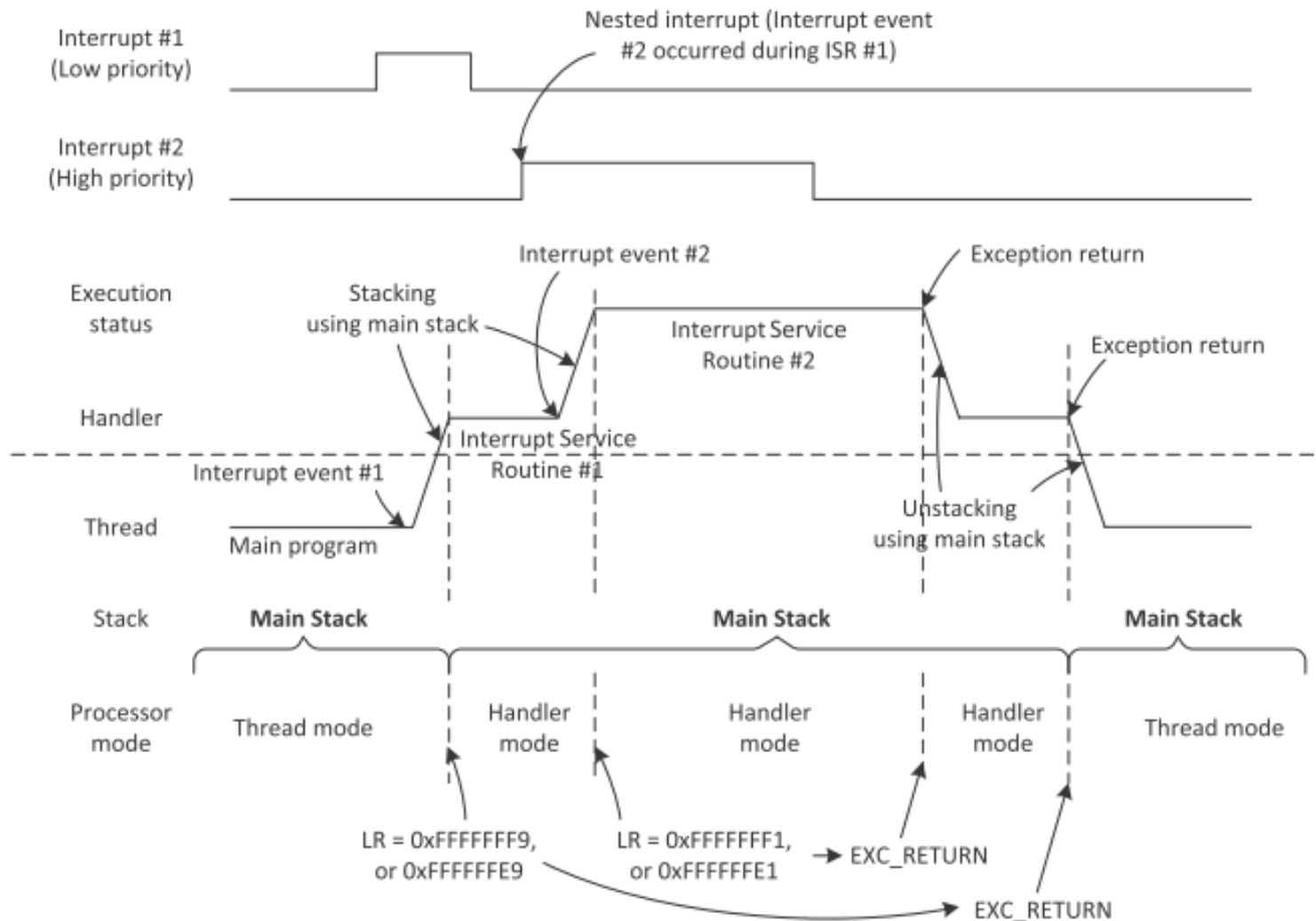


Równoległe zachowywanie rejestrów na stosie (RAM) oraz pobieranie wektora (adresu) ISR (Flash – ROM).

Przerwania w CORTEX M4 - różne szczególne przypadki



Przerwania w CORTEX M4 - priorytety



- Poza szczególnymi przypadkami procedura ISR nie powinna zawierać elementów „blokujących” możliwość powrotu do kontynuacji innych zadań).
- Czas wykonania ISR należy zminimalizować.

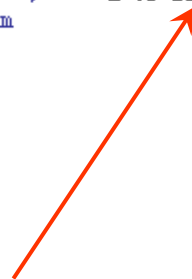
Przerwania w CORTEX M4 – wywołanie ISR

Pomijając sprawdzanie priorytetów, sama idea wywołania (skoku do) ISR jest podobna do wywołania podprogramu (biblioteki) – po powrocie musi być zapewniona możliwość kontynuowania przerwanego w dowolnym momencie programu.

- Procesor kończy aktualnie wykonywaną instrukcję*,
- przerywany jest aktualnie wykonywany ciąg instrukcji,
- rozpoczynając obsługę ISR Cortex M4 **automatycznie odkłada na stos rejestry: R0-R3, R12, LR**, PC,**
- w tym samym czasie jest pobierany wektor przerwania (rozdzielone szyny i bloki pamięci danych i programu - arch. Harvardzka),
- procesor zapisuje **adresu powrotu w **Link Register** (LR, w odróżnieniu do x86),
- kończąc ISR ściąga ze stosu zachowane wartości, zaczynając od PC (i może jednocześnie pobierać do potoku instrukcje kontynuowanego zadania).

Liczba cykli maszynowych

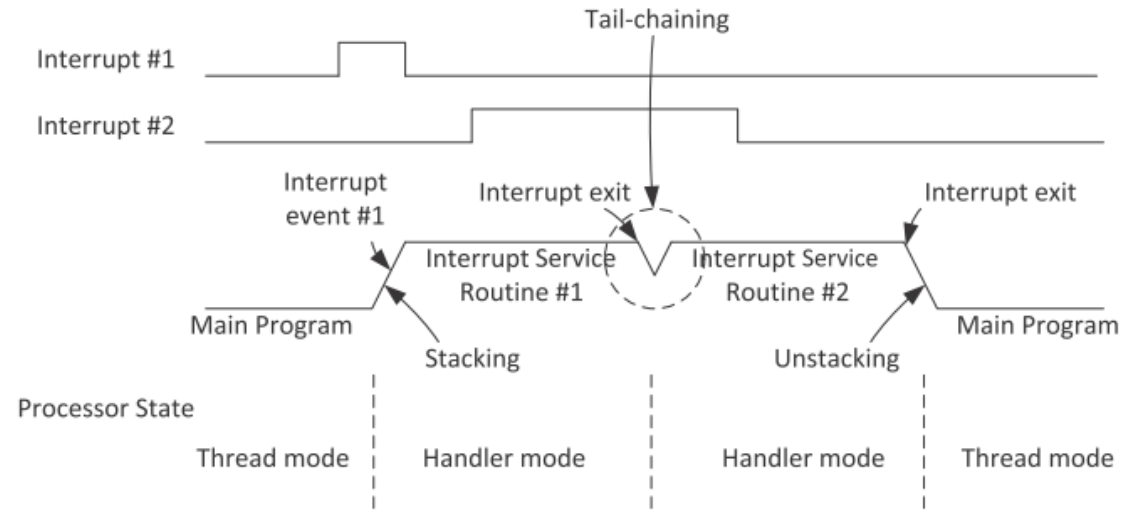
Multiply	Multiply	MUL Rd, Rn, Rm	1	
	Multiply accumulate	MLA Rd, Rn, Rm	1	
	Multiply subtract	MLS Rd, Rn, Rm	1	
	Long signed	SMULL RdLo, RdHi, Rn, Rm	1	
	Long unsigned	UMULL RdLo, RdHi, Rn, Rm	1	
	Long signed accumulate	SMLAL RdLo, RdHi, Rn, Rm	1	
	Long unsigned accumulate	UMLAL RdLo, RdHi, Rn, Rm	1	
	Divide	Signed	SDIV Rd, Rn, Rm	2 to 12
Unsigned		UDIV Rd, Rn, Rm	2 to 12	



*niektóre instrukcje wielocyklowe: LDM/STM (Load/Store Multiple - ładowanie wielu rejestrów z podanej listy, dzielenie, VSQRT) zostają „zawieszane” na czas wykonania ISR, a po zakończeniu obsługi przerwania – kontynuowane (równolegle wraz z np. obsługą stosu).

Przerwania w CORTEX M4 - optymalizacja i skrócenie czasu obsługi

Tail chaining



Late arrival

