

Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Eletrônica (DAELN)

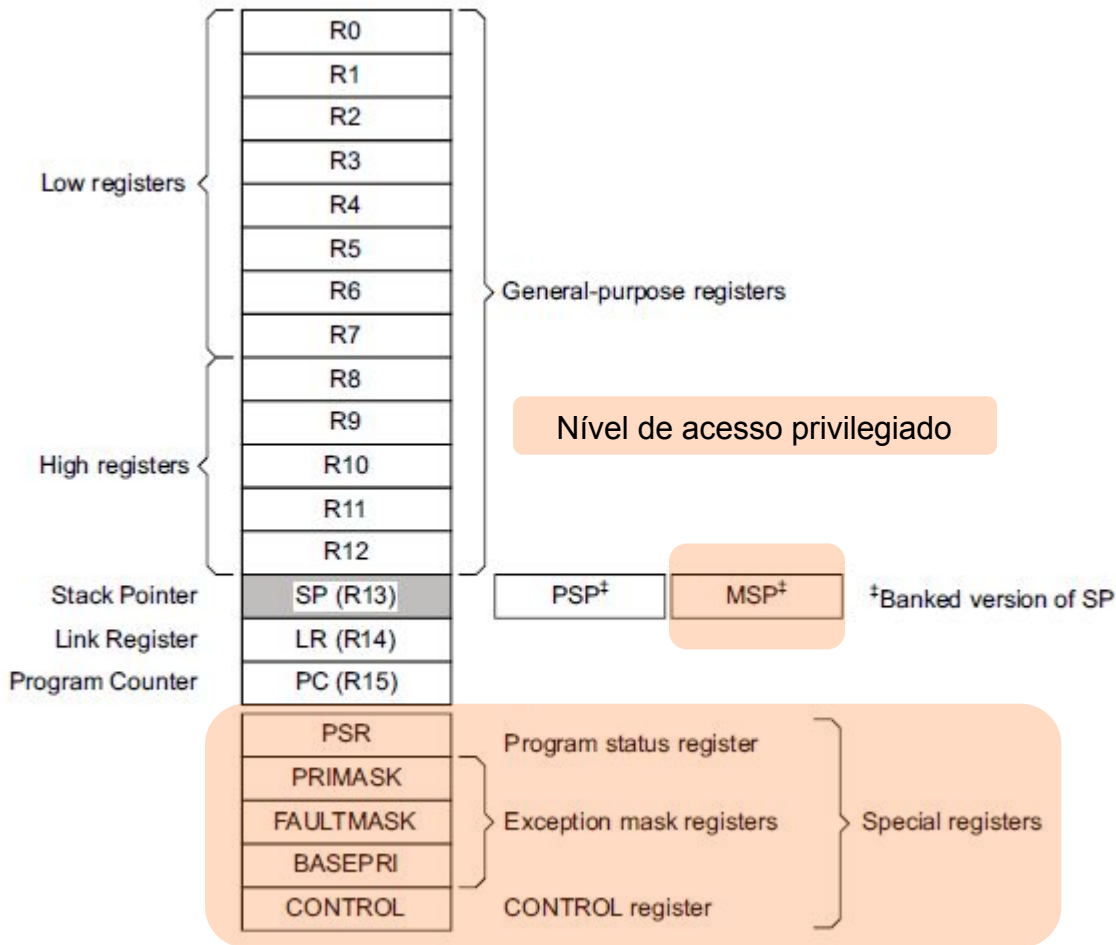
SISTEMAS EMBARCADOS

Registradores, Instruções Especiais, Exceções e Memórias

Prof. André Schneider de Oliveira

andreoliveira@utfpr.edu.br

Registradores



MSP = main SP
Kernel do S.O.
Exceções

PSP = process SP
Aplicações (threads)

PRIMASK
Desabilita interrupções

CONTROL
Seleciona modo
Seleciona pilha

Stack Pointer

- Pode-se trabalhar apenas com o MSP
- O SP atual é acessado como R13 ou SP
- O SP sempre está alinhado em 32 bits (i.e. endereços múltiplos de 4)
- Instruções:
 - PUSH (empilha)
 - POP (desempilha)
- Full descending stack

Link Register

- Acessado como R14 ou LR
- Armazena o endereço de retorno de uma subrotina
- Deve ser salvo antes de se chamar outra subrotina

Contador de Programa

- Acessado como R15 ou PC
- PC contém endereço da instrução sendo executada + 2 ou 4 (dependendo do tamanho da instrução atual)
- O endereço de uma instrução é sempre par (bit0 = 0)
- Bit0 do PC é usado para indicar modo Thumb
 - Em saltos, o bit 0 do PC deve ser sempre 1, caso contrário uma exceção será gerada pelo Cortex-M3.

Registradores Especiais

- Program Status Registers (xPSR):
 - Application Program Status Register (APSR) – RW
 - Interrupt Program Status Register (IPSR) – RO
 - Execution Program Status Register (EPSR) – RO
- Acesso por meio de instruções específicas:
 - MRS R0, APSR
 - MSR APSR, R0

Program Status Registers (PSR)

	31	30	29	28	27	26:25	24	23:20	19:16	15:10	9	8	7	6	5	4:0
APSR	N	Z	C	V	Q											
IPSR												Exception number				
EPSR						ICI/IT	T				ICI/IT					

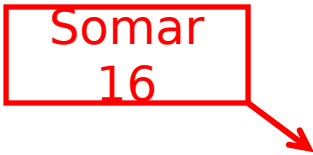
	31	30	29	28	27	26:25	24	23:20	19:16	15:10	9	8	7	6	5	4:0
xPSR	N	Z	C	V	Q	ICI/IT	T				ICI/IT		Exception number			

Bit	Description
N	Negative
Z	Zero
C	Carry/borrow
V	Overflow
Q	Sticky saturation flag
ICI/IT	Interrupt-Continuable Instruction (ICI) bits, IF-THEN instruction status bit
T	Thumb state, always 1; trying to clear this bit will cause a fault exception
Exception number	Indicates which exception the processor is handling

Interrupt Program Status Register (IPSR)

- “Exception number” é diferente de “IRQ number 16+”!

Bits	Name	Function
[31:6]	-	Reserved
[5:0]	Exception number	This is the number of the current exception: 0 = Thread mode 1 = Reserved 2 = NMI 3 = HardFault 4-10 = Reserved 11 = SVCcall 12, 13 = Reserved 14 = PendSV 15 = SysTick 16 = IRQ0 . . . 47 = IRQ31 48-63 = Reserved. see Section 22–22.3.3.2 for more information.



Registadores Especiais

- Interrupt Mask Registers:
 - PRIMASK: usado para habilitar/desabilitar todas as interrupções e exceções (exceto NMI e Hard Fault)
 - FAULTMASK: usado para habilitar/desabilitar exceções de falta
 - BASEPRI: define o mascaramento de interrupções abaixo de um limiar de prioridade
- Acesso pelas instruções MRS e MSR
 - PRIMASK somente em nível de acesso privilegiado
- Instruções
 - CPSIE: change processor state
 - CPSID: interrupt enable/disable)

Registradores Especiais

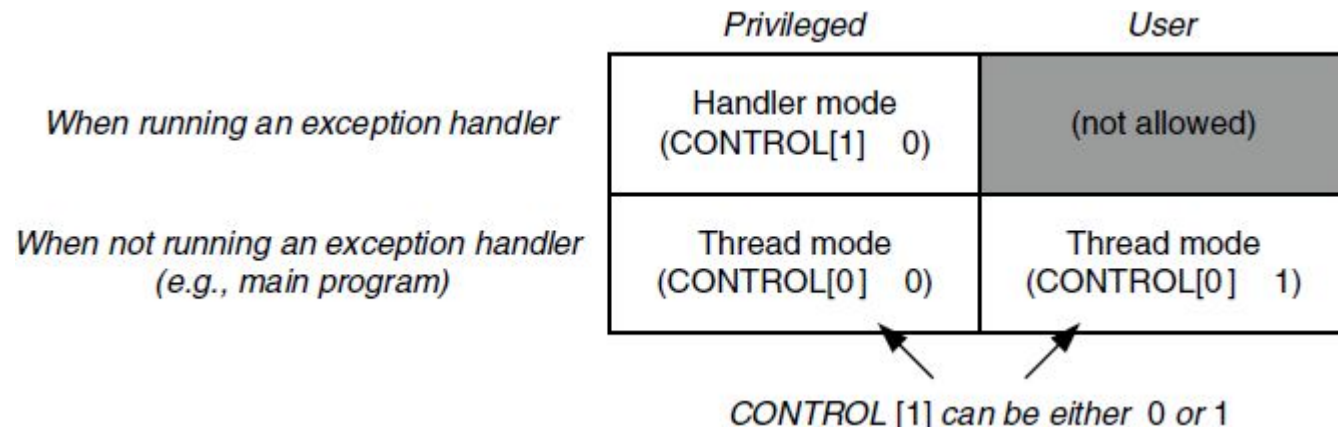
- CONTROL
 - Bit 1: define o ponteiro de pilha (MSP ou PSP)
 - Bit 0: define o nível de acesso (Privilegiado ou Usuário)
- O bit 0 possui acesso de escrita apenas em nível privilegiado – uma vez em nível usuário, a única forma de retornar ao nível privilegiado é por meio de uma interrupção.
- Acesso pelas instruções MRS e MSR
 - Usar instrução ISB após a instrução MSR para garantir uso imediato da nova pilha

Modos, Privilégios e Pilhas

- Thread Mode / Handler Mode
 - Modo Thread: execução normal de aplicações
 - Modo Handler: exceções ou interrupções
- Operação privilegiada / não privilegiada
 - Modo Thread: pode ser privilegiada ou não
 - Modo Handler: sempre privilegiada
- Main Stack / Process Stack
 - Ambas as pilhas possuem seu próprio ponteiro
 - Exceções sempre utilizam MSP em modo Handler
 - Aplicações (modo Thread) utilizam MSP ou PSP

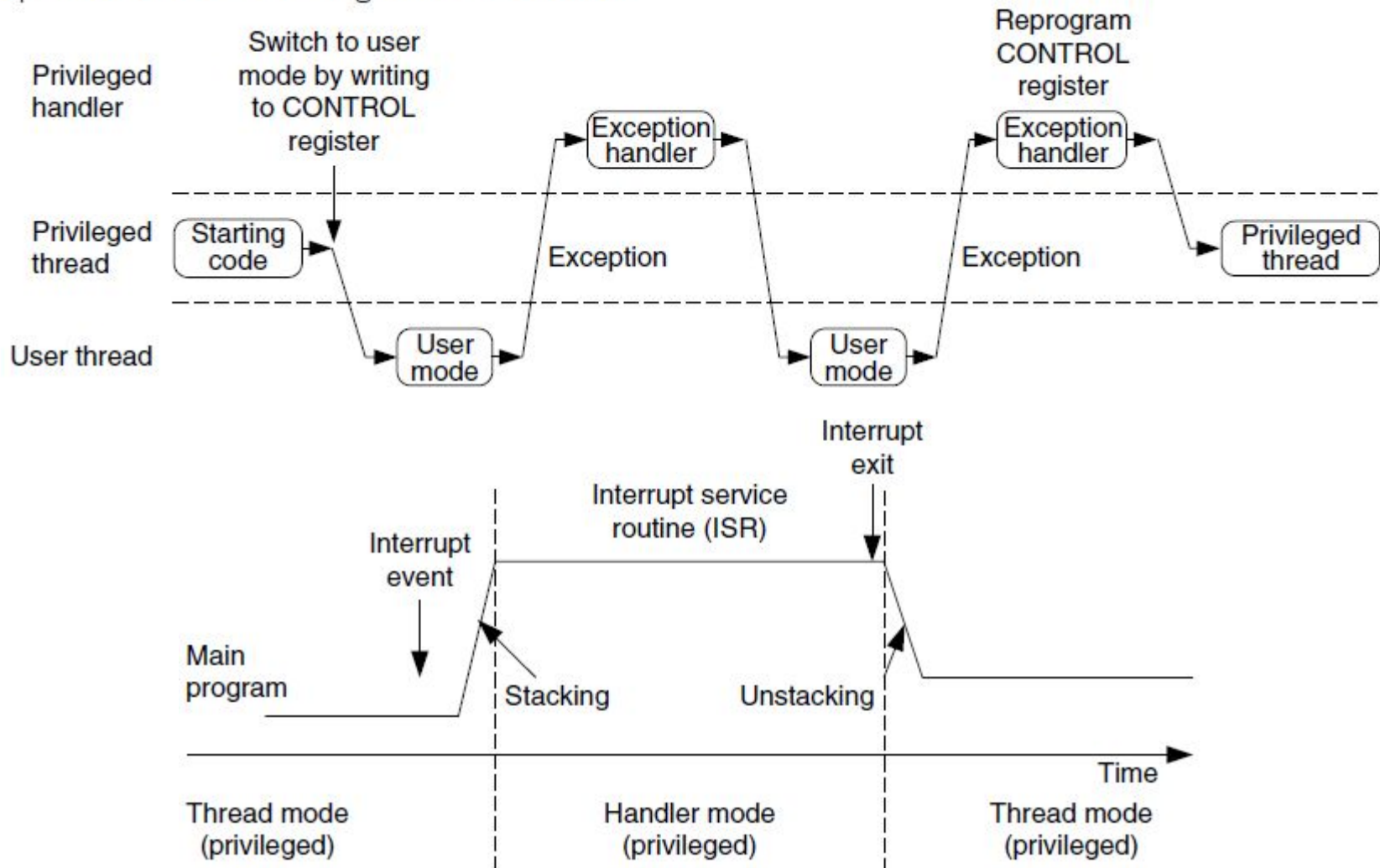
Modos, Privilégios e Pilhas

		Operations (privilege out of reset)	Stacks (Main out of reset)
Modes (Thread out of reset)	Handler - An exception is being processed	Privileged execution Full control	Main Stack Used by OS and Exceptions
	Thread - No exception is being processed - Normal code is executing	Privileged/Unprivileged	Main/Process



Modos e Privilégios

Operation Modes and Privilege Levels in Cortex-M3.



Privilégios

- No nível Privilegiado o código tem acesso a TODOS os recursos.
- No nível usuário o código NÃO tem acesso a:
 - Instruções como CPS que permitiriam alterar FAULTMASK e PRIMASK
 - A maioria dos registradores do System Control Block

Carga de Registradores Especiais

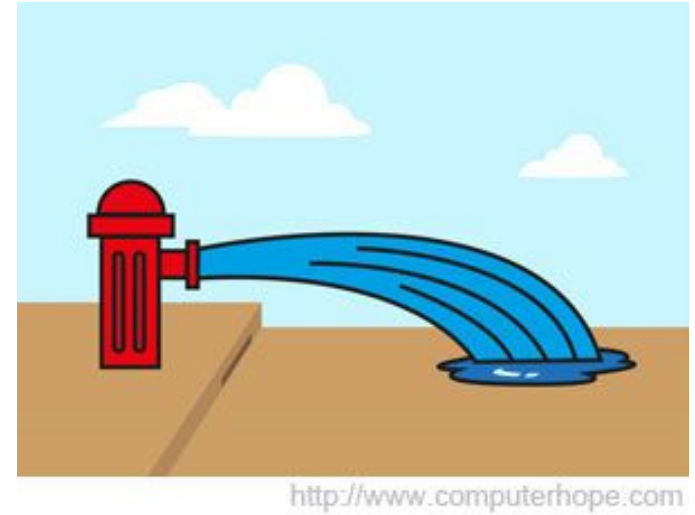
- MSR (Move to special register from register)
 - MSR special_reg, Rn
- MRS (Move to register from special register)
 - MRS Rd, special_reg
- Special Register:
APSR, IPSR, EPSR, IEPSR, IAPSR, EAPSR, PSR, MSP,
PSP, PRIMASK ou CONTROL
- Rd, Rn:
Não podem ser SP ou PC

Barreiras de Memória

- DMB (Data Memory Barrier)
 - Completa transações pendentes antes de realizar outra transação
- DSB (Data Synchronization Barrier)
 - Completa transações pendentes antes de executar a próxima instrução
- ISB (Instruction Synchronization Barrier)
 - Garante que o efeito das transações efetuadas seja percebido antes da próxima instrução (pipeline flush)

Pipeline Flush

- Pipeline flush ou "quebra de pipeline"
 - procedimento executado quando se é incapaz de garantir o correto processamento do pipeline no próximo ciclo de clock.
- se duas instruções pendentes forem dependentes dos dados (elas dependem ou modificam o mesmo conjunto de dados na memória)
- a CPU pode precisar limpar (ou "flush") o pipeline de instruções para garantir que os cálculos não sejam corrompidos pelo processo pipelining.



Ordem de Acessos à Memória

- A ordem das instruções no fluxo do programa nem sempre garante a ordem de transações na memória, pelos seguintes motivos:
 - O processador pode reordenar alguns acessos à memória para melhorar a eficiência, desde que isto não afete o comportamento da sequência de instruções
 - Dispositivos no mapa de memória podem necessitar de números diferentes de estados de espera (wait states)
 - Alguns acessos à memória são intermediados (buffer de escrita, cache externo) ou especulativos
- Se a ordem de acessos à memória for crítica, o software deve contar com instruções de barreira de memória (DMB, DSB, ISB) para garantir essa ordem

Exemplos – Barreiras de Memória

- Tabela de vetores
 - Usar DMB entre a alteração de um vetor e a habilitação da IRQ correspondente
- Alteração no mapa de memória
 - Usar DSB imediatamente após a instrução que altera o mapa de memória
- Alteração no código
 - Usar ISB logo após a instrução que altera o código

Outras Instruções

- NOP (no operation)
 - Utilizada para alinhamento de código
 - Pode ser eliminada no pipeline (i.e. execução em tempo 0)
- SVC (Supervisor Call)
 - Chamada do supervisor (S.O.)
 - Gera uma exceção do tipo SVC

Outras Instruções

- WFI (Wait for Interrupt)
 - Entra em modo de baixo consumo até a ocorrência de uma exceção, IRQ (mesmo que mascarada pelo PRIMASK) ou solicitação de DEBUG
- WFE (Wait for Event)
 - Entra em modo de baixo consumo até a ocorrência de um evento (usada com SEV em sistemas multicore)
- SEV (Send Event)
 - Informa a ocorrência de um evento

Exceções

- Qualquer solicitação para mudança do fluxo normal de um programa:
 - Interrupção
 - Detecção de falta / erro

Visão Geral – Cortex-M

- NVIC (Nested Vector Interrupt Controller)
 - Suporte a múltiplas fontes de interrupção
 - Tratamento eficiente de interrupções aninhadas
 - Arquitetura flexível (altamente configurável)
 - Suporte intrínseco a RTOS

Visão Geral – Cortex-M

- Arquitetura de interrupções com baixa latência
- Algumas instruções com múltiplos ciclos de execução podem ser interrompidas
- Entrada/saída de interrupções controlada por hardware
 - Salvamento e restauração de contexto realizado automaticamente
 - Tratamento de chegada tardia (**late arriving**) de interrupções com maior prioridade
 - Tratamento de interrupções pendentes sem restauração/salvamento completo de contexto (**tail-chaining**)

Estados das Exceções

- Inativo: nem pendente nem ativo
- Pendente: a exceção foi gerada, mas ainda não foi processada
- Ativo: o processamento da exceção foi iniciado, mas ainda não foi completado
 - O processamento de uma exceção pode interromper o processamento de outra
 - nesse caso ambas exceções estão no estado ativo
- Ativo e Pendente: a exceção está sendo processada e existe uma exceção pendente da mesma fonte

Tipos de Exceção

- Reset - interrupção durante power up or warm reset prioridade -3 (modo Thread)
- NMI - interrupção não-mascarável sinalizada por periférico ou acionada por SW (não-mascarável) – prioridade -2
- Hard Fault - interrupção por erro de execução - prioridade -1
- SVCcall - interrupção iniciada por instrução de supervisão
- Debug Monitor - interrupção por debug
- PendSV - interrupção no nível de serviço (ex. troca de contexto em SO)
- SysTick Interrupt - interrupção do system timer
- External Interrupt - interrupção externa
- IRQ – pedido de interrupção por periférico

Tipos de Exceção de Falta

- **MemManageFault:** faltas no acesso a memória detectadas pela MPU – se desabilitada, escala para HardFault
- **BusFault:** outros tipos de falta no barramento de memória que não as do tipo MemManage
- **UsageFault:** faltas não relacionadas ao barramento de memória (ex. instrução indefinida ou estado inválido)

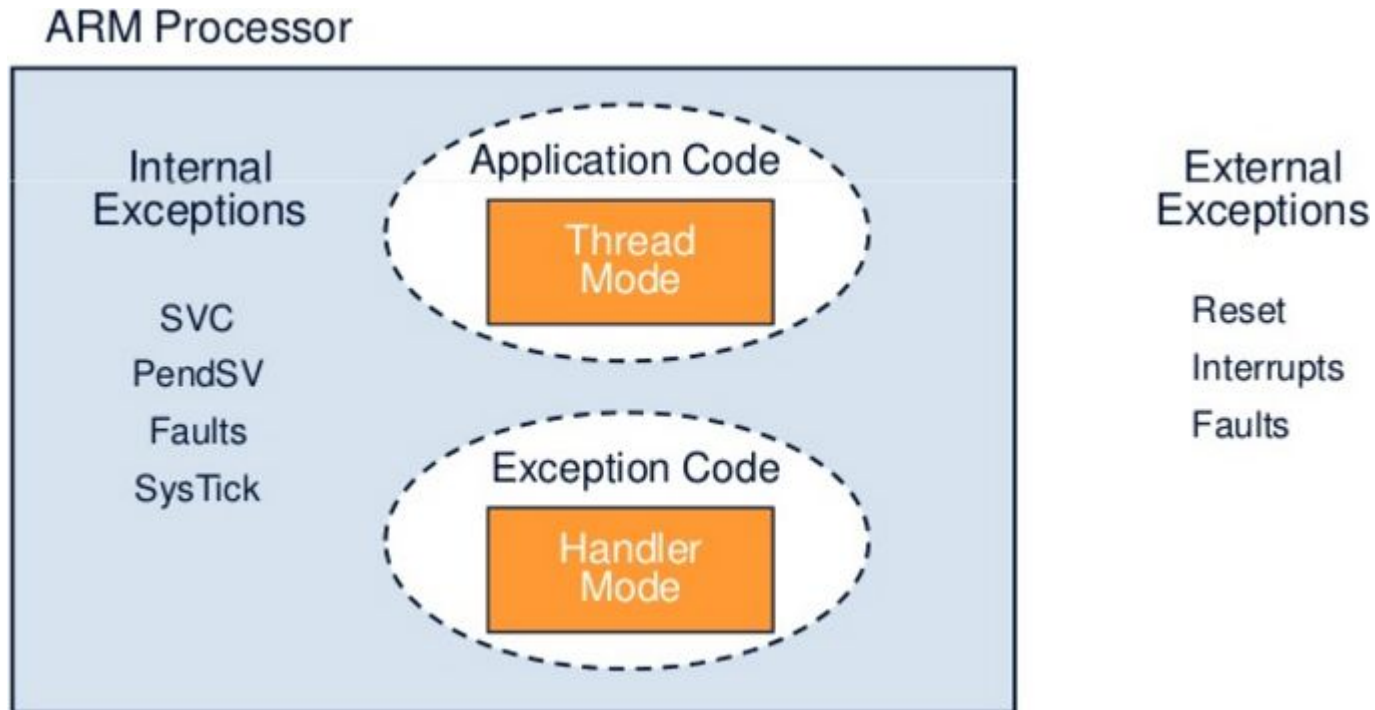
Propriedades dos Tipos de Exceção

Properties of the different exception types

Exception number ^[1]	IRQ number ^[1]	Exception type	Priority	Vector address or offset ^[2]	Activation
1	-	Reset	-3, the highest	0x00000004	Asynchronous
2	-14	NMI	-2	0x00000008	Asynchronous
3	-13	Hard fault	-1	0x0000000C	-
4	-12	Memory management fault	Configurable ^[3]	0x00000010	Synchronous
5	-11	Bus fault	Configurable ^[3]	0x00000014	Synchronous when precise, asynchronous when imprecise
6	-10	Usage fault	Configurable ^[3]	0x00000018	Synchronous
7-10	-	-	-	Reserved	-
11	-5	SVCall	Configurable ^[3]	0x0000002C	Synchronous
12-13	-	-	-	Reserved	-
14	-2	PendSV	Configurable ^[3]	0x00000038	Asynchronous
15	-1	SysTick	Configurable ^[3]	0x0000003C	Asynchronous
16 and above	0 and above	Interrupt (IRQ)	Configurable ^[4]	0x00000040 and above ^[5]	Asynchronous

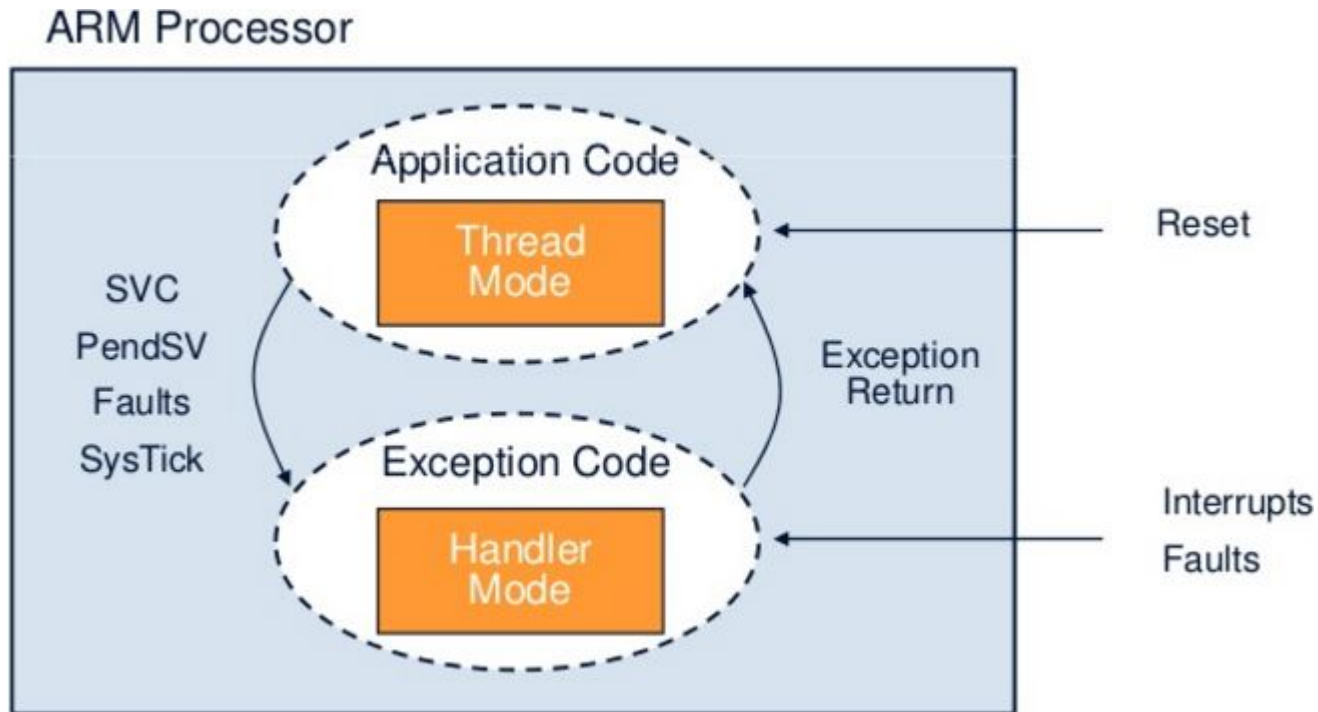
Eventos Causadores

- Exceções podem ser causadas por eventos internos ou externos ao processador



Modo de Processamento

- Modo Thread após Reset
- Modo Handler após qualquer outro tipo de exceção

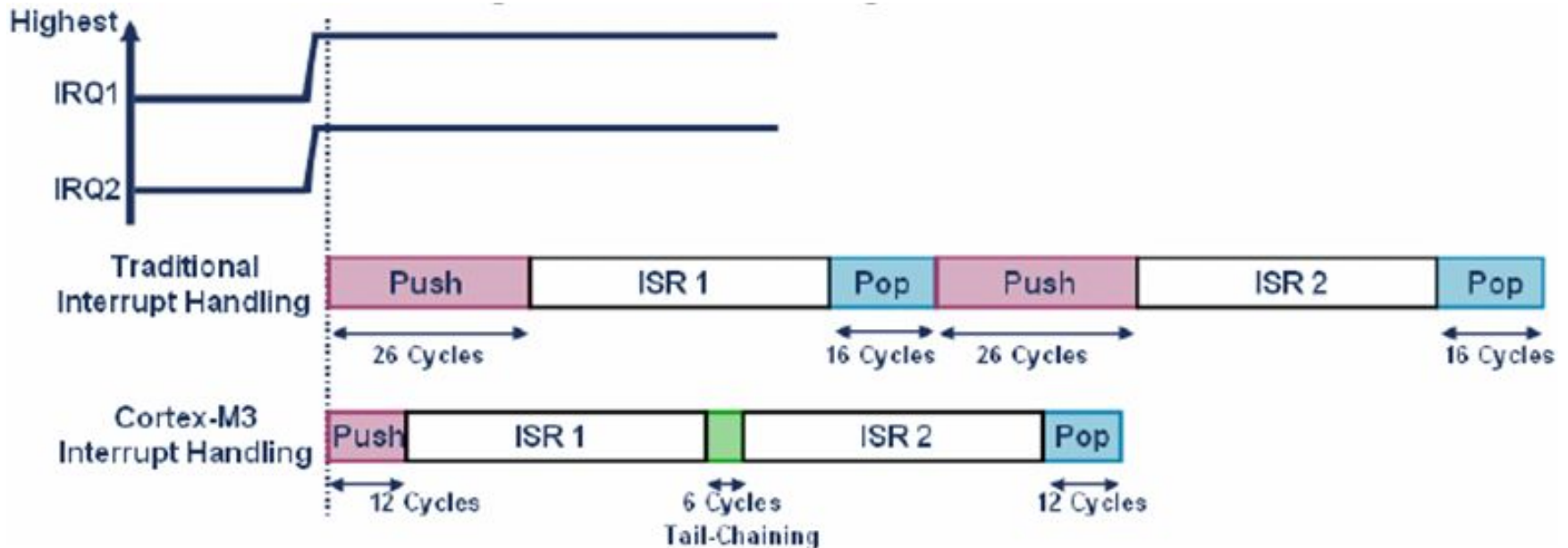


Definições Importantes

- **Preempção:** uma exceção de maior prioridade pode causar preempção de uma rotina de tratamento (handler) em execução
- **Retorno:** ocorre ao término da execução da rotina de tratamento (**handler**), caso:
 - Não exista exceção pendente com prioridade suficiente para ser atendida
 - A rotina que terminou não seja de uma exceção com chegada tardia (**late arriving**)

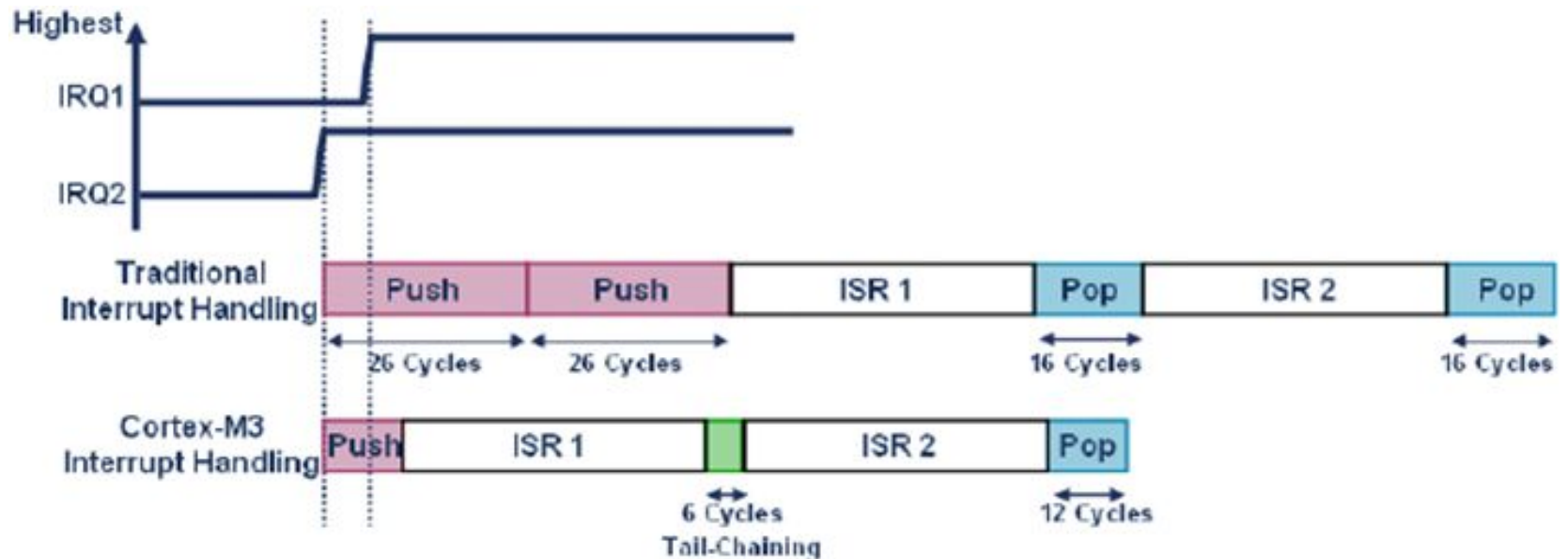
Definições Importantes

- **Tail-chaining (encadeamento)**: se ao término de uma handler existir uma exceção pendente apta a ser tratada então não se executa a sequência de restauração/salvamento de contexto (POP/PUSH)



Definições Importantes

- **Late arriving (chegada tardia)**: mecanismo que acelera preempção caso uma exceção de maior prioridade ocorra durante o salvamento de contexto (PUSH)
 - nesse caso a exceção de maior prioridade é atendida primeiro e em seguida, por tail-chaining, a de menor prioridade.



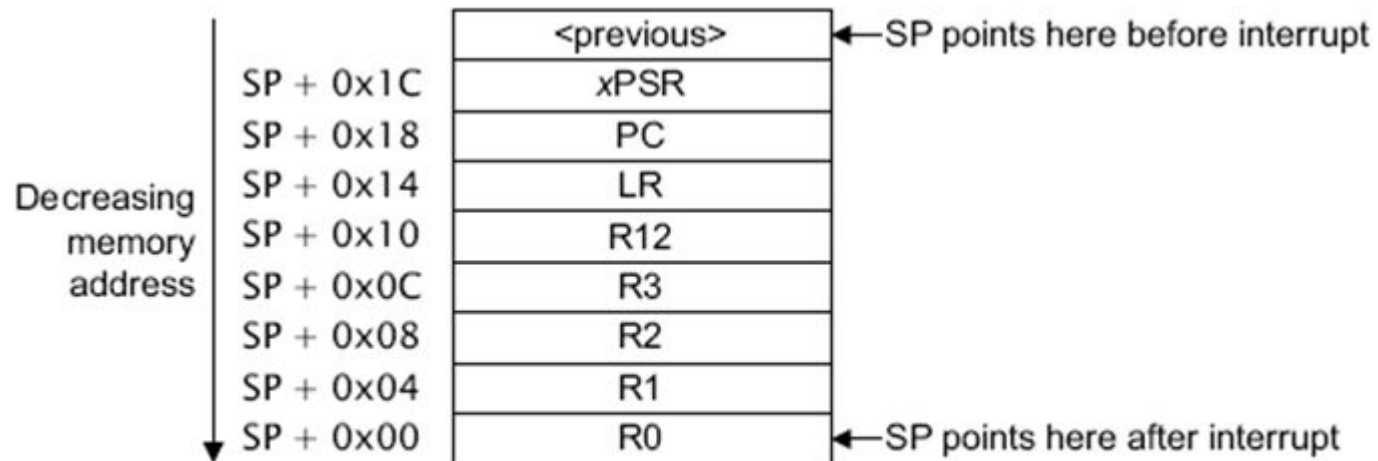
Atendimento de Exceções

- Ocorre quando existe uma exceção no estado pendente com prioridade suficiente* e
 - O processador está em modo Thread, ou
 - Essa exceção pendente possui prioridade maior que a que uma exceção que esteja sendo atendida (preempção em modo Handler)

*Prioridade suficiente significa prioridade maior do que o limiar estabelecido em BASEPRI

Atendimento de Exceções

- Ocorre salvamento automático de contexto (**stacking**) na entrada, bem como recuperação automática de contexto na saída



Exception entry stack contents

Atendimento de Exceções

- O processador obtém o endereço da rotina de tratamento (**handler**) a partir da `vector_table`

Address		Exception #
$0x40 + 4*N$	External N	$16 + N$
...
0x40	External 0	16
0x3C	SysTick	15
0x38	PendSV	14
0x34	Reserved	13
0x30	Debug Monitor	12
0x2C	SVC	11
0x1C to 0x28	Reserved (x4)	7-10
0x18	Usage Fault	6
0x14	Bus Fault	5
0x10	Mem Manage Fault	4
0x0C	Hard Fault	3
0x08	NMI	2
0x04	Reset	1
0x00	Initial Main SP	N/A

Atendimento de Exceções

- Após o stacking, a execução se dá a partir da rotina de tratamento (**handler**) e a exceção passa ao estado ativo
- Se ocorrer outra exceção de maior prioridade durante o stacking (**late-arriving exception**), então esta será atendida antes
- LR := EXC_RETURN (próximo slide)

Números Mágicos de Retorno (LR)

Exception return behavior

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 gets state from MSP. Execution uses MSP after return.
0xFFFFFFFDD	Return to Thread mode. Exception return gets state from PSP. Execution uses PSP after return.
All other values	Reserved.

Retorno de Exceções

- Ocorre quando uma instrução escreve no PC um dos valores de EXC_RETURN
- Instruções usadas para retorno de exceção:
 - LDR PC, ...
 - LDM/POP incluindo PC
 - BX LR (preferencial)