

ARM Cortex M4

Intro

Núcleos ARM Cortex-M™ cores

- **Esqueça a tradicional classificação 8/16/32-bit e obtenha**
 - Arquitetura semelhante através de todas aplicações
 - Cada produto otimizado para baixíssimo consumo

Cortex-M0/M0+

Aplicações 8/16-bit

Cortex-M3

Aplicações 16/32-bit

Cortex-M4

Aplicações 32-bit/DSC

Compatibilidade de código binário e ferramentas



STM32 F0 Value line

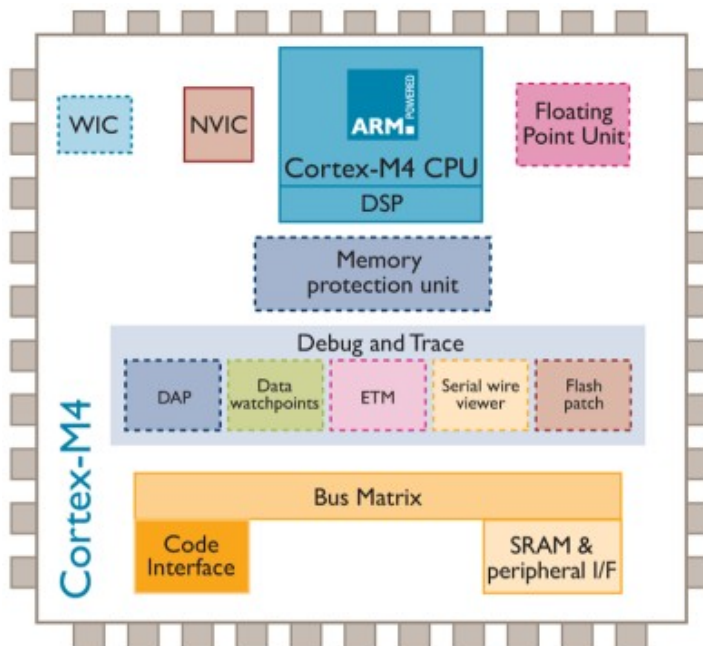


Cortex™

Low-Power Leadership from ARM

32-bit Core

- STM32 F4 - Cortex-M4
- STM32 F3 - Cortex-M4
- STM32 F2 - Cortex-M3
- STM32 F1 - Cortex-M3
- STM32 F0 - Cortex-M0
- STM32 L0 – Cortex-M0+
- STM32 L1 - Cortex-M3



Processadores Cortex-M compatibilidade binária

Ponto Flutuante(FPU)

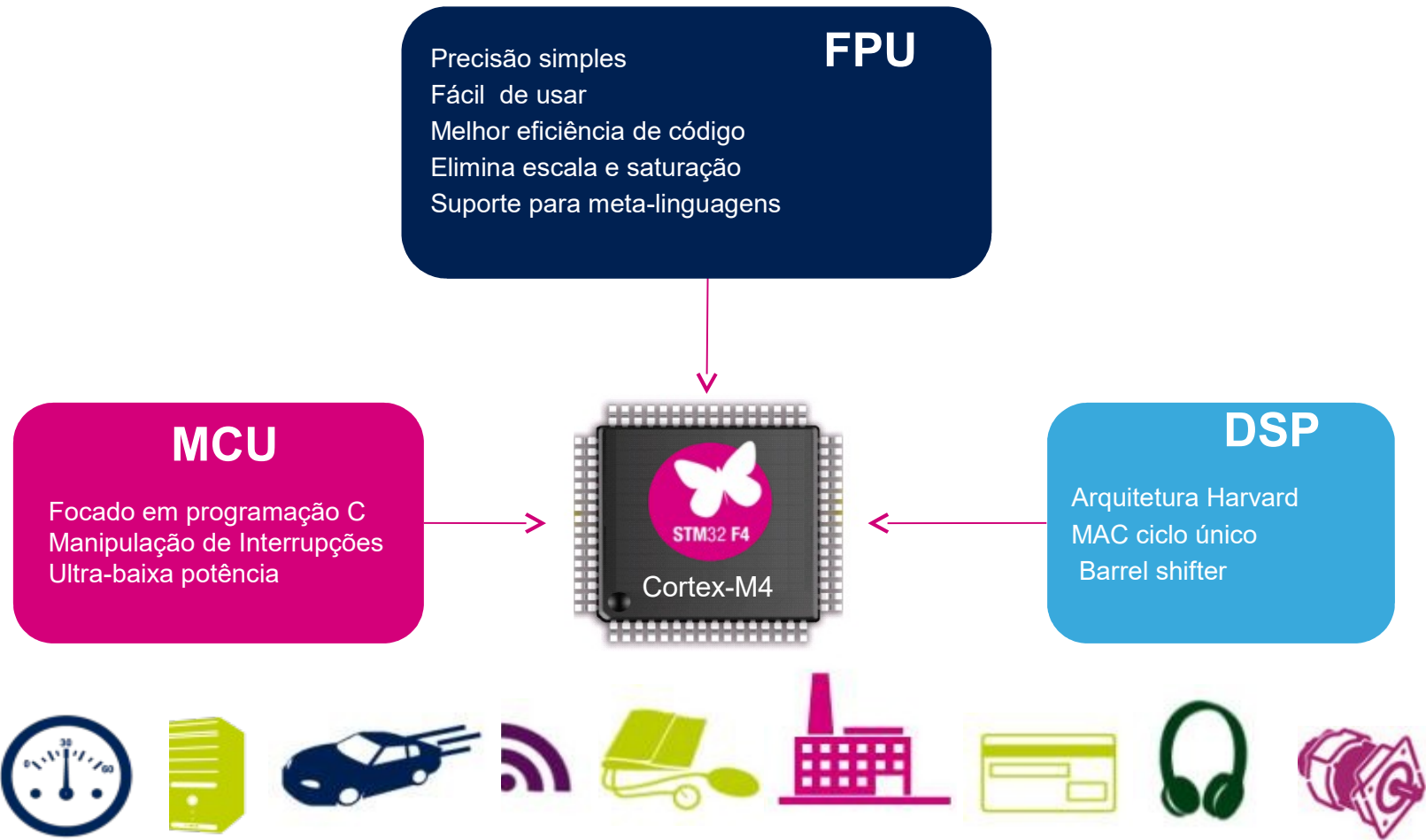
DSP (SIMD, fast MAC)

Processamento avançado
Manipulação de campo de bits

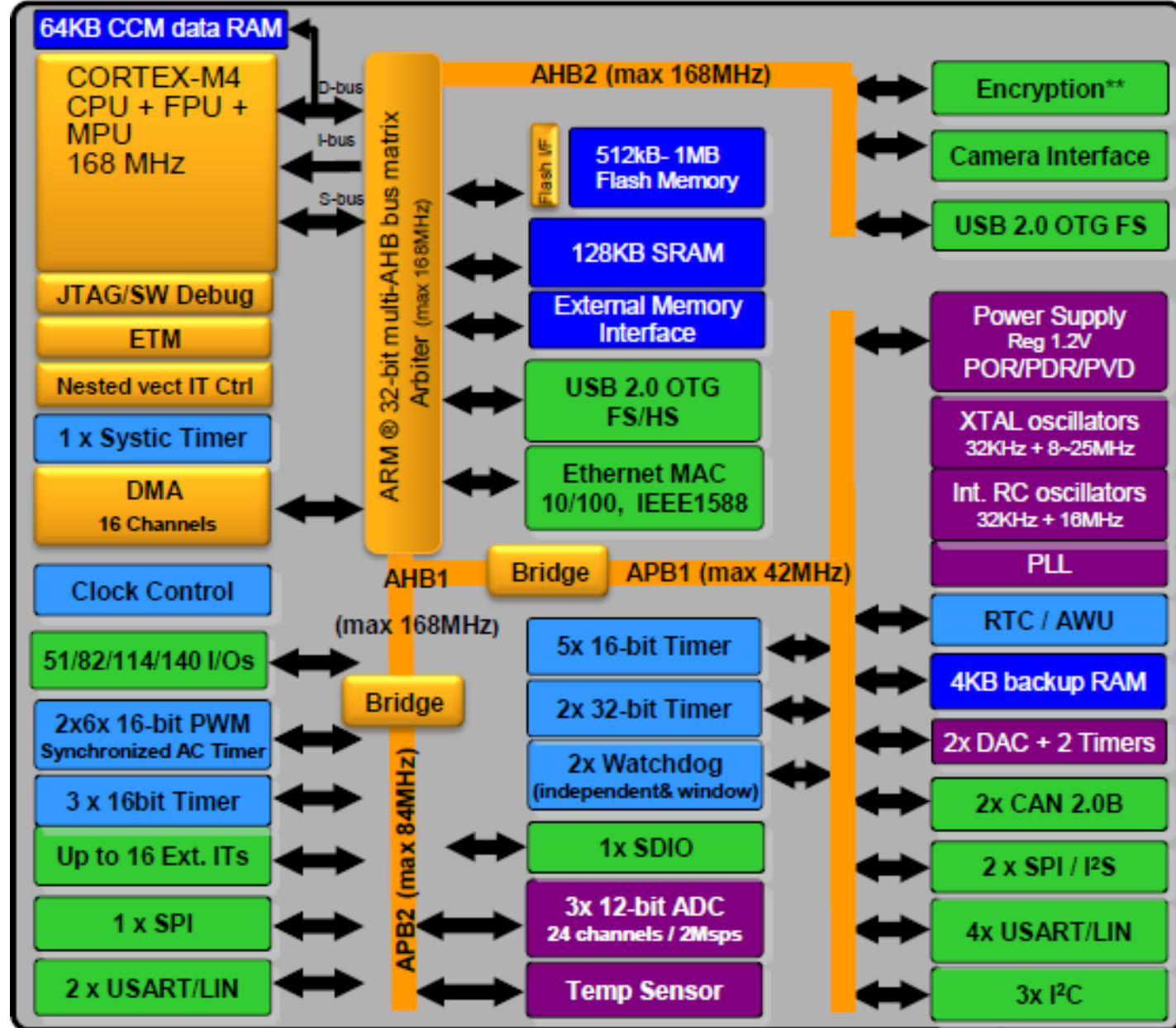
Processamento geral
Tarefas controle I/O

VABS	VADD	VCMP	VCMPE	VCVT	VCVTR	VDIV	VLDH
VLDL	VMLA	VMLS	VMOV	VMRS	VMSR	VMUL	VNEG
VNMLA	VNMLS	VNMUL	VPOP	VPUSH	VSQRT	VSTH	VSTR
VSUB	VFMA	VFMS	VFNMA	VFNMS	Cortex-M4 FPU		
PKH	QADD	QADD16	QADD8	QASX	QDADD	QDSUB	QSAX
QSUB	QSUB16	QSUB8	SADD16	SADD8	SASX	SEL	SHADD16
SHADD8	SHASX	SHSAX	SHSUB16	SHSUB8	SHLALB	SHLALB	SHLATB
SHLATT	SMLAD	SMLALB	SMLALB	SMLALB	SMLALB	SMLALD	SMLAWB
SMLAWT	SMLSD	SMLSDB	SMMLA	SMMLS	SMML	SMUAD	SMULB
ADC	ADD	ADR	AND	ASR	B	SHLBT	SHULTT
CLZ	BFC	BFI	BIC	CDP	CLREX	SHULTB	SHULWT
CBNZ	CBZ	CMN	CMR	DRG	EOR	SHULWB	SHUSD
LDHIA	LDHDB	LDR	LDRB	LDRBT	LDRD	SSAT16	SSAX
LDREX	LDREXB	LDREXH	LDRH	LDRHT	LDRSB	SSUB16	SSUB8
LDRSBT	LDRSHT	LDRSH	LDRT	MCR	LSL	SXTAB	SXTAB16
LSR	MCRB	MLS	MLA	MOV	MOVT	SXTAH	SXTAB16
MRC	MRRB	MUL	MYN	NOP	ORN	UADD16	UADD8
ORR	PLD	PLDW	PLI	POP	PUSH	UASX	UHADD16
RBIT	REV	REV16	REVSH	ROR	RUB	UHADD8	UHASX
BKPT	BLX	ADC	ADD	ADR	RSB	UHADD8	UHSAX
BX	CPS	AND	ASR	B	SDIV	UHSAX	UHSUB16
DHR	BL	BIC	SHLL	SSAT	SHLL	UHSUB8	UHAAL
DSB	CMN	CMR	STHIA	STHDB	STR	UQADD16	UQADD8
ISB	LDR	LDRB	STRB	STRBT	STRD	UQASX	UQASX
MRS	LDRH	LDRSB	STREX	STREXB	STREXH	UQSUB16	UQSUB8
MRR	LSL	LSR	STRH	STRHT	STRT	USAD8	USADA8
NOP	REV	MUL	SUB	SXTB	SKTH	USAT16	USAX
REV16	REVSH	POP	TBB	TBH	TEQ	USUB16	USUB8
SEV	SXTB	RSB	TST	UBFX	UDIV	UXTAB	UXTAB16
SXTH	UXTB	STR	UHLAL	UHLL	USAT	UXTAH	UXTB16
UXTH	WFE	SUB	UXTB	UXTH	WFE		
WFI	YIELD		WFI	YIELD	IT		
Cortex-M0/M0+/M1				Cortex-M3			Cortex-M4

ARM Cortex™-M4 Core



ARM Cortex M4 Core



ARM Cortex M4 Core STM32F411

Core ARM M4 32 bits
Cortex com unidade de
ponto flutuante

512 Kbytes memória

Flash

128 Kbytes RAM

Gerenciamento de clock,
reset e alimentação

Conversor A/D de 12 bits,
2,4Msps e até 16 canais

16 controladores de DMA
de uso geral

RTC

Até 11 Timers :

6 de 16 bits,

2 de 32 bits,

2 watchdogs

SysTick

Até 81 pinos de E/S

13 interfaces de comunicação

I2C

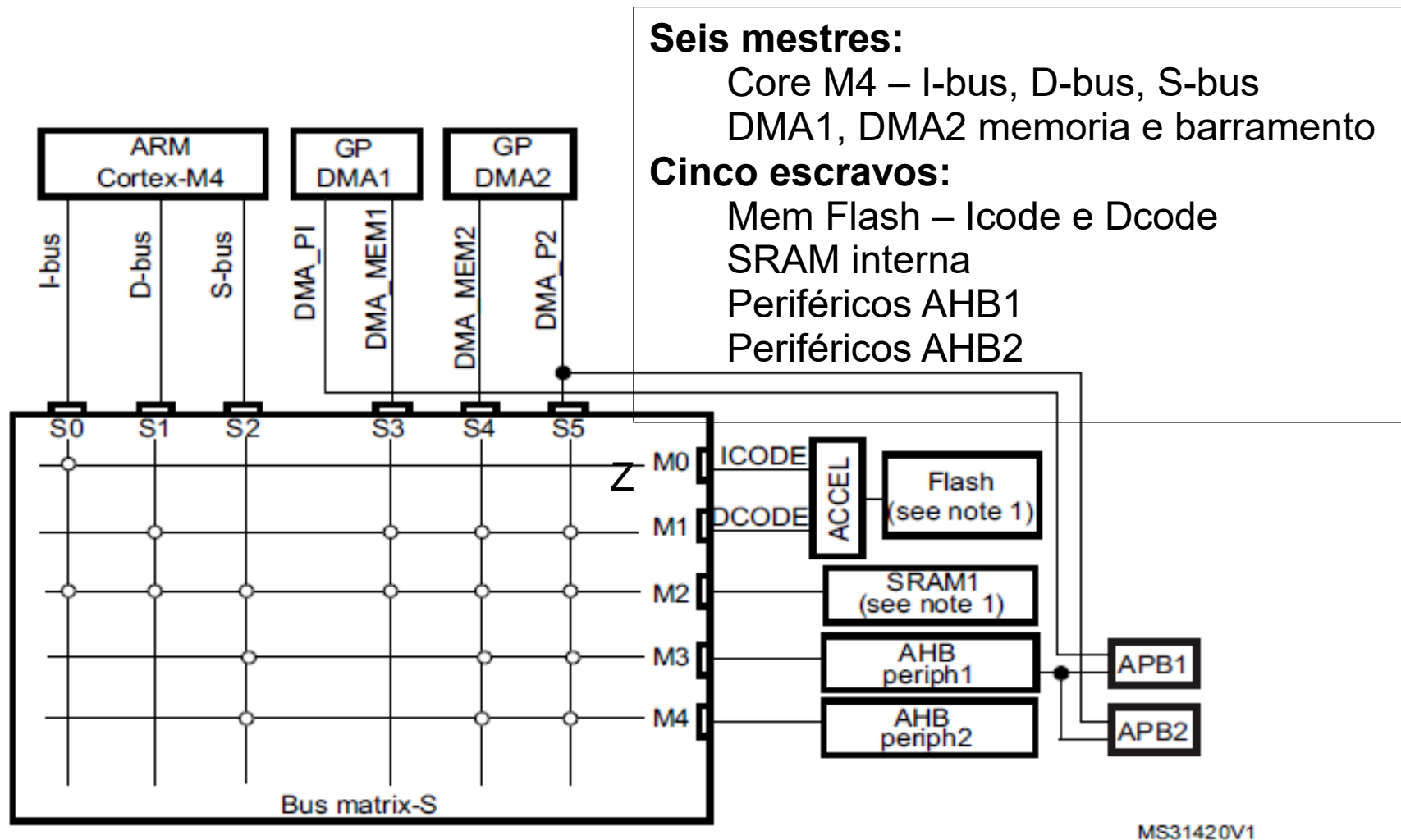
USART

SPI

SDIO -interface de audio

USB 2.0

Arquitetura do sistema



Periféricos Cortex M4

- SysTick
- Controlador de Interrupção Aninhado e Vetorado(NVIC)
- Bloco de Controle do Sistema (SCB)
- Unidade de Proteção de Memória (MPU)
- Unidade de Ponto Flutuante

SysTick

(Temporizador do sistema)

- SysTick fornece um contador decrementador de 24-bits, com um mecanismo de controle flexível.

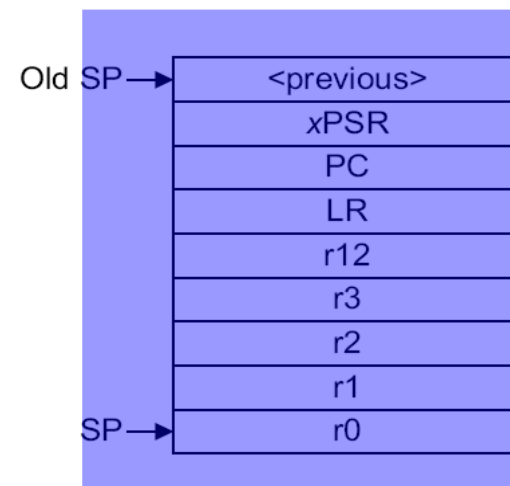
O contador pode ser usado de diferentes maneiras, como :

- Um temporizador de tick para RTOS tick com disparo a uma taxa programável (por exemplo, 100 Hz) e chama a rotina de SysTick.
- Um temporizador alarme de alta velocidade .
- Um alarme ou temporizador de sinalização de taxa variável.
- Um contador simples que pode ser utilizado pelo software para medir tempo de execução ou tempo utilizado em determinada tarefa.
- Uma fonte de clock interno.

O campo de bits COUNTFLAG nos registradores de controle e status podem ser utilizados para determinar se uma ação é completada dentro de um tempo pré-definido, como parte do laço de controle dinâmico do clock.

NVIC - Interrupções

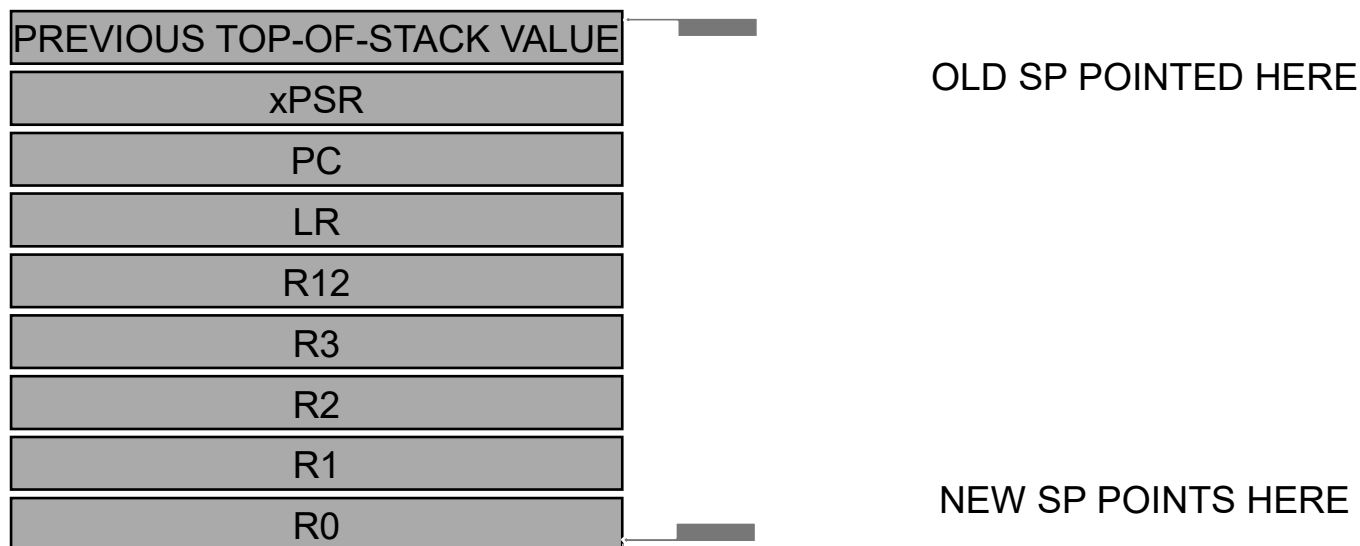
- Salvamento / restauração automática do contexto e entrada direta na tabela de vetores de funções.
- Nested Vectored Interrupt Controller (NVIC) prioriza e manipula todas as excessões
- Suporte à Tail-chaining
- Implementação de interrupções Preemptivas e Aninhadas
- Agrupamento de prioridades
- Possibilidade de relocação da tabela de vetores
- Processos só tem acesso completo ao NVIC em modo privilegiado
- 10 tipos de excessões do núcleo do Cortex M3 com prioridade programável, exceto Reset, NMI, Hard fault (maiores prioridades)
- Até 53 interrupções de periféricos(GPIOs, PWMs, ADCs etc.)



Modelo de Excessões

- Modelo de Excessões manipula todas as interrupções, falhas síncronas e excessões SVC
 - Excessões causam travamento na máquina de estados da cpu
- Manipulação de excessões é tão trivial quanto manipulação de registradores. É realizado em hardware.
 - Não é necessário código em assembler.
 - Rotinas de serviço de interrupção em 'C'

```
void IRQ(void) { /* my handler */ }
```



MPU

- Memory Protection Unit (MPU)
- Benefícios
 - Reforça as regras de privilégio
 - Reforça as regras de acesso
 - Separa os processos
- Características
 - 8 regiões de proteção (sem acesso, somente leitura, leitura escrita) de faixas a partir de 32 B a 4GB.
 - Permissões de acesso (privilegiada/usuário)
 - Regiões com sobreposição de proteção com prioridades por região.
 - Erros e violação de permissões da MPU invocam o manipulador de falhas “MemManage”

FPU

- Floating Point Unit
- A FPU suporta operações de precisão simples, como soma, subtração, multiplicação, divisão, multiplicação e acumulação e raiz quadrada.
- Provê conversão entre os formatos de ponto fixo e ponto flutuante, além de instruções com constantes do tipo ponto flutuante.
- A FPU fornece funcionalidades de computação em ponto flutuante compatíveis com as normas ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referidas como padrão IEEE 754.
- A FPU contém registradores de 32 bits, precisão simples, que podem ser acessados como 16 registradores tipo double-word, para operações de leitura, armazenagem e movimentação.
-