Esse é o sétimo artigo da série escrita pelo engenheiro Ismael Lopes da Silva, exclusivamente para o site "www.embarcados.com.br". Nessa série focarei no Microcontrolador da STMicroelectronics, o MCU STM32F103C8T6, que é um ARM Cortex-M3. Os pré-requisitos para uma boa compreensão dos artigos é ter o domínio da Linguagem C Embedded e conceitos de eletrônica.

Os Registradores do Núcleo (Core) do Processador ARM Cortex-M3

Os Registradores do core do processador ARM Cortex-M3, são de 32 bits, ilustrados a seguir:

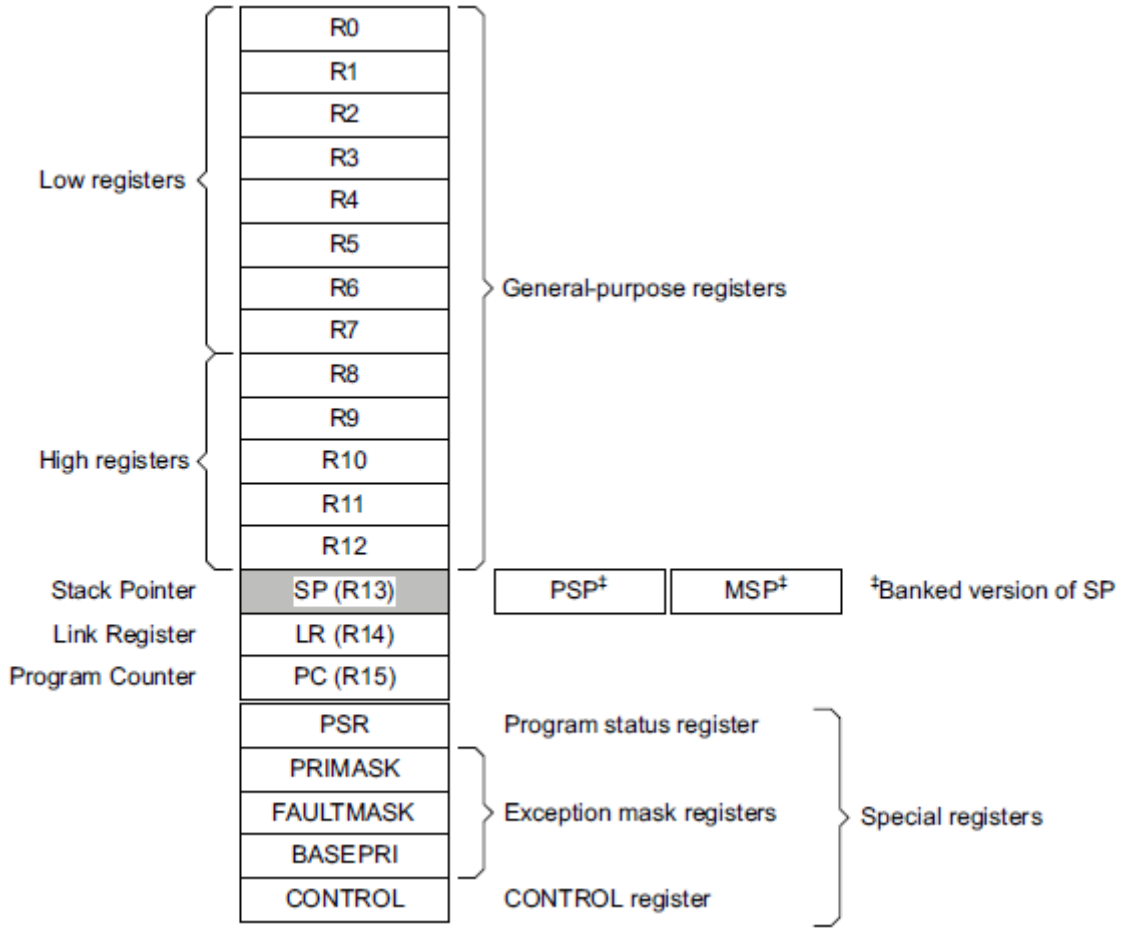

Figura 1 – O núcleo dos Registradores do processador ARM Cortex-M3

- Os Registradores de R0 à R12 são de Propósito Geral;
- O Registrador R13 é chamado de Stack Pointer (SP) ou Registrador Ponteiro de Pilha;
- Ao lado do Registrador SP, temos dois Registradores, que são PSP (Process Stack Pointer) e o MSP (Main Stack Pointer), chamado de versão Banked do Stack Pointer;
- O Registrador R14 é chamado Link Register (LR);
- O Registrador R15 é chamado de Program Counter (PC) ou Registrador Contador de Programa;
- Os cinco (5) Registradores especiais:
  - ✔ O Registrador Program Status Register (PSR) ou Registrador de Status de Programa;
  - ✔ Os três (3) Registradores de máscara das exceções:

- ➢ O Registrador PRIMASK;
- ➢ O Registrador FAULTMASK;
- ➢ O Registrador BASEPRI.

✔ O Registrador CONTROL.

Os Registradores de Proposito Geral

Os Registradores de R0 à R12 são de propósito geral para operação com dados.

O Registrador Ponteiro da Pilha - Stack Pointer (SP)

O Registrador Stack Pointer é o Registrador R13. Esse Registrador é usado para rastrear a seção stack da memória. Em operação modo Thread, o bit [1] do Registrador especial CONTROL indica o ponteiro da pilha a ser usado no Banker:

✔ 0 = Ponteiro da pilha principal (MSP – Main Stack Pointer). Este é o valor quando resetado;

✔ 1 = Ponteiro de pilha do processo (PSP - Process Stack Pointer).

No reset, o processador carrega o Registrador MSP com o valor contido no endereço 0x00000000.

O Registrador Link (LR)

O Registrador Link é o registro R14. Ele armazena o endereço de retorno das sub-rotinas, funções chamadas e exceções. No reset, o processador define o valor LR para 0xFFFFFFFF.

O Registrador Contador do Programa - Program Counter (PC)

O Registrador Program Counter é o Registrador R15. Ele contém o endereço da instrução atual do programa. No reset, o processador carrega o Program Counter com o valor do vetor de reset, que está no endereço 0x00000004.

O Registrador de Status do Programa – Program Status Register (PSR)

O Registrador de Status do Programa combina três Registradores, que são:

✔ Registrador de Status do Programa de Aplicação (APSR - Application Program Status Register);

✔ Registrador de Status do Programa de Interrupção (IPSR - Interrupt Program Status Register);

✔ Registrador de Status do Programa de Execução (EPSR - Execution Program Status Register).

Esses Registradores são campos de bits mutuamente exclusivos no Registrador PSR de 32 bits. As atribuições de bits são:

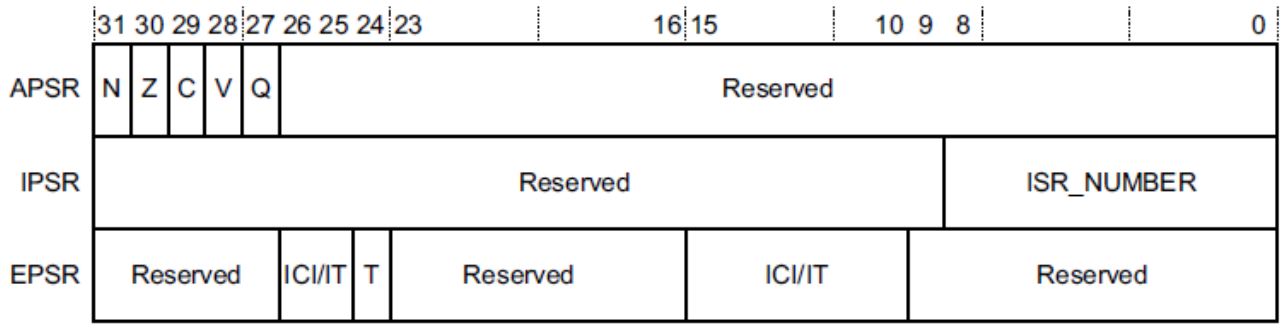

Figura 2 – Registradores mutuamente exclusivos APSR, IPSR e EPSR.

Acesse esses Registradores individualmente, ou como uma combinação de dois ou três Registradores, usando o nome do Registrador como um argumento para as instruções MSR ou MRS.

Registrador de Status do Programa de Aplicação (APSR)

O Registrador APSR contém o estado atual dos flags de condição da instrução que acabou de ser executada. As atribuições de bits são:

Bits	Nome	Função
[31]	N	Flag Negativo
[30]	Z	Flag de Zero
[29]	C	Flag de Carry ou Borrow
[28]	V	Flag de Overflow
[27]	Q	Flag Saturação
[26:0]	-	Reservados

Tabela 1 – Flags de condições do Registrador APSR

Registrador de Status do Programa de Interrupção (IPSR)

O Registrador IPSR contém o número do tipo da exceção da atual Rotina de Serviço de Interrupção (ISR). As atribuições de bits são:

Bits	Nome	Função
[31:9]	-	Reservados
[8:0]	ISR_NUMBER	O número da exceção atual

Tabela 2 – Bits do Registrador IPSR

Os ISR_NUMBER são:

- ✔ 0 = Modo Thread;
- ✔ 1 = Reservado;
- ✔ 2 = NMI;
- ✔ 3 = HardFault;
- ✔ 4 = MemManage;
- ✔ 5 = BusFault;
- ✔ 6 = UsageFault;
- ✔ 7-10 = Reservado;
- ✔ 11 = SVCall;
- ✔ 12 = Reservado para Depuração (Debugging);
- ✔ 13 = Reservado;
- ✔ 14 = PendSV;
- ✔ 15 = SysTick;
- ✔ 16 = IRQ0;
- ✔ n+15 = IRQ(n-1)[a].

([a]) número de interrupções, n, é definido pela implementação, e está na faixa de 1 à 240.

Registro de Status do Programa de Execução (EPSR)

O Registrador EPSR contém o estado do bit Thumb, e os bits de estado da execução. As atribuições de bits são:

Bits	Nome	Função
[31:27]	-	Reservado
[26:25], [15:10]	ICI/IT	Indica a posição interrompida de uma instrução contínua ou o estado de execução de uma instrução de TI
[24]	T	Bit da instrução Thumb
[23:16]	-	Reservado
[9:0]	-	Reservado

Tabela 3 – Bits do Registrador EPSR

Tenta ler o Registrador EPSR diretamente através do software de aplicação, usando a instrução MSR, sempre retorna zero. Tenta escrever no Registrador EPSR, usando a instrução MSR no software de aplicação, é ignorado.

- Instruções interrompíveis-continuáveis

Quando ocorre uma interrupção durante a execução de uma instrução LDM, STM, PUSH ou POP, o processador:

- ✔ Temporariamente para a operação da instrução de múltiplo carregamento ou armazenamento;
- ✔ Armazena o próximo operando do Registrador de uma operação múltipla, para os bits [15:12] do Registrador EPSR.

Após o serviço de interrupção, o processador:

- ✔ Retorna ao Registrador apontado pelos bits [15:12];
- ✔ Retoma a execução da instrução de carregamento ou armazenamento múltiplo.

Quando o Registrador EPSR mantém o estado de execução do ICI, os bits [26: 25,11: 10] são zerados.

- Bloco If-Then

O bloco If-Then contém até quatro instruções seguidas de uma instrução de IT. Cada instrução no bloco é condicional. As condições para as instruções são todas iguais, ou algumas podem ser o inverso das outras.

- Instruções Thumb

O processador Cortex-M3 suporta apenas a execução de instruções Thumb, portanto, bit [24] = T = 1. Tentar executar instruções quando o bit T é 0, resulta em uma falha.

Os Registradores de Máscara de Exceção

Os Registradores de máscara de exceção desabilitam o tratamento de exceções pelo processador. Desativar exceções em que podem afetar o tempo de tarefas críticas. Para acessar esses Registradores, use as instruções MSR e MRS, ou CPS para modificar o valor dos Registradores PRIMASK ou FAULTMASK.

Registrador de Máscara Prioritária - PRIMASK

O Registrador PRIMASK previne a ativação de todas as exceções com prioridade configurável.

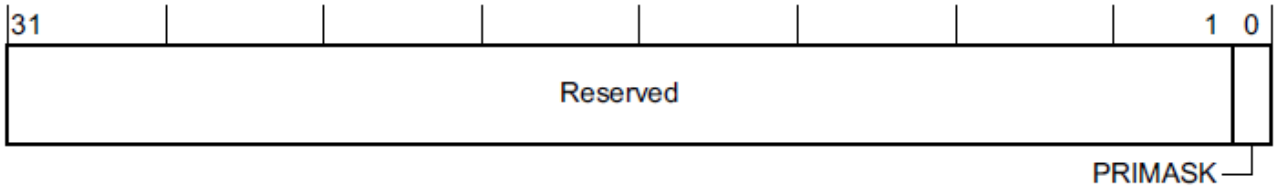

Figura 3 – Bits do Registrador PRIMASK

As atribuições de bits são:

Bits	Nome	Função
[31:1]	-	Reservado
[0]	PRIMASK	0 = sem efeito 1 = previne a ativação de todas as exceções com prioridade configurável

Tabela 4 – Bits do Registrador PRIMASK

Registrador de Máscara de Falha - FAULTMASK

O Registrador FAULTMASK previne a ativação de todas as exceções, exceto as Interrupções Não-Mascaráveis Interrupção (NMI – Non-Maskable Interrupt).

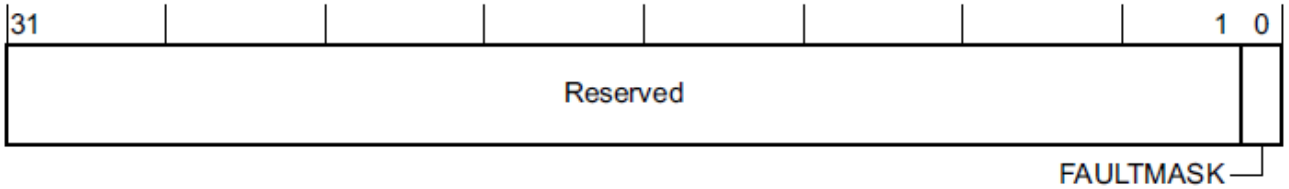

Figura 4 – Bits do Registrador FAULTMASK

As atribuições de bits são:

Bits	Nome	Função
[31:1]	-	Reservado
[0]	FAULTMASK	0 = sem efeito 1 = previne a ativação de todas as exceções com exceção as interrupções NMI

Tabela 5 – Bits do Registrador PRIMASK

Registro de Máscara com Prioridade Básica - BASEPRI

O Registrador BASEPRI define a prioridade mínima para o processamento de exceções. Quando o BASEPRI é definido como um valor diferente de zero, previne a ativação de todas as exceções com o mesmo valor ou menos nível de prioridade como o valor BASEPRI.

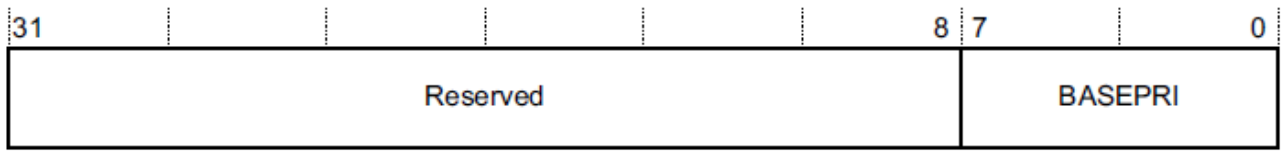

Figura 5 – Bits do Registrador BASEPRI

As atribuições de bits são:

Bits	Nome	Função
[31:8]	-	Reservado
[7:0]	BASEPRI	0 = sem efeito Não zero = define a prioridade base para o processamento de exceção. O processador não processa nenhuma exceção com um valor de prioridade maior ou igual a BASEPRI

Tabela 6 – Bits do Registrador BASEPRI

O Registrador CONTROL

O Registrador CONTROL controla a pilha usada e o nível de privilégio para a execução do software quando o processador estiver no modo Thread.

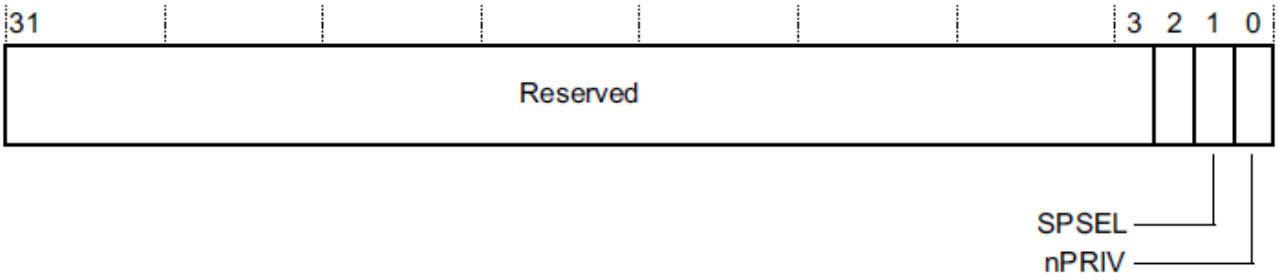

Figura 6 – Bits do Registrador CONTROL

As atribuições de bits são:

Bits	Nome	Função
[31:2]	-	Reservado
[1]	SPSEL	Define o Ponteiro de Pilha atualmente ativo: No modo Handler, este bit é lido como zero e ignora as escritas. O Cortex-M3 atualiza esse bit automaticamente no retorno de exceção. 0 = MSP é o Ponteiro de Pilha atual 1 = PSP é o Ponteiro de Pilha atual
[0]	nPRIV	Define o nível de privilégio em modo Thread. 0 = Privilegiado 1 = Sem privilégio

Tabela 7 – Bits do Registrador CONTROL

O modo Handler sempre usa o MSP, como Ponteiro de Pilha; portanto, o processador ignora escritas explícitas no bit do Registrador CONTROL, quando está no modo Handler. A entrada e o retorno de exceção atualizam automaticamente o Registrador CONTROL.