



EA871

Chaves

Interrupções

Wu Shin – Ting

DCA – FEEC - Unicamp
Segundo Semestre de 2020

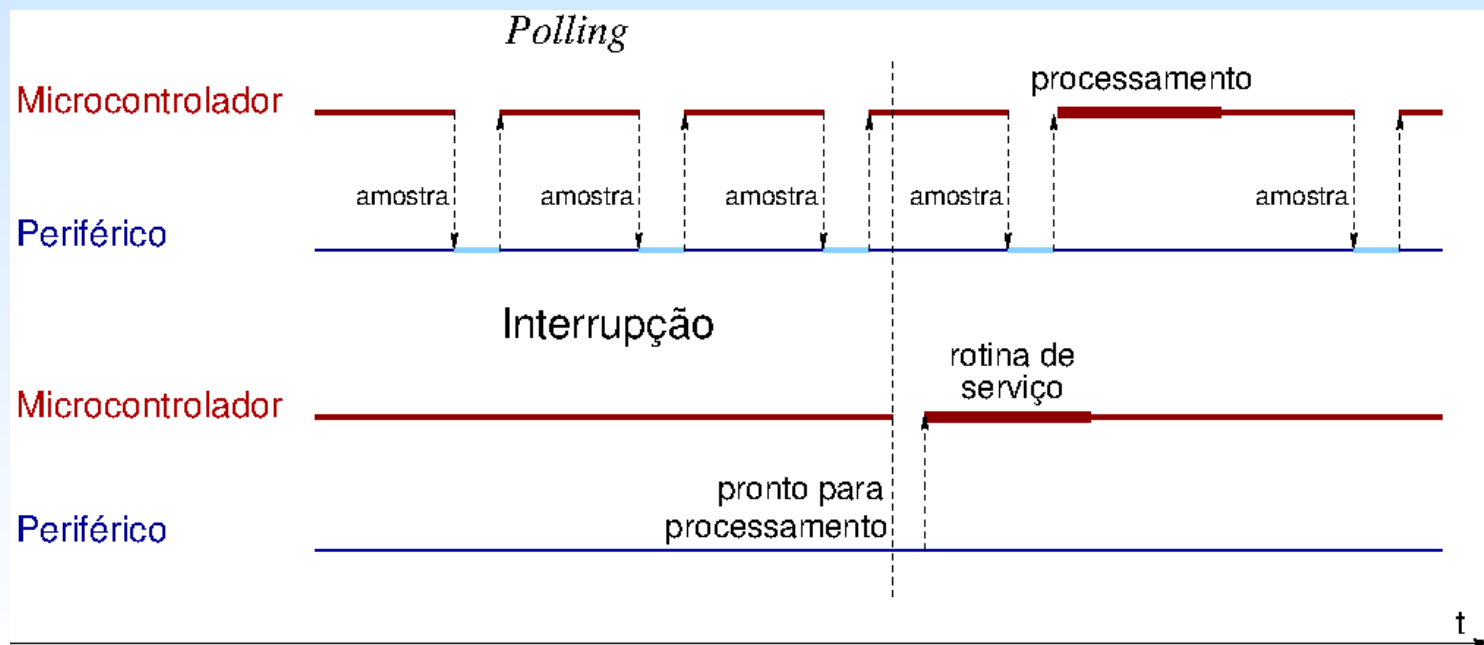
Técnicas

- **Polling**

- Processador amostra continua/periodicamente o estado do recurso para certificar se ele está pronto para executar uma operação.

- **Interrupção**

- Processador executa uma operação com uso de um recurso somente quando o recurso estiver pronto.



Polling vs. Interrupção

<i>Polling</i> (Amostragem)	Interrupção
Maior probabilidade na perda de dados.	Menor probabilidade na perda de dados.
Alocação de uma parte de tempo de processamento em tarefa de amostragem.	Uso mais eficiente do tempo do processador.
Maior consumo de energia.	Menor consumo de energia (modo de dormir).
Sequencialização do fluxo de controle.	Viável à paralelização dos fluxos de controle.
<i>Hardware</i> mais simples.	Necessário um <i>hardware</i> dedicado.
Estrutura de programa mais simples	Estrutura de programa mais complexa

Conceitos

- **Exceções (*Exceptions*):** eventos internos, gerados pelo próprio processador ao longo do fluxo de execução de um programa.
 - **Falha (*Fault*):** evento que ocorre antes da execução de uma instrução, como identificação de uma instrução não definida.
 - **Armadilha (*Trap*):** evento gerado na execução de uma instrução, como divisão por zero ou acesso a um endereço de memória inválido.
 - **Aborto (*Abort*):** evento de erro grave para o qual não se consegue determinar o momento em que ocorreu nem reiniciar o programa, como exceções não recuperáveis.
- **Interrupções (*Interrupts*):** eventos externos assíncronos.

Conceitos

- **Eventos/Requisições de interrupção:** eventos externos que disparam uma interrupção.
 - **Sensíveis a pulsos/bordas:** de subida e/ou de descida.
 - **Sensíveis a níveis lógicos:** 0 ou 1
- **Número de exceção:** é o número associado a uma requisição de interrupção através do qual consegue-se acessar o endereço inicial da sequência de instruções que processa a requisição identificada.
- **Interrupção vetorizada:** quando, junto com a requisição de interrupção, o periférico externo se auto-identifica por meio do seu número de exceção no ciclo de *interrupt acknowledgment*.
- **Interrupção não-vetorizada:** quando não há auto-identificação. O processador precisa varrer os registradores de estado dos periféricos para identificar a fonte de uma requisição.

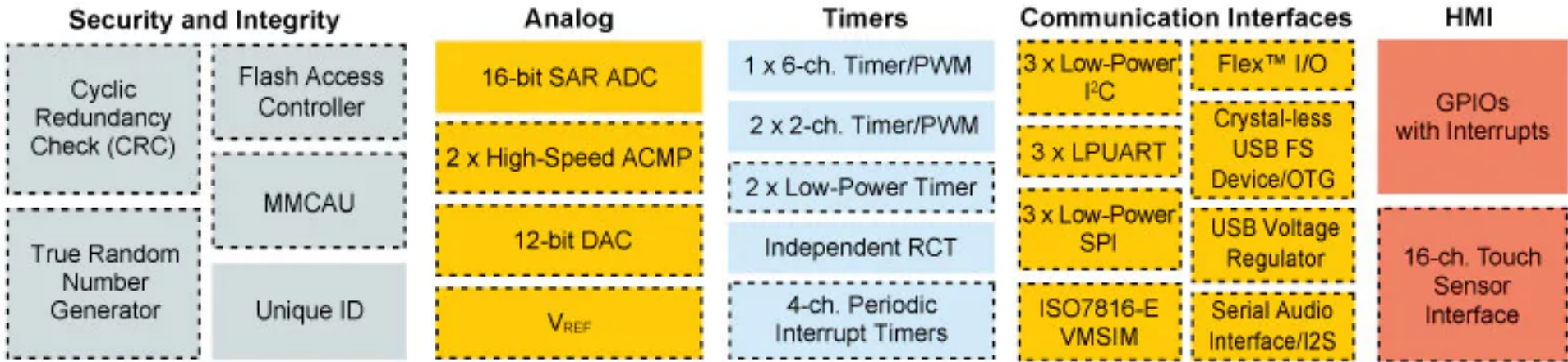
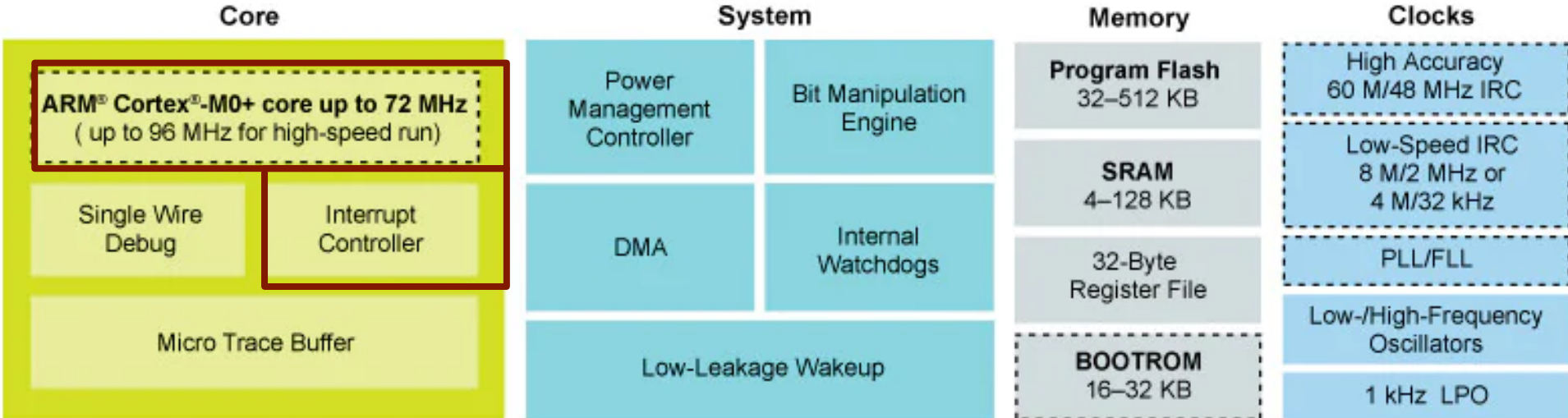
Conceitos

- **Rotina de serviço (*event handler*):** é uma sequência de instruções executada como resposta a uma requisição ativa de interrupção. É dependente do aplicativo.
- **Prioridade de Interrupção:** estabelece a ordem de atendimento a um conjunto de requisições de interrupção num dado instante de tempo.
- **Interrupções aninhadas:** são interrupções em que as de prioridade maior são capazes de interromper uma rotina de serviço em curso.
- **Interrupções não-aninhadas:** são interrupções que não conseguem interromper a execução de uma rotina de serviço, mesmo que elas tenham uma prioridade maior.

Conceitos

- **Estados de requisição de uma interrupção:**
 - **Interrupções habilitadas:** requisições de interrupção conseguem disparar o processamento de uma interrupção.
 - **Interrupções ativas:** interrupções em curso de processamento por um processador.
 - **Interrupções pendentes:** interrupções habilitadas aguardando o processamento.
- **Vetor de interrupções:** é uma tabela de endereços de rotinas de serviço para tratamento de interrupções. Tipicamente, acessíveis pelos números de exceção.

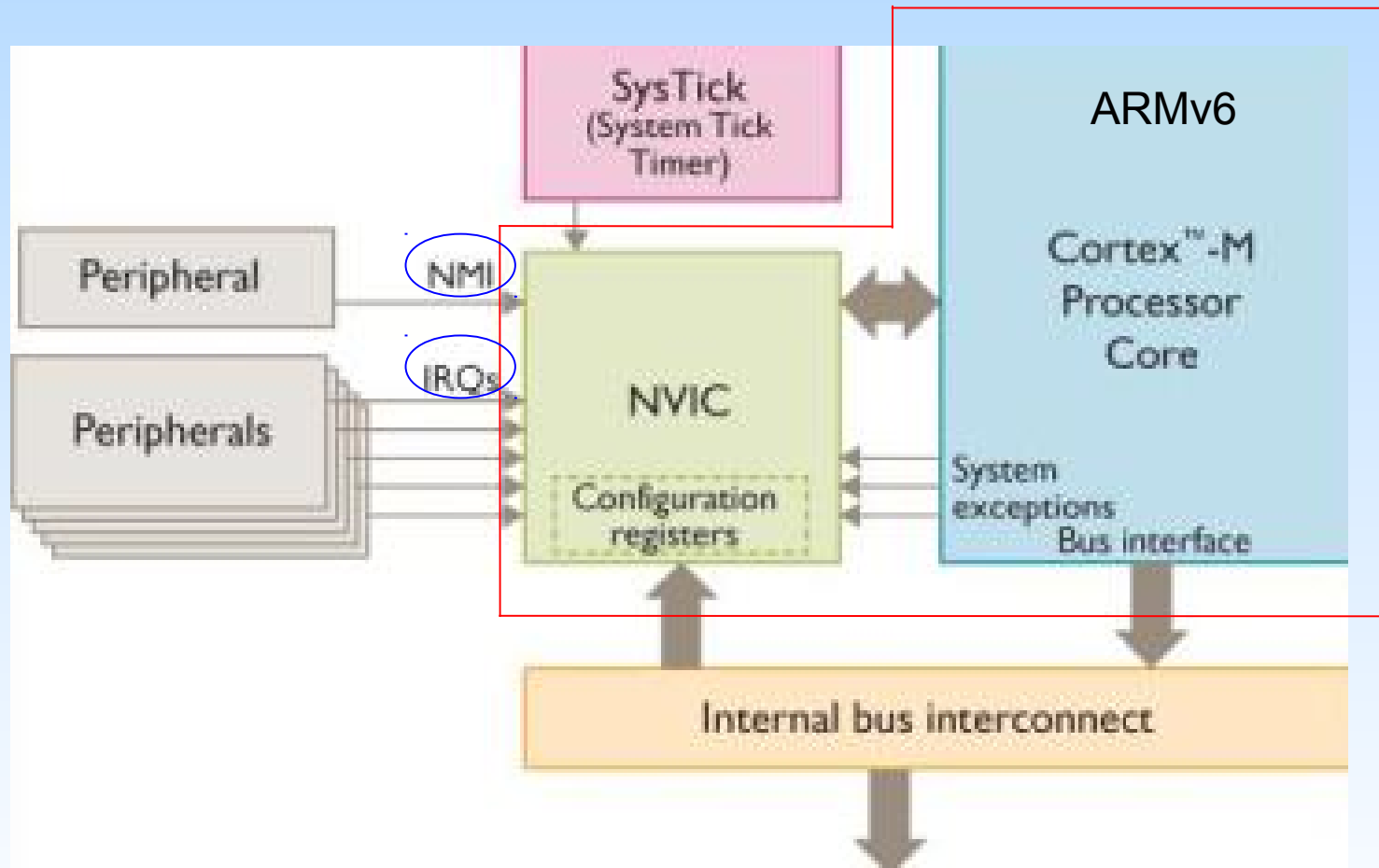
Microcontrolador Kinetis KL25Z



Optional

Controlador de Interrupção

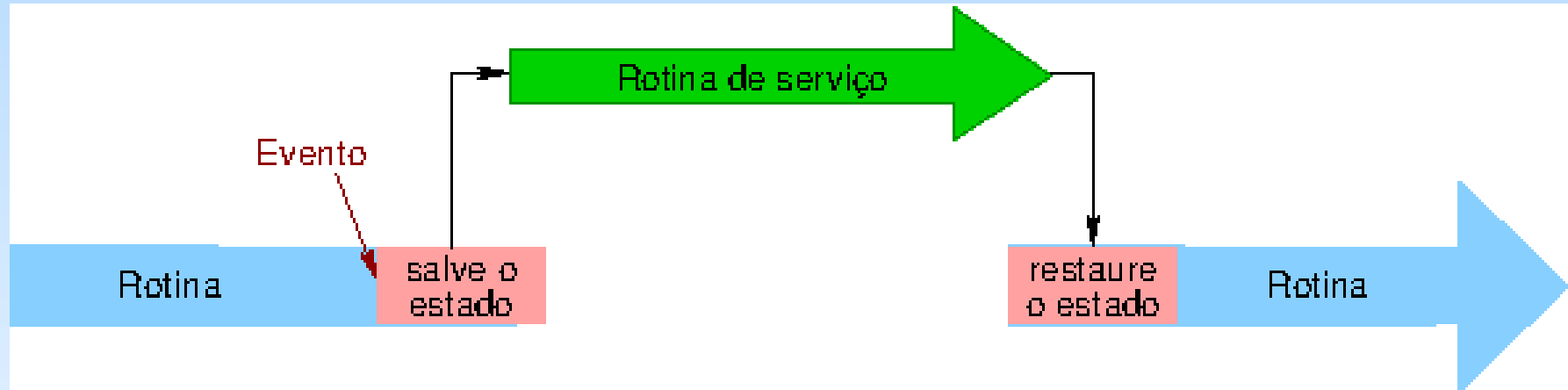
- **NVIC** (*Nested Vectored Interrupt Controller*): Controlador de interrupções aninhadas e vetorizadas



Microcontrolador – Chave Mecânica

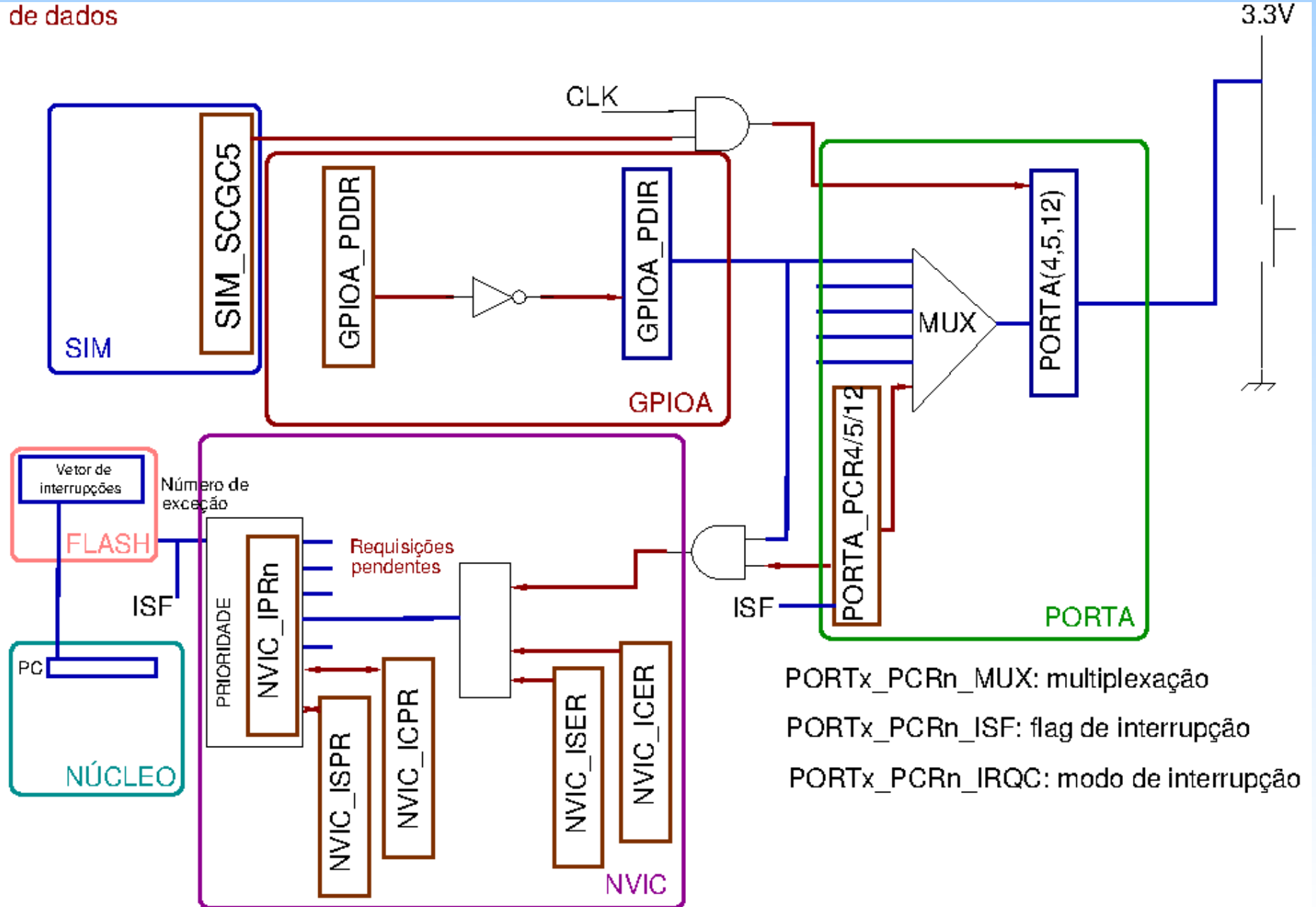
	Microcontrolador	Chave
Funcional	1/0	Aberta (estado normal)/Fechada
Elétrico	3V3/0V	Técnica de <i>debounce</i>: 3V3/0V
Temporal	1 ciclo de instrução ~ 50ns	Técnica de interrupção
Mecânico	Pino de saída e terra	Um pólo e uma posição.

Modificadores de Fluxo de Execução



Fluxo de Dados

Fluxo de dados



Número de Exceções e IRQ

Address	Vector	IRQ ¹	NVIC IPR register number ²	Source module	Source description
0x0000_0000	0	—	—	ARM core	Initial Stack Pointer
0x0000_0004	1	—	—	ARM core	Initial Program Counter
0x0000_0008	2	—	—	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	—	—	ARM core	Hard Fault
0x0000_0010	4	—	—	—	—
0x0000_0014	5	—	—	—	—
0x0000_0018	6	—	—	—	—
0x0000_001C	7	—	—	—	—
0x0000_0020	8	—	—	—	—
0x0000_0024	9	—	—	—	—
0x0000_0028	10	—	—	—	—
0x0000_002C	11	—	—	ARM core	Supervisor call (SVCall)
0x0000_0030	12	—	—	—	—
0x0000_0034	13	—	—	—	—
0x0000_0038	14	—	—	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	—	—	ARM core	System tick timer (SysTick)
Non-Core Vectors					
0x0000_0040	16	0	0	DMA	DMA channel 0 transfer complete and error

Número de Exceções e IRQ

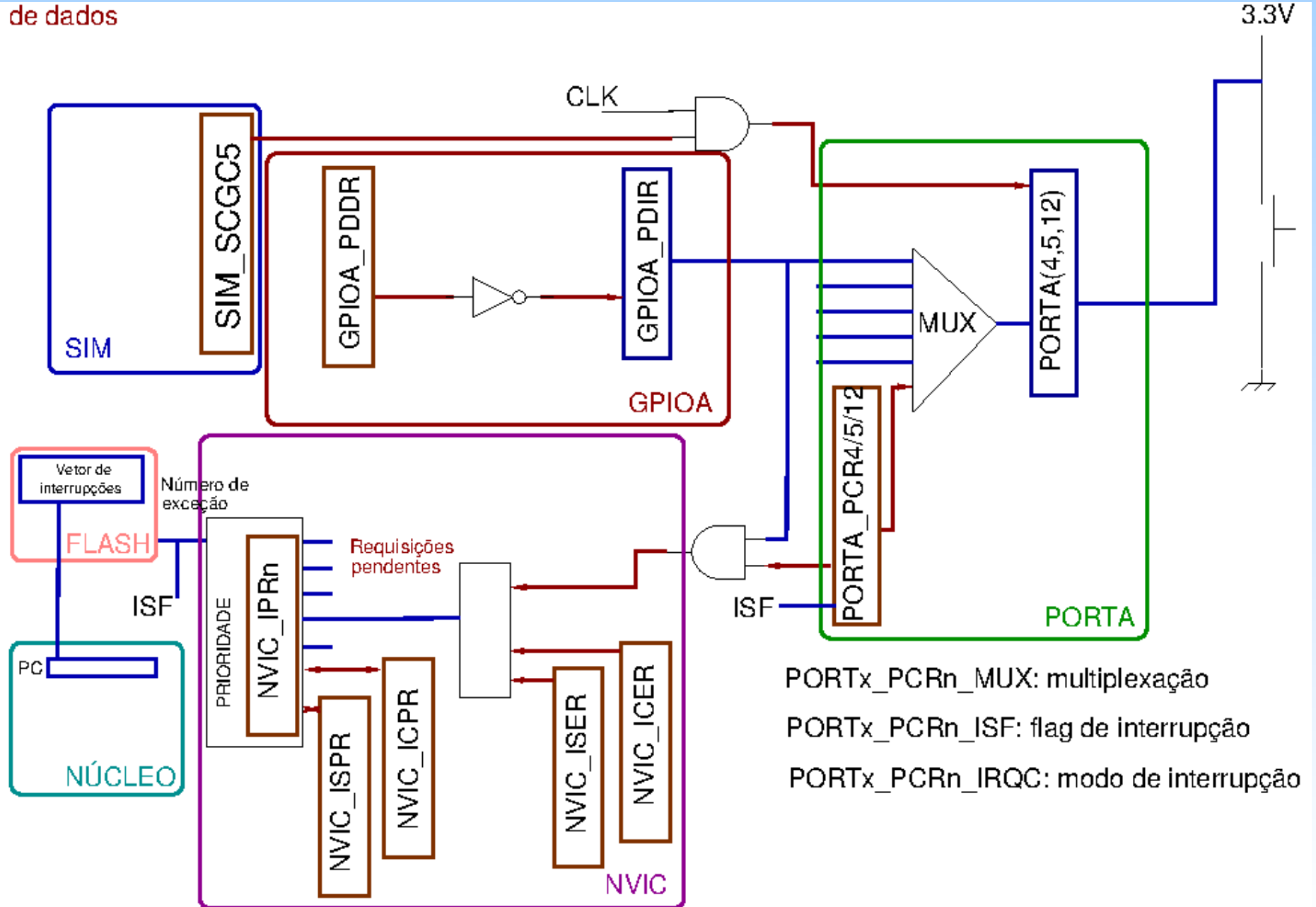
Address	Vector	IRQ ¹	NVIC IPR register number ²	Source module	Source description
0x0000_0090	36	20	5	RTC	Alarm interrupt
0x0000_0094	37	21	5	RTC	Seconds interrupt
0x0000_0098	38	22	5	PIT	Single interrupt vector for all channels
0x0000_009C	39	23	5	—	—
0x0000_00A0	40	24	6	USB OTG	
0x0000_00A4	41	25	6	DAC0	
0x0000_00A8	42	26	6	TSI0	
0x0000_00AC	43	27	6	MCG	
0x0000_00B0	44	28	7	LPTMR0	
0x0000_00B4	45	29	7	—	
0x0000_00B8	46	30	7	Port control module	Pin detect (Port A)
0x0000_00BC	47	31	7	Port control module	Pin detect (Port D)

1. Indicates the NVIC's interrupt source number.

2. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: $IRQ \div 4$

Fluxo de Dados

Fluxo de dados

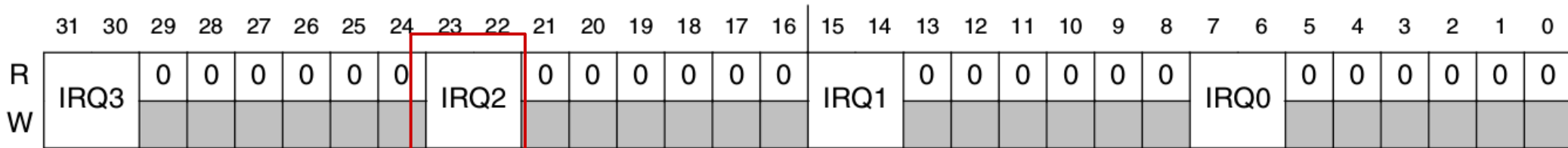


NVIC: Controle de IRQs

Table B3-18 NVIC register summary

Address	Name	Type	Reset	Description
0xE000E100	NVIC_ISER	RW	0x00000000	<i>Interrupt Set-Enable Register, NVIC_ISER on page B3-284</i>
0xE000E104- 0xE000E17F	-	-	-	Reserved
0xE000E180	NVIC_ICER	RW	0x00000000	<i>Interrupt Clear Enable Register, NVIC_ICER on page B3-285</i>
0xE000E184- 0xE000E1FF	-	-	-	Reserved
0xE000E200	NVIC_ISPR	RW	0x00000000	<i>Interrupt Set-Pending Register, NVIC_ISPR on page B3-286</i>
0xE000E204- 0xE000E27F	-	-	-	Reserved
0xE000E280	NVIC_ICPR	RW	0x00000000	<i>Interrupt Clear-Pending Register, NVIC_ICPR on page B3-287</i>
0xE000E300- 0xE000E3FC	-	-	-	Reserved
0xE000E400- 0xE000E41C	NVIC_IPR _n	RW	0x00000000	<i>Interrupt Priority Registers, NVIC_IPR0 - NVIC_IPR7 on page B3-288</i>

Configuração de Níveis de Prioridade NVIC_IPRn



aumenta ↑

-3	<i>Reset</i>	
-2	NMI	
-1	<i>HardFault</i>	
IRQ0-IRQ3		0
IRQ4-IRQ7		1
IRQ8-IRQ11		2
IRQ12-IRQ15		3
IRQ16-IRQ19		4
IRQ20-IRQ23		5
IRQ24-IRQ27		6
IRQ28-IRQ31		7

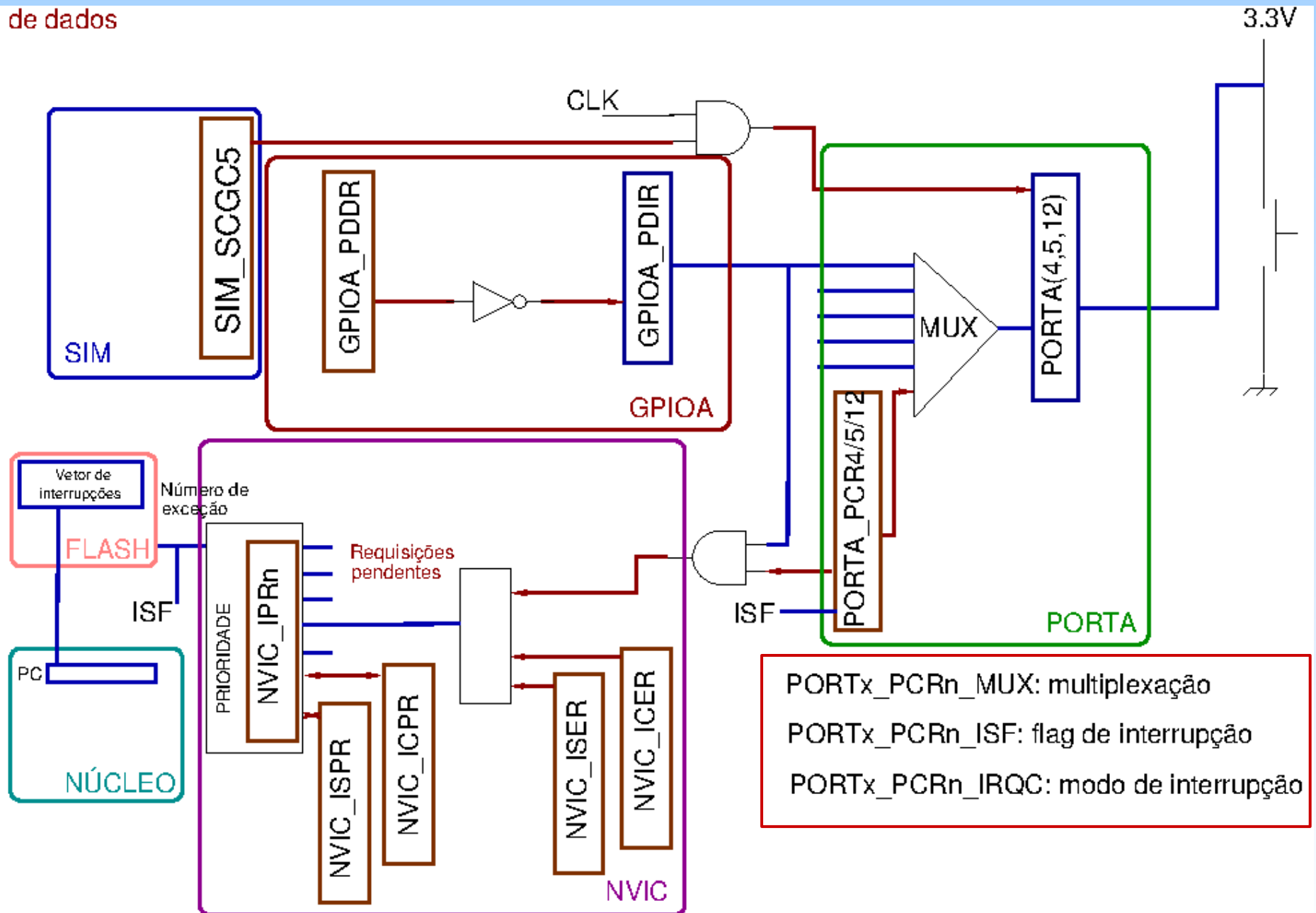
configuráveis

Só há **4 níveis de prioridade** dentro de cada grupo n
 $n = \text{IRQ}/4$

Para setar o nível de prioridade de IRQ dentro de um grupo, os 2 bits correspondentes são:
 $8 * (\text{IRQ} \bmod 4) + 6$
 e
 $8 * (\text{IRQ} \bmod 4) + 7$

Fluxo de Dados

Fluxo de dados



Configuração de PORTx_PCRn

Address: Base address + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0							ISF	0				IRQC				
W	█							w1c	█				█				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0						MUX			0	DSE	0	PFE	0	SRE	PE	PS
W	█						█			█	█	█	█	█	█	█	█
Reset	0	0	0	0	0	x*	x*	x*	0	x*	0	x*	0	x*	x*	x*	

* Notes:

- x = Undefined at reset.

- IRQC (*Interrupt Request Configuration*): configuração do modo.

0b0000: desabilita interrupção

0b0001: requisição DMA na borda subida

0b0010: requisição DMA na borda descida

0b0011: requisição DMA em ambas bordas

0b1000: interrupção no nível lógico 0

0b1001: interrupção na borda subida

0b1010: interrupção na borda descida

0b1011: interrupção em ambas as bordas

0b1100: interrupção no nível lógico 1

Outros reservados

Configuração de Portx_PCRn

Address: Base address + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0							ISF	0								
W	[Shaded]							w1c	[Shaded]								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0						MUX			0	DSE	0	PFE	0	SRE	PE	PS
W	[Shaded]						[Shaded]			[Shaded]	[Shaded]	[Shaded]	[Shaded]	[Shaded]	[Shaded]	[Shaded]	[Shaded]
Reset	0	0	0	0	0	x*	x*	x*	0	x*	0	x*	0	x*	x*	x*	

* Notes:

- x = Undefined at reset.

- ISF (*Interrupt Status Flag*): indica a presença de uma interrupção.
- Para algumas requisições, é necessário escrever 1 para limpar uma requisição (*w1c – write 1 to clear*) após o seu atendimento.

IRQ 30 - PORTA

- Qualquer pino da porta A pode ser fonte de interrupção IRQ 30 (ou número de exceção 46) do NVIC.
- Para certificar qual pino gerou uma requisição de interrupção, é necessário **consultar a flag de estado ISF de cada pino** programado para gerar requisições de interrupção.
- Por se tratar de um pino de sinal digital de propósito geral, **a flag de estado ISF deve ser limpa** (w1c) pelo desenvolvedor para que não sejam geradas novas requisições que já foram tratadas.

Rotinas de Serviço

- Respostas às requisições são dependentes do aplicativo.
- Técnica de programação em C para inserir os endereços das rotinas de serviço no Vetor de Interrupções (**kinetis_sysinit.c**).
 - Declaração das funções
 - Inserção dos nomes no vetor de interrupções

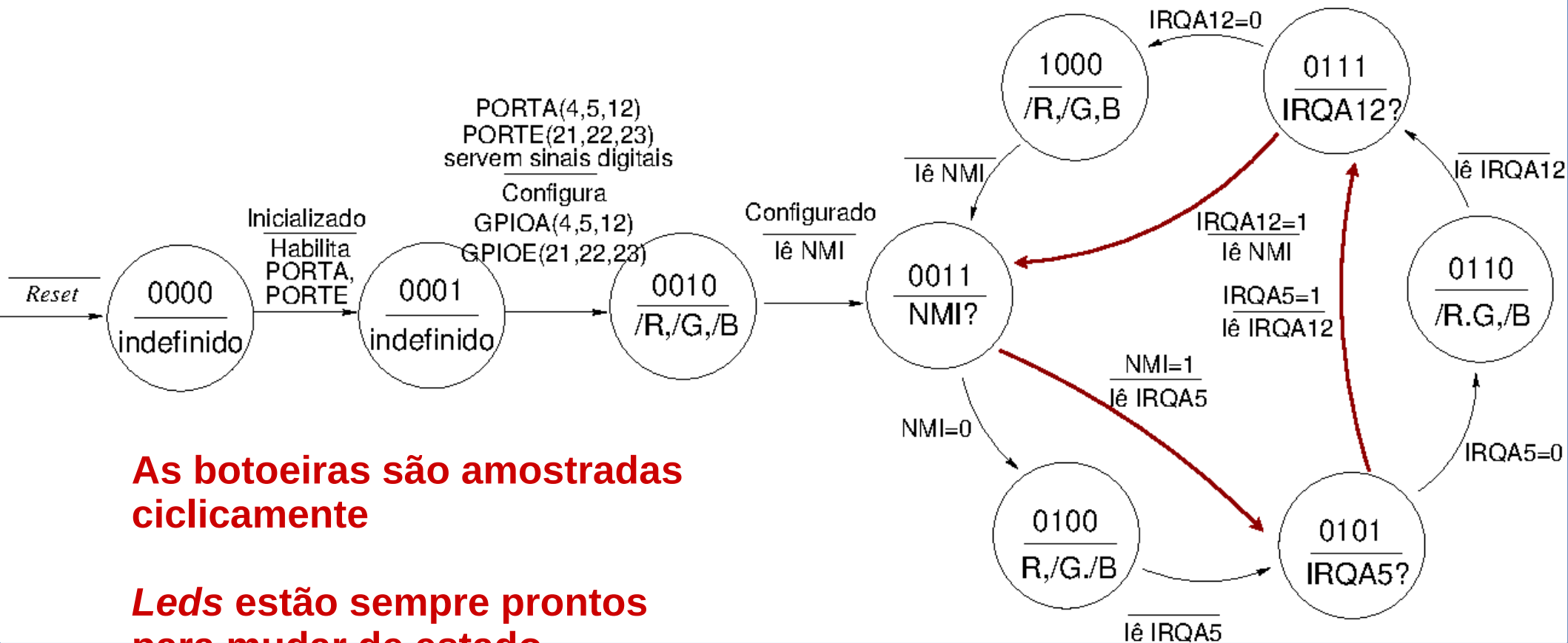
```
InterruptVector[] = {  
    &_estack,  
    __thumb_startup,  
    NMI_Handler,  
    :  
    PORTA_IRQHandler,  
    PORTD_IRQHandler  
};
```

Projeto-exemplo

- Acender os *leds* R, G e B acionando, respectivamente, as botoeiras NMI, IRQ5 e IRQ12.
 - Periféricos de saída (digitais): *leds* R, G e B
 - Nível lógico 1 (3.3V): aceso
 - Nível lógico 0 (0V): apagado
 - Periféricos de entrada (digitais): botoeiras remotas
 - Nível lógico 1 (3.3V) de controle : botão em estado normal.
 - Nível lógico 0 (0V) de controle: botão acionado

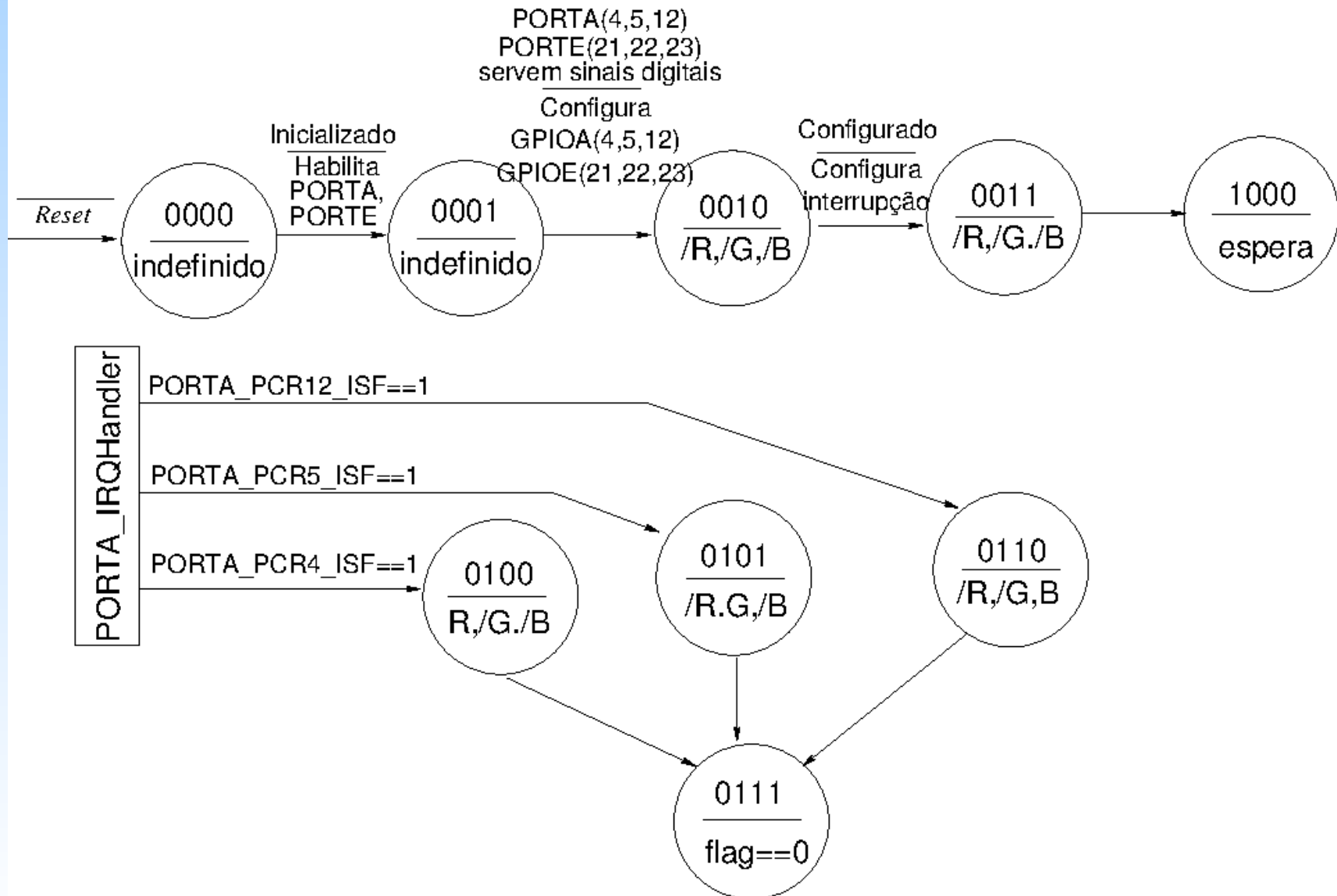
Solução por *Polling*

Máquina de Estados



Solução por Interrupção

Máquina de Estados



Configuração de Processamento de Interrupção

- **PORTA_PCR4_IRQC, PORTA_PCR5_IRQC, PORTA_PCR12_IRQC**
 - Interrupção na borda de descida: 0b1010
 - **NVIC_ISER**: *bit* 30, correspondente a IRQ30, deve estar habilitado (em 1).
 - **NVIC_ICPR**: escrever 1 no *bit* 30 para remover todas as pendências associadas a IRQ30.
 - Setar o nível de prioridade nos *bits*, se não quiser o valor *default* (0)
 - $8 \cdot (30 \bmod 4) + 6 = 22$
 - $8 \cdot (30 \bmod 4) + 7 = 23$
- no registrador **NVIC_IPR7**
- $n = 30/4 = 7$

Pseudocódigo

- `GPIO_initLedRGB()`; inicializa os registradores de controle dos *leds*
- `GPIO_initSwitches()`; inicializa os registradores de controle das chaves
- `GPIO_enableSwInterrupt()`; habilita para interrupções
- Laço de espera.

- Rotina de Serviço (**`PORTA_IRQHandler`** declarada em **`kinitis_sysinit.c`**):
 - Se (`PORTA_PCR4_ISF == 1`) então `GPIO_LedRGB(cor[VERMELHO]);`
Limpa `PORTA_PCR4_ISF`; retorna;
 - Se (`PORTA_PCR5_ISF == 1`) então `GPIO_LedRGB(cor[VERDE]);`
Limpa `PORTA_PCR5_ISF`; retorna;
 - Se (`PORTA_PCR12_ISF == 1`) então `GPIO_LedRGB(cor[AZUL]);`
Limpa `PORTA_PCR12_ISF`; retorna;



CodeWarrior IDE Development Suite

Informações Adicionais

- ARMv6-M Architecture Reference Manual

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/ARMv6-M.pdf>

- Modelo de Exceção em ARMv6: Seção B1.5
- NVIC: Seção B3.4

- KL25 Sub-Family Reference Manual

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf>

- NVIC: Capítulo 3 (página 51)
- PORT: Capítulo 11 (página 184)