



**ENGENHARIA ELETRÔNICA**

**EEN241 – MICROCONTROLADORES E SISTEMAS EMBARCADOS**

**4º NOTURNO**

**PROF. RAFAEL CORSI**

**HENRIQUE PEREIRA ROSA**

**11.02741-0**

**EDUARDO GALINSKAS KARWOSKI**

**13.01129-4**

**11/MAIO/2016**

# **Interrupção e Exceções**

## **Questão 1.1: NMI vs IRQ**

Uma interrupção não mascarável (NMI) pode ser sinalizada por um periférico ou desencadeada pelo software. Esta é a maior prioridade de exceção depois do reset. Ela é permanentemente ativada e tem uma prioridade fixa de -2 e não pode ser mascarada ou impedida de ativação por qualquer outra exceção. Já a IRQ é uma solicitação de interrupção, sinalizada por um periférico. Todas as interrupções são assíncronas para a execução da instrução.

## **Questão 2.1: IRQ vs ISR**

A diferença entre IRQ e ISR é que IRQ é um tipo de interrupção enquanto ISR é uma das maneiras de como o processador manipula exceções. ISR significa Rotinas de Serviço de Interrupção onde as interrupções IRQ0 até IRQ34 são manipuladas pelo ISR.

## **Questão 2.2: SAM4S**

O SAM4S suporta 35 interrupções e tem 15 níveis de prioridade onde o décimo quinto é o que tem menor prioridade.

## **Questão 2.3: FIQ**

Os pedidos de interrupção rápida (FIQs) são um tipo especializado de pedido de interrupção, uma técnica padrão usada em CPUs de computadores para lidar com eventos que precisam ser processados à medida que ocorrem, tal como receber dados a partir de uma placa de rede, ou ações do teclado ou do mouse. FIQs são específicas para a arquitetura ARM CPU, que suporta dois tipos de interrupções; FIQs para um manuseamento rápido, baixa latência interrupções e pedidos de interrupção (IRQs), para interrupções mais gerais.

## **Questão 2.4: IRQ vs FIQ**

Em um sistema ARM o FIQ sempre tem prioridade com relação ao IRQ.

## **Questão 2.5: SAM4S número da interrupção dos periféricos**

Os IDs são:

**PIOA = 11**

**PIOB = 12**

**TC0 = 23**

## **Questão 2.6: Limpando interrupção**

Caso a interrupção não seja retirada ela nunca vai parar de ocorrer, pois o Microcontrolador sempre vai “olhar” para ela.

## Questão 2.7: Latência de interrupção

Latência de interrupção corresponde ao intervalo de tempo que decorre entre o pedido de interrupção e o início do seu atendimento. A latência depende de vários fatores como arquitetura de projeto, prioridade de interrupções, forma interna de manipulação dos registradores e sequências de código que não admitem interrupção, portanto nesse período o uC pode estar verificando a prioridade da interrupção, como pode estar tratando outra interrupção de maior prioridade, etc.

## Questão 2.8: Latência Cortex M4

A latência de interrupção é de 12 ciclos.

## Questão 4.1: ASF – PIO

\*\*\*\*\*Latência de interrupção corresponde ao intervalo de tempo que decorre entre o pedido de interrupção

## Questão 4.2: ASF – Timer Counter (TC)

**tc\_init** – configura o timer counter para temporizador, gerador de waveform ou captura.

**tc\_start** – inicializa o clock do timer counter em canal especificado.

**tc\_enable\_interrupt** – habilita as interrupções do timer counter em um canal especificado.

## Questão 5.1: PIO – Interrupção Botão

Falling Edge detection ou Low-level detection.

## Questão 5.2: PIO – Interrupção

Basicamente, quando há uma detecção no pino, o pino correspondente do ISR register (PIO\_ISR) é setado. Estando o pino correspondente do PIO\_IMR também setado e então a interrupção é enviada ao NVIC para tratamento.

## Questão 5.3: Registradores Interrupção

**PIO\_IER/PIO\_IDR:** Habilitam interrupção por detecção de mudança de estado.

**PIO\_AIMER/PIO\_AIMDR:** Habilitam opções de detecção na interrupção por detecção de mudança de estado, como detecção por borda de sinal low ou high.

**PIO\_ELSR:** Selecionam se a detecção é por borda ou por nível de sinal.

**PIO\_FRLHSR:** Seleciona o nível do sinal que será detectado, ou seja, se a borda será de subida ou descida ou, em caso de detecção por nível, se será sinal high ou low.