

# Atividades aula 11

## Interrupção e exceções

Prof. Rafael Corsi Ferrão  
12 de maio de 2016

### 1 Exceções

#### 1.1 Qual a diferença entre as exceções NMI e IRQ?

*NMI é uma interrupção que não pode ser impedida, que já é nativa do hardware. Já o IRQ é uma interrupção que pode ser ignorada, já que possui um bit de máscara de interrupção que pode ser configurado.*

### 2 Interrupção

#### 2.1 Qual a diferença entre as exceções IRQ e ISR?

*IRQ é a interrupção em si, disparada automaticamente pelo hardware – Interrupt Request; ISR, por sua vez, é o código que será rodado durante a interrupção do programa – Interrupt Service Request.*

#### 2.2 No ARM que utilizamos no curso, quantas são as interrupções suportadas e qual a sua menor prioridade?

*O ARM M4 suporta até 240 interrupções, sendo que cada uma tem 256 níveis de prioridade.*

#### 2.3 Descreva o uso do FIQ.

*O FIQ (fast interrupt request) é utilizado de maneira prioritária em relação ao IRQ, como o próprio nome diz. Esta interrupção é utilizada quando se deseja prioridade em alguma operação.*

2.4 No diagrama anterior, quem possui maior prioridade IRQ ou FIQ?

*De acordo com o diagrama, o FIQ possui maior prioridade do que o IRQ.*

2.5 No datasheet, secção 11.1 informa o ID do periférico que está associado com a sua interrupção. Busque a informação e liste o ID dos seguintes periféricos:

- PIOA -> ID 11
- PIOB -> ID 12
- TC0 -> ID 23

2.6 O que aconteceria caso não limpemos a interrupção?

*Caso a interrupção não fosse limpa, ela seria executada novamente a cada ciclo do programa e não apenas no momento desejado, já que a interrupção é chamada automaticamente, rodando o código o qual está escrito nela*

2.7 O que é latência na resolução de uma interrupção, o que é feito nesse tempo? (Interrupt latency).

*A latência de uma interrupção é o tempo que o hardware leva para reconhecer a interrupção, ou seja, o tempo entre perceber que o flag está setado e iniciar a rotina da interrupção.*

2.8 De quantos ciclos é a latência do ARM Cortex M4?

## 3 Software – CMSIS

## 4 Software – ASF

4.1 Via documentação disponível no ASF, verifique as funções disponíveis para controlar o PIO. Qual a semelhança com as funções desenvolvidas em sala?

*A semelhança são os parâmetros passados às funções, tais como o ponteiro do PIO a ser controlado, a máscara, entre outros que podem ser observados através do link [http://asf.atmel.com/docs/latest/sam4s/html/group\\_sam\\_drivers\\_pio\\_group.html](http://asf.atmel.com/docs/latest/sam4s/html/group_sam_drivers_pio_group.html).*

4.2 Via documentação disponível no ASF, descreva o uso das seguintes funções do Timer Counter.

- `tc_init()`

*Esta função inicializa o módulo do Timer Counter, de modo que o default tem as seguintes características: contagem crescente com reset automático, uso do fPBA/8 como fonte de clock e habilitação do RC como comparador.*

- `tc_start()`

*Esta função inicia um timer ou um contador do módulo Timer Counter.*

- `tc_enable_interrupt()`

*Esta função habilita a interrupção do módulo Timer Counter.*

## 5 PIO - Interrupção

5.1 Qual deve ser a configuração para operarmos com interrupção no botão do kit SAM4S-EK2?

*Deve-se configurar o registrador `PIO_IMR` a partir do registrador `PIO_IER`, que configura o anterior para habilitar a interrupção, e então configurar em qual evento ocorrerá a interrupção a partir do registrador `PIO_ELSR`, que habilitará um dos quatro modos de interrupção: rising edge detector, falling edge detector, high level detector e low level detector.*

5.2 Com base no texto anterior e nos diagramas de blocos descreva o uso da interrupção e suas opções.

*O PIO está programado de forma que gera uma interrupção a partir da detecção de uma borda de subida ou descida ou de um nível lógico. É possível utilizar estas interrupções, inclusive, de forma simultânea, em diferentes eventos que ocorrem no botão, de acordo com os valores dos registradores responsáveis por habilitar/desabilitar cada um deles.*

5.3 Descreva as funções dos registradores:

- `PIO_IER` / `PIO_IDR`

*São os registradores que habilita e desabilita, respectivamente, a interrupção.*

- `PIO_AIMER` / `PIO_AIMDR`

*São os registradores que habilita e desabilita o modo de interrupções adicionais.*

- `PIO_ELSR`

*É o registrador que definirá em qual evento no botão irá ocorrer a interrupção: nível lógico alto ou nível lógico baixo.*

- PIO\_FRLHSR

*É o registrador que definirá em qual evento no botão irá ocorrer a interrupção: borda de descida ou borda de subida (rising edge or high level).*