Aluno: Carlos Eduardo Puttini Rosa

Professor: Rafael Corsi

Disciplina: Computação Embarcada

Data: 02/04/17

Pesquisa: Interrupções

**Questão 2.1: NMI vs. IRQ**

NMI (NON-Maskable Interrupt) é a interrupção de maior prioridade. Este tipo de interrupção pode interromper todo o software e dispositivos de hardware não vitais. O uso do NMI não é comum, mas para situações em que um grave erro ocorreu ou quando todas as operações do sistema param devido a uma falha. Um exemplo de uso do NMI é o ctrl+Alt+Del para interromper o computador quando ele para de responder aos nossos comandos.

IRQ é um pedido de interrupção, do inglês interrupt request (IRQ). Ao receber um IRQ do hardware, o processador interrompe o programa em execução, executa um outro programa de um manipulador de interrupções, e depois volta a execução do programa que estava em execução.

**Questão 3.1: IRQ vs. ISR**

O ISR é um manipulador de interrupção, é um handler. Ele interrompe o processo em execução, trata a execução, envia para a CPU e quando o ISR for concluído, retoma o processo anterior a interrupção. Um exemplo de uso do ISR é quando pressionamos uma tecla em um editor de texto: o ISR processa a entrada, sinaliza para a CPU, envia as informações para o editor de texto, e volta a executar o programa anterior a interrupção.

**Questão 3.2: SAME70**

**Questão 3.3: FIQ**

O FIQ é um pedido de interrupção com prioridade elevada. O FIQ desativa todos os manipuladores IRQ e FIQ durante a sua execução, de forma que nenhuma outra interrupção ocorre durante o período ativo da interrupção FIQ.

**Questão 3.4: IRQ vs. FIQ**

No diagrama apresentado, o FIQ tem prioridade em relação ao IRQ.

**Questão 3.5: SAME70 identificador (ID) da interrupção dos periféricos**

|  |  |
| --- | --- |
| Periférico | ID |
| PIOA | 10 |
| PIOC | 12 |
| TC0 | 23 |

**Questão 3.6: Limpando interrupção**

Caso não limpemos a interrupção através do PIO\_ISR, todas as interrupções pendentes não seriam tratadas através do manipulador.

**Questão 3.7: Latência da interrupção**

É o tempo que uma interrupção leva entre a sua geração até o seu tratamento.

**Questão 5.1: Interrupção Botão**

Para operarmos com interrupção no botão do kit SAME70-EK2 devemos:

1. Ativar o clock do periférico;
2. Configurar a interrupção de acordo com a borda escolhida;
3. Chamar o handler ou manipulador da interrupção no CORE;
4. Habilitar a interrupção do periférico que controla o botão;
5. Configurar a prioridade da interrupção.

**Questão 5.2: PIO – Interrupção**

Podemos programar o PIO para gerar interrupções na detecção de borda ou de nível em uma linha I/O, através do uso dos registradores PIO\_IER (Interrupt Enable Register) ou PIO\_IDR (Interrupt Disable Register) que colocam ou limpam o PIO\_IMR (Interrupt Mask Register). Dado que a detecção ocorre entre a comparação entre duas amostras na linha I/O, o periférico de clock também deve ser ativado.

Alguns modos de interrupção adicionais são realizados através dos registradores PIO\_AIMER (Additional Interrupt Modes Enable Register) ou PIO\_AIMDR (Additional Interrupt Modes Disable Register) e PIO\_AIMMR (Additional Interrupt Modes Mask Register). Estes modos adicionais são: detecção por borda de subida ou de descida, e detecção por nível baixo ou nível alto. No entanto, o tipo de detecção deve ser selecionado PIO\_ESR (Edge Select Register) ou PIO\_LSR (Level Select Register).

A polaridade da detecção deve ser escrita em PIO\_FELLSR (Falling Edge/Low-Level Select Register) e PIO\_REHLSR (Risisng Edge/ High-Level Select Register).

Um único sinal de interrupção é gerado para o controller e engloba os 32 canais do PIO e quando o programa lê PIO\_ISR, todas as interrupções são limpas automaticamente.

**Questão 5.3: Registradores**

PIO\_IER: ativa o PIO de interrupção;

PIO\_IDR: desativa o PIO de interrupção;

PIO\_AIMER: ativa o PIO para o modo de interrupção adicional;

PIO\_AIMDR: desativa o PIO para o modo de interrupção adicional;

PIO\_ELSR: mostra o status de borda ou de nível do PIO.