

# Laboratório de Introdução à Arquitetura de Computadores

IST - Taguspark

2017/2018

## Interrupções

### Guião 7

13 a 17 de novembro 2017

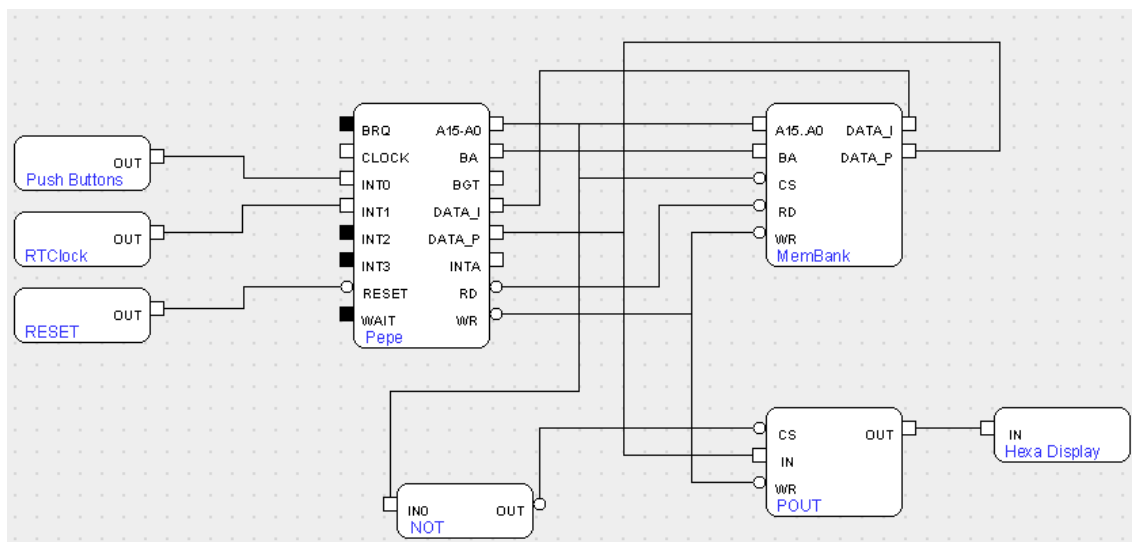
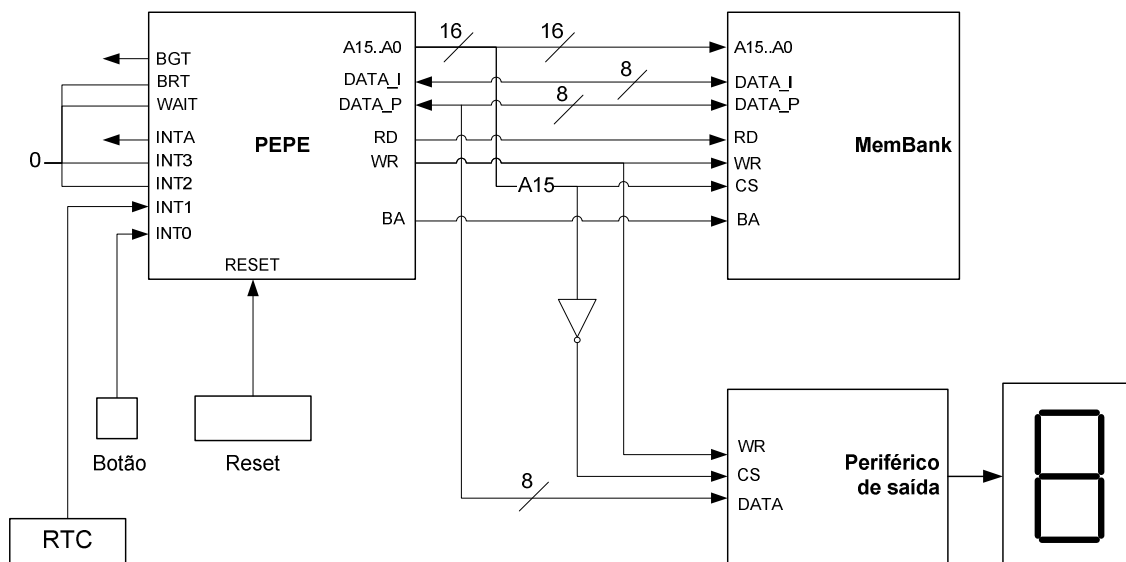
(Semana 9)

#### 1 – Objetivos

Com este trabalho pretende-se que os alunos pratiquem a utilização de rotinas de interrupção.


#### 2 – O circuito de simulação

Use o seguinte circuito, contido no ficheiro **lab7a.cmod**.



### 3 – Execução do trabalho de laboratório

#### 3.1 – Interrupção sensível ao flanco

Compile e carregue (  ) o programa *assembly lab7a.asm*.

Verifique a sua funcionalidade (o mostrador de 7 segmentos é incrementado sempre que se carrega no botão, através da interrupção 0).

Analise o ficheiro **lab7a.asm** e verifique a sua estrutura face ao funcionamento observado. Note a inicialização do SP, pilha, BTE, tabela de exceções e bits de permissão das interrupções.

Altere o programa para passar a reagir ao RTC (relógio de tempo real), diminuindo o valor do mostrador de 2 unidades sempre que a saída do RTC transitar de 0 para 1. Agora com o botão pode fazer subir o valor do mostrador, que periodicamente vai diminuindo (devido ao relógio ligado à interrupção 1).

Verifique que sabe responder às seguintes perguntas e justificar a resposta:

- O que é que sucede se se carregar no botão quando a rotina de interrupção que trata da interrupção 1 estiver a executar?
- E se o sinal de saída do RTC mudar de 0 para 1 a meio da execução da rotina que trata da interrupção 0?
- Em algum dos casos é possível o sistema baralhar-se, por meio de interferência de uma interrupção na outra?

#### 3.2 – Interrupção sensível ao nível

Altere agora a programação da interrupção 0, tornando-a sensível ao nível 1 e não ao flanco (veja como fazer na tabela A.4 do livro ou na secção 3.4 do manual do PEPE). Qual a alteração de comportamento do mostrador face ao botão?

#### 3.3 – Interrupções e programação cooperativa

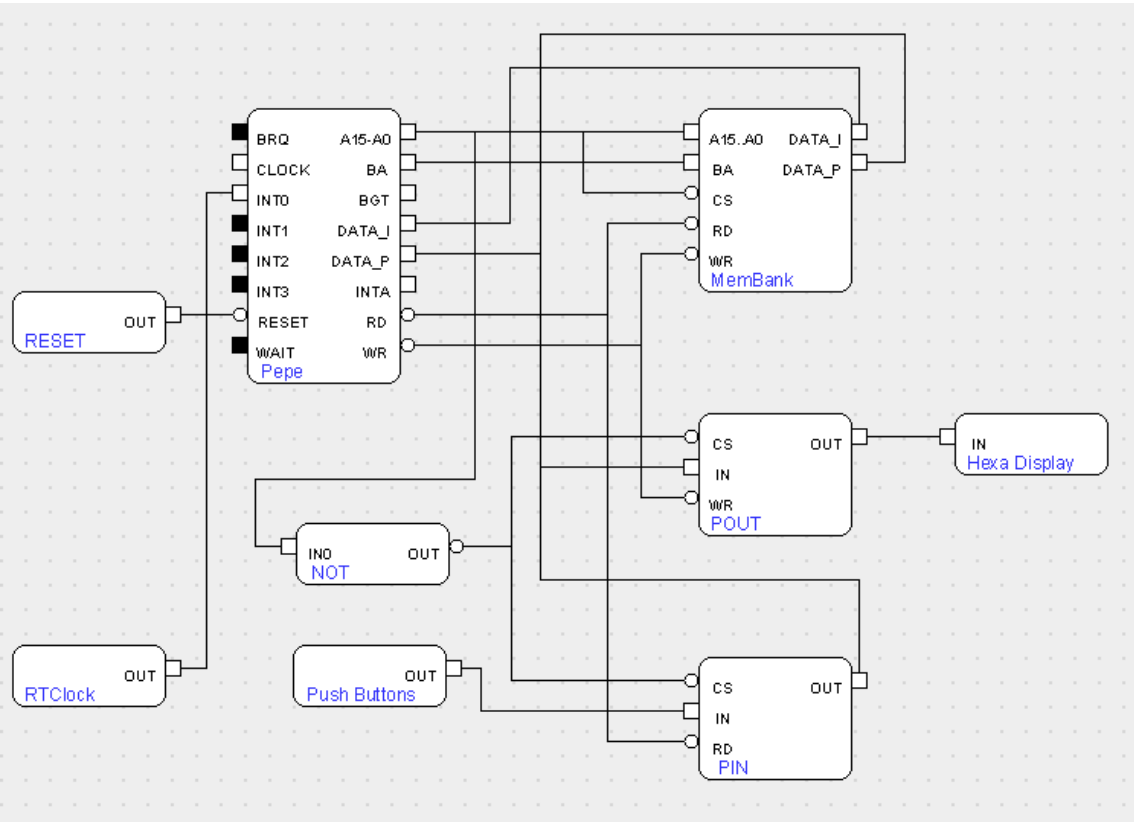
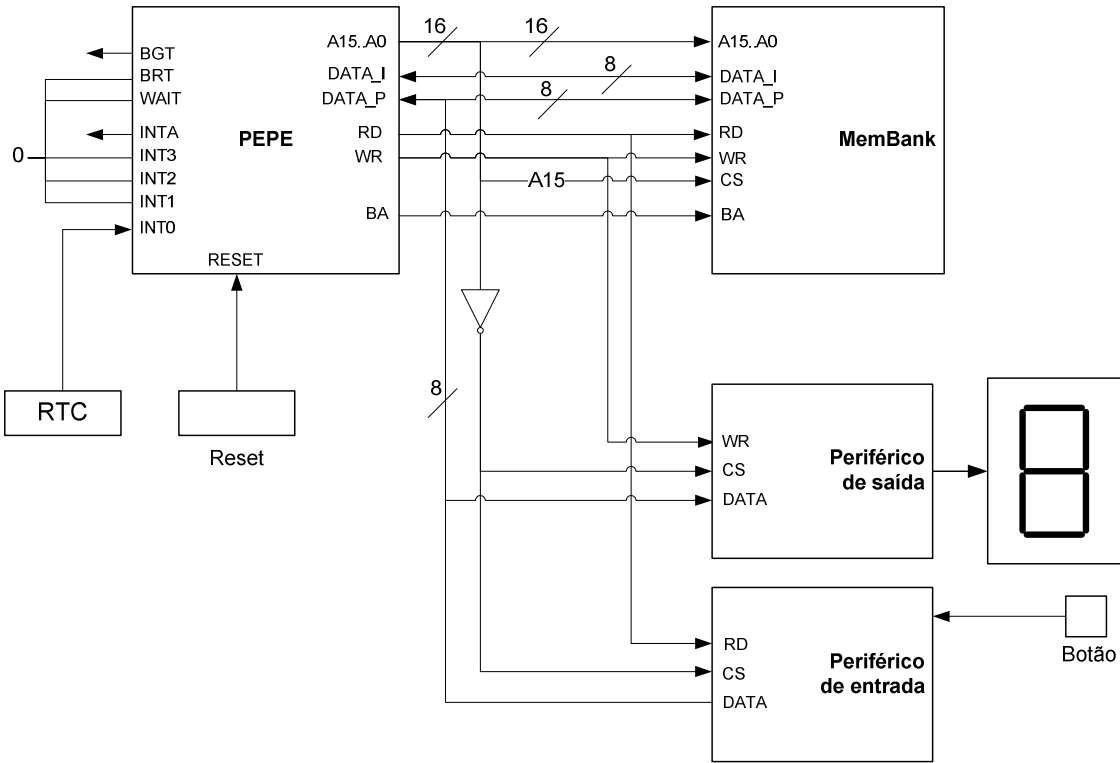
Outro aspeto importante é a relação entre as interrupções e os processos, para o que iremos recuperar o programa do Guião 6 (reveja este guião), que está repetido no ficheiro **lab7b.asm**.

Por exemplo, em vez de se estar a ler o valor do RTC num bit do periférico de entrada, pode ligar-se o RTC a uma interrupção, cuja rotina coloca um valor numa variável que um processo (P2, neste caso) estará a ler. A vantagem disto é que os processos não têm de lidar com os periféricos diretamente.

Para testar isto, faça as seguintes alterações numa cópia do **lab7b.asm**:

- Use o **lab7b.cmod** (cuja figura está na página seguinte), em que o RTC está ligado à interrupção INT0 e o botão está ligado ao bit 1 do periférico de entrada;
- Programe uma rotina de interrupção (sensível ao flanco), que coloca um valor (pode usar o ON) numa variável de comunicação com o processo P2 (use um registo que ainda não tenha sido usado. Atenção que este registo tem de ser inicializado no início, com OFF).
- Modifique o processo P2 para, em vez de estar a ler o periférico de entrada, verificar se último registo tem o valor ON. Se tiver, deve lá colocar OFF e avançar para o incremento do contador.

A tabela A.9 do livro contém informação sobre cada instrução do PEPE.



#### **4 – Objetivos a cumprir para o Projeto**

Na semana seguinte à deste guião, e em termos do projeto, deve ser mostrado ao docente uma versão do programa do projeto a funcionar que inclua pelo menos parte das especificações do projeto que devam ser implementadas com interrupções.

Deve usar o circuito do projeto, **jogo.cmod**, e não os circuitos deste guião.