

Guião de Laboratório de Arquitectura de Computadores

Simulação 5.3 – Ordenação por bolha com apontadores

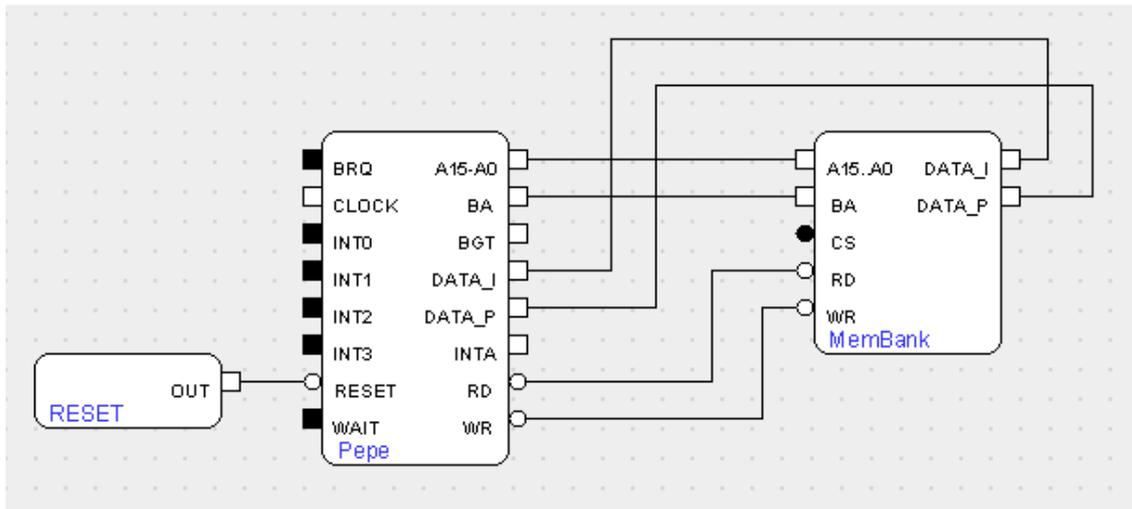
1 – Objectivos

Esta simulação ilustra o funcionamento do programa em linguagem assembly da Tabela 5.6. Os aspectos cobertos incluem os seguintes:

- Execução passo a passo e com pontos de paragem;
- Verificação da evolução dos registos relevantes e da memória, iteração a iteração, nomeadamente o apontador e a célula para que ele aponta.

2 – Circuito

O ficheiro “pepe.cmod” implementa o circuito da Fig. 4.7. A simulação 4.1 contém indicações mais detalhadas sobre a sua utilização no simulador.



3 – Simulação do programa da Tabela 5.6

Carregue este circuito no simulador e passe para Simulação.

Abra o painel do PEPE e compile e carregue (📁) o ficheiro “programa-tabela-5-6.asm”.

Esta simulação é semelhante à simulação 5.1, com a diferença que em vez de usar um array com o estrutura de dados se usam apontadores.

Usar um array corresponde a ter um registo com o endereço de base (endereço do primeiro elemento do array), que na simulação 5.1 (e no programa da Tabela 5.2 – não confundir com programa 5.2, que está na Tabela 5.6) corresponde ao registo R0. Os

acessos à memória são sempre feitos por soma ao R0 (que nunca muda) de registos que permitem aceder ao N-ésimo elemento do array.

Usar apontadores significa ter um endereço directo para o elemento em causa e ir variando esse endereço caso se pretenda aceder a vários elementos do array. No programa da Tabela 5.6, o R0 é também usado para aceder à memória, mas em cada iteração o R0 é incrementado, ao contrário do caso do array em que R0 nunca muda.

Execute as instruções passo a passo e vá vendo os registos, percebendo o que o programa vai fazendo.

Verifique a correspondência com o programa em C, para mais facilmente perceber o que o programa está a fazer. O fluxograma da Fig. 5.3 mantém-se válido, a forma de aceder aos dados é que variou..

Quando quiser, coloque pontos de paragem em pontos estratégicos, nomeadamente no fim dos ciclos de iteração, para poder inspeccionar o valor dos registos e da memória, e passe para modo de execução contínua, carregando no botão .

No mínimo, recomenda-se a colocação de um ponto de paragem na instrução JMP iteração e outro na instrução JMP fim (para se ver quando o programa acaba).

Ao longo da execução do programa (apenas quando ele estiver parado, em passo a passo ou num ponto de paragem), pode ver-se o resultado na própria memória, a partir do endereço 1000H, exactamente como indicado na Fig. 5.1.

Note que nas últimas iterações não há alterações e parece que o programa não faz nada, mas após a iteração 3 da ronda 4 o programa chega finalmente à última instrução.