

Guião de Laboratório de Arquitectura de Computadores

Simulação 6.3 – Ciclos de acesso à memória e periféricos

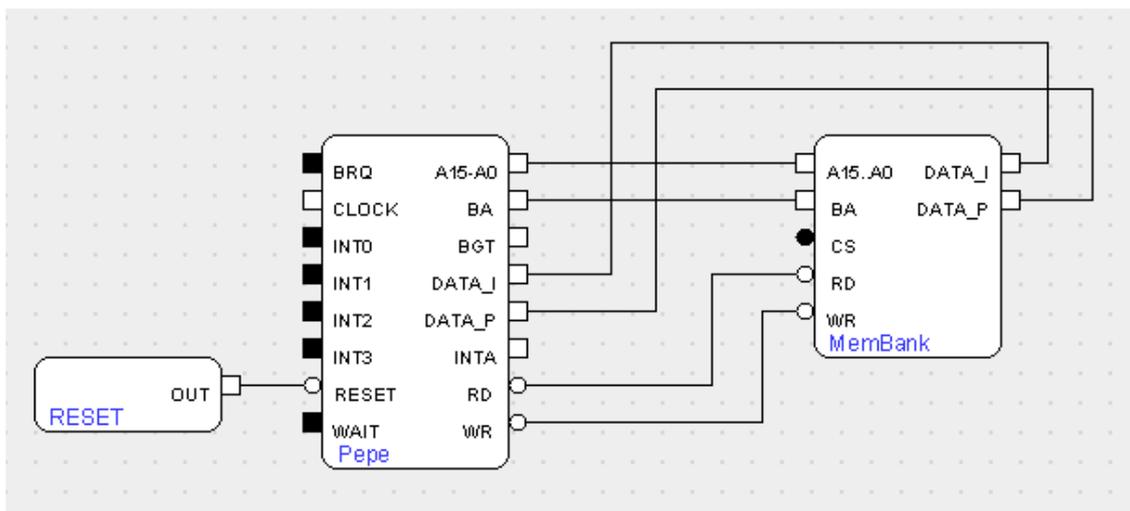
1 – Objectivos

Esta simulação exemplifica o funcionamento do processador nos ciclos de acesso à memória. Os aspectos cobertos incluem os seguintes:

- Evolução detalhada dos sinais durante um ciclo de leitura;
- Evolução detalhada dos sinais durante um ciclo de escrita;
- Verificação da evolução dos sinais nos vários barramentos durante o processamento completo de uma instrução SWAP (troca de um valor de um registo com o de uma célula de memória), incluindo leitura da própria instrução da memória.

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

Carregue este circuito no simulador e passe para Simulação.

Abra o painel do PEPE e compile e carregue (📁) o ficheiro “acesso-memoria.asm”, que contém um programa em assembly muito simples, mas que exercita os acessos à memória, em 16 e em 8 bits:

```

PLACE 1000H
base: WORD 1234H      ; palavra de memória a ler e a escrever
      WORD 5678H      ; palavra de memória a ler e a escrever

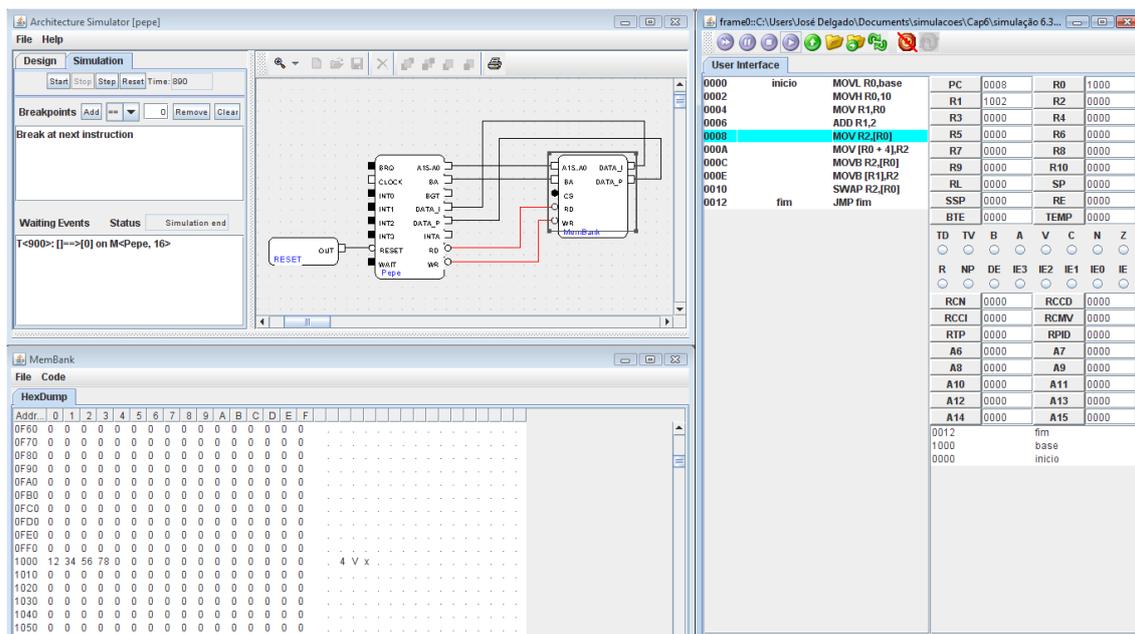
PLACE 0000H          ; localiza bloco de código
inicio:
MOV   R0, base       ; endereço de memória a ler e escrever
MOV   R1, R0
ADD   R1, 2          ; R1 = R0 + 2 (palavra seguinte)
MOV   R2, [R0]       ; leitura em 16 bits
MOV   [R0+4], R2     ; escrita em 16 bits
MOVB  R2, [R0]       ; leitura em 8 bits
MOVB  [R1], R2       ; escrita em 8 bits
SWAP  R2, [R0]       ; leitura e escrita em 16 bits
fim:  JMP   fim       ; acaba aqui

```

Execute em passo a passo os dois MOVs iniciais e o ADD. Quando a barra azul ficar na instrução MOV R2, [R0], arrange forma de no écran conseguir ver os seguintes painéis, sem sobreposição:

- Do simulador (neste, deverá conseguir ver o circuito completamente. Use as barras de scroll, se necessário);
- Do pepe;
- Da memória (deverá ver a linha do endereço 1000H).

Por exemplo, desta forma:



A partir de agora, não carregue mais no botão de passo a passo do PEPE (), mas sim no botão de passo a passo no próprio simulador ().

De cada vez que carregar neste botão, o simulador executa uma iteração de simulação, em que são executados os módulos cujas entradas variaram de valor na última iteração de simulação.

É normal haver várias iterações no mesmo ciclo de relógio, até os sinais estabilizarem. O PEPE é um circuito complexo e não é praticável estar a ver exactamente tudo o que

sucedem em cada iteração de simulação. Mas, com cuidado, consegue ver-se o que vai sucedendo nos acessos à memória.

A partir de agora, e após cada STEP, terá de observar com atenção o valor em cada um dos seguintes sinais do PEPE:

- A15..A0 – Bus de endereços (indica o endereço da palavra de memória a ser acedida)
- BA – Byte addressing (se BA=1, o acesso é em byte; se BA=0, o acesso é em 16 bits);
- DATA_I – Bits 7..0 do bus de dados – Liga aos endereços ímpares;
- DATA_P – Bits 15..8 do bus de dados – Liga aos endereços pares;
- RD – Sinal de leitura (RD=0 se estiver a ser feita uma leitura, RD=1 caso contrário);
- WR – Sinal de escrita (WR=0 se estiver a ser feita uma leitura, WR=1 caso contrário).

Os sinais de um bit (BA, RD e WR) são fáceis de observar. A vermelho estão a 1, a preto estão a zero. Os restantes nunca mudam de cor, mas o seu valor pode ser visto colocando o cursor (sem fazer clique) sobre a ligação correspondente. O estado de alta impedância, quando for o caso, é representado por 'Z'.

O que se pretende é que vá clicando sucessivamente em STEP, e após cada um observar o estado dos sinais e comprovar os diagramas das figuras 6.18 e 6.19.

É importante notar que:

- A janela no campo inferior esquerdo do simulador indica os eventos que ficaram programados para ser executados quando carregar em STEP da primeira vez. Indicam o tempo de simulação em que ocorrerão mudanças em dados sinais de certos módulos (tudo é indicado). O simulador avança o tempo de simulação automaticamente para o primeiro (em ordem temporal) evento a ser executado;
- A execução de cada instrução é precedida da sua busca (leitura) da memória. Por isso, conte com esses acessos pelo meio dos acessos de dados decorrentes dos MOVs;
- A barra azul no painel do PEPE evolui de vez em quando (ao acabar a instrução e começar outra), mas note que vai em avanço. Quando a barra muda, é na altura em que o simulador vai começar a fazer a busca e depois a execução da instrução anterior;
- Pode haver alguns STEPs em que nada sucede na interface de memória, enquanto o PEPE faz processamento interno.