Guião de Laboratório
de
Arquitectura de Computadores

Simulação 3.6 – Funcionamento dos periféricos

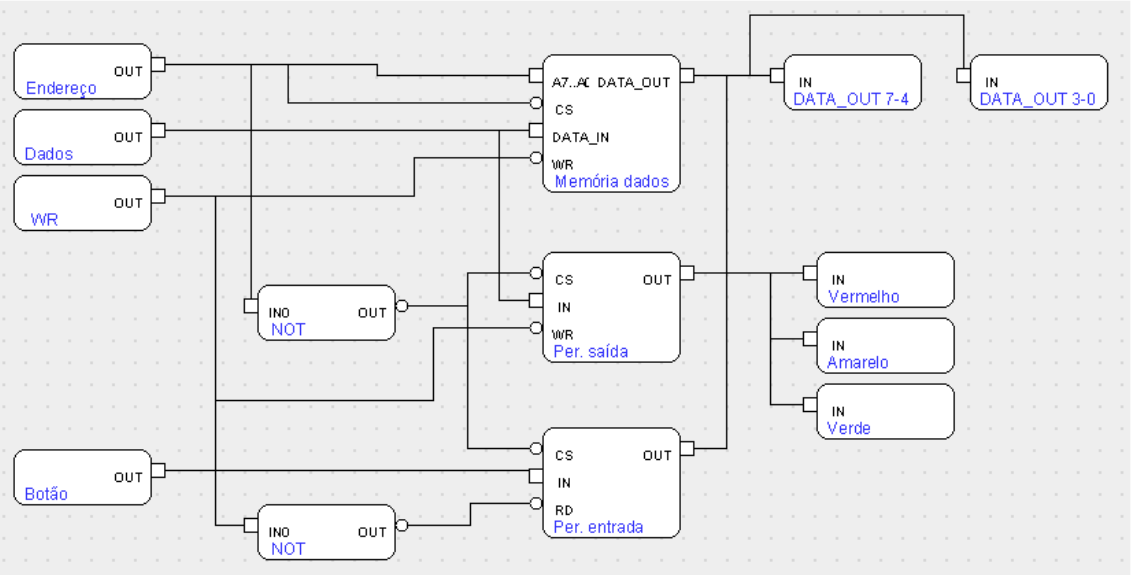
1 – Objectivos

Esta simulação ilustra o funcionamento dos periféricos. O circuito a usar é basicamente o da Fig. 3.26, mas em que o processador é substituído por dispositivos que permitem controlar manualmente os sinais normalmente gerados pelo PEPE-8, para ilustrar o funcionamento dos periféricos em detalhe e com controlo total. Os aspectos cobertos incluem os seguintes:

- Funcionamento básico do periférico de saída (com memorização do valor), e do periférico de entrada, com ênfase na sua interface tristate;
- Distinção entre acesso à memória e acesso aos periféricos, com ilustração do que acontece quando se activa simultaneamente a memória e um periférico, quer em leitura quer em escrita.

2 – Circuito

O ficheiro "semaforos-perifericos.cmod" implementa o circuito da Fig. 3.26, mas em que o processador é substituído por dispositivos que permitem controlar manualmente os sinais normalmente gerados pelo PEPE-8.



O bus de endereços e de dados (de saída) do PEPE-8 são simulados pelos módulos de entrada hexadecimal "Endereço" e "Dados" (8 bits cada). O interruptor WR simula o sinal correspondente do PEPE-8.

Os displays DATA_OUT, de 4 bits cada um, permitem ver o valor lido da memória ou do periférico de entrada.

Notas:

- Tanto o sinal CS da memória como o NOT de cima estão ligados ao bit de maior peso do bus de endereços. Como resultado, os endereços de 00H a 7FH (bit de maior peso a 0) activam a memória. Os endereços 80H a FFH (bit de maior peso a 1) activam os CS dos periféricos (os sinais CS são activos a 0);
- Com o interruptor WR a 0, escreve-se na memória ou no periférico de saída (o sinal CS discrimina qual). Com WR = 1, lê-se a memória ou o periférico de entrada (o sinal CS discrimina qual).

A Tabela 3.15 clarifica estes aspectos.

3 – Simulação

Carregue este circuito no simulador e passe para Simulação. Abra o painel de controlo do botão WR e coloque-o a 1 (para não escrever na memória).

Faça START e efectue as esperiências seguintes.

Ler da memória

1. Abra (com duplo clique) os paineis de controlo dos displays dos módulos de entrada e da memória;
2. Certifique-se que o interruptor WR está a 1;
3. No Endereço coloque um valor menor que 80H (carregando no seu botão OK), por exemplo 04H. Verifique que nos displays aparece o valor que a memória contém nesse endereço.

Escrever na memória

1. No módulo Dados coloque um valor qualquer, em hexadecimal, entre 00H e FFH (carregue em OK);
2. Coloque o interruptor WR com o valor 0;
3. Verifique que o valor que indicou aparece no painel da memoria e nos displays;
4. Indique um novo valor no módulo Dados e carregue em OK. Verifique que o novo valor aparece no painel da memória e nos displays;
5. Coloque o interruptor WR com o valor 1, para deixar de escrever;
6. Indique um novo valor no módulo Dados e carregue em OK. Verifique que o novo valor não afecta nem painel da memória nem os displays. A memória memorizou, ou seja, tem um comportamento de trinco.

Escrever no periférico de saída (semáforo)

1. Abra os paineis de controlo dos leds para visualizar as luzes;

2. Proceda da mesma forma que para escrever na memória, mas agora especificando o endereço 80H (ou outro até FFH);
3. Note que apenas os três bits de menor peso afectam os leds;
4. O periférico de saída tem também um comportamento de trinco. Com WR=0 está transparente, com WR=1 memoriza.

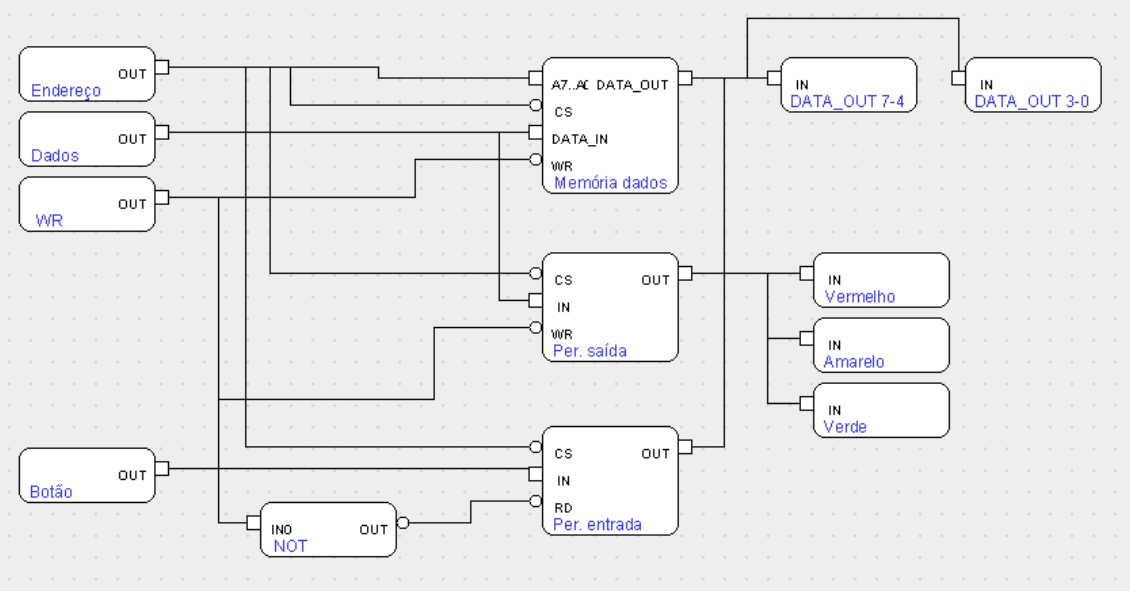
Ler do periférico de entrada (botão)

1. Certifique-se que WR=1;
2. Abra o painel de controlo do botão e experimente carregar no botão de pressão. Verifique que o display de menor peso passa a 1 enquanto está a carregar no botão e volta a 0 quando deixa de carregar. Isto quer dizer que o periférico de entrada está a ler, e que só o bit de menor peso, aquele a que o botão liga, é afectado.
3. O periférico de entrada está sempre a ler até WR deixar de estar a 1 ou o endereço seja menor que 80H (para o bit de menor peso ser 0).

4 – Conflito entre a memória e os periféricos

Uma das regras básicas de qualquer computador é um acesso a um endereço nunca activar mais de um dispositivo na mesma operação (leitura ou escrita), sob pena do funcionamento incorrecto do circuito.

Para ilustrar isto, carregue no simulador o ficheiro "semaforos-conflito.cmod", em que o NOT que faz actuar os CS da memória e dos periféricos em separado foi eliminado. Todos os CS ligam agora ao bit de maior peso do bus de endereços. Qualquer acesso a um endereço menor que 80H (para o bit de maior peso ser 0, valor a que os CS são activos) faz activar quer a memória, quer os periféricos.



Passe para Simulação, faça START e abra os paineis de controlo de todos os dispositivos.

Com WR=0 (escrita), verifique que ao colocar um valor no módulo Endereço faz aparecer o valor da célula de memória desse endereço nos displays (pode alterar o valor dessa célula no painel da memória) e os três bits de menor peso nos leds do semáforo. Ou seja, a escrita está a ocorrer ao mesmo tempo na memória e nos semáforos. É um funcionamento incorrecto.

Coloque agora WR=1, o que activa a saída do periférico de entrada e da memória ao mesmo tempo (estão ambos em leitura). O resultado é termos dois dispositivos a querer forçar a ligação para dois valores diferentes.

Em termos de hardware, isto poderia ser o suficiente para provocar uma avaria permanente no circuito, por correntes excessivas. Em simulação, o simulador reporta simplesmente um erro de conflito no bus, na parte de baixo da janela do simulador, do género:

ERROR: Time: 63:Update elements: ist.ac.simulador.nucleo.SSignalConflictException: STRONG CONFLICT on Connection Connection between P<DATA_OUT, 8> of M<Memória dados, 8>, P<IN, 4> of M<DATA_OUT 7-4, 4>, P<IN, 4> of M<DATA_OUT 3-0, 4> and P<OUT, 8> of M<Per. entrada, 8>

O circuito não funciona bem sem corrigir este problema.