Guião de Laboratório
de
Arquitectura de Computadores

Simulação 3.2 – Unidade de dados

1 – Objectivos

Esta simulação ilustra a utilização do circuito da Fig. 3.10, incluindo o da Fig. 3.11. Os aspectos cobertos incluem os seguintes:

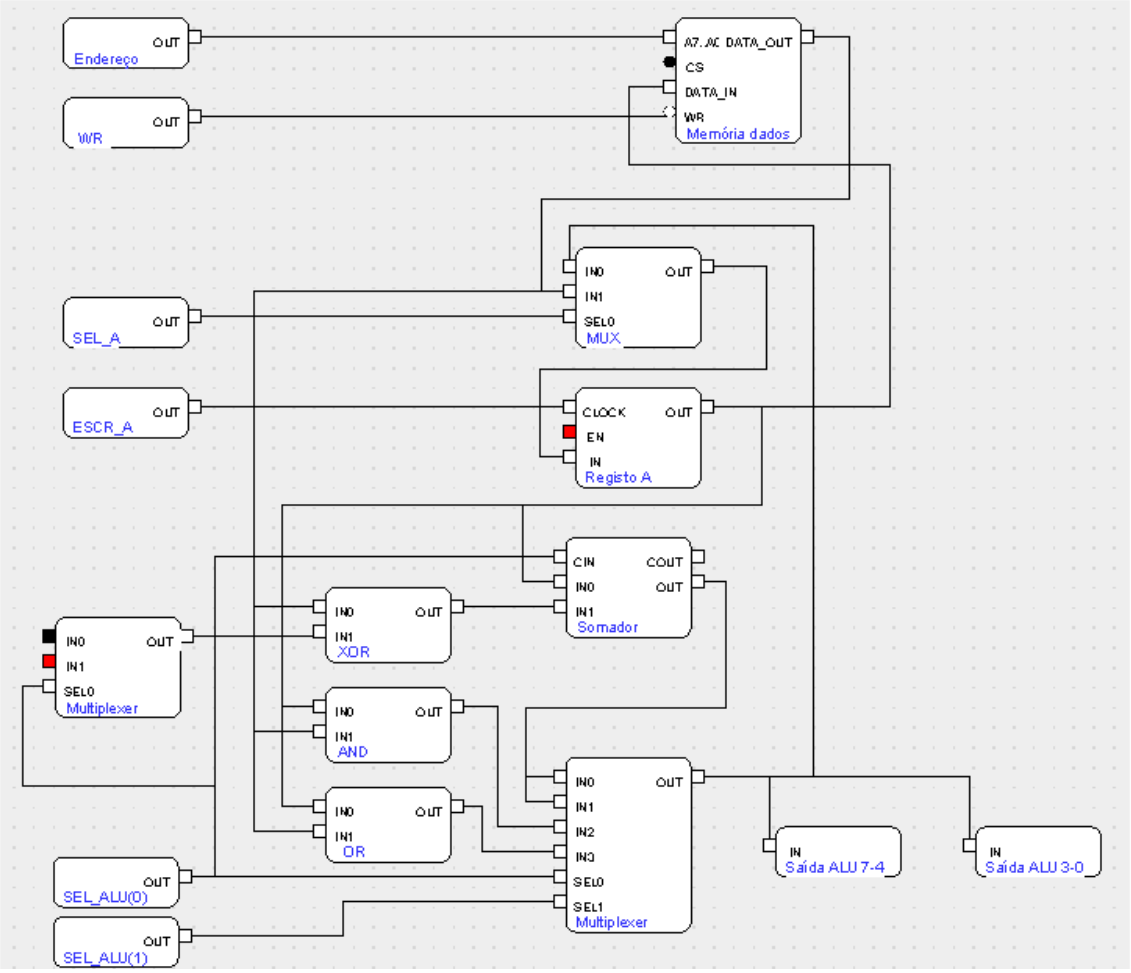
- Inicialização da memória de dados (valores dos operandos) pelo utilizador;
- Especificação pelo utilizador de cada um dos sinais da Fig. 3.10;
- Verificação do funcionamento de cada operação básica por actuação desses sinais, com visualização dos dados presentes em cada um dos pontos do circuito, incluindo o interior da ALU;
- Detalhes do funcionamento dos sinais WR e ESCR_A (flancos em que são activos);
- Implementação de uma soma completa (algoritmo 2);
- Implementação das restantes operações suportadas pela ALU;
- Verificação da possibilidade de utilização do resultado de uma operação no registo A como operando da operação seguinte.

2 – Circuito

O ficheiro "unidade-dados.cmod" implementa o circuito da Fig. 3.10, com a ALU implementada pelo circuito da Fig. 3.11. A memória tem 8 bits de dados e de endereços. Todos os módulos por onde os dados podem passar (registo, multiplexers, somador, XOR, AND e OR, têm 8 bits, tal como indicado na Fig. 3.11.

Os botões e o módulo de entrada (endereço), do lado esquerdo, permitem especificar os diversos sinais necessários para controlar o circuito. O multiplexer do lado esquerdo é auxiliar, tem 8 bits, uma entrada forçada a 00H e outra a FFH e destina-se simplesmente a produzir um número de 8 bits, todos a 0 ou todos a 1, de acordo com o sinal SEL_ALU(0), de modo que o XOR de 8 bits complemente todos os bits ou não. A Fig. 3.11 é um circuito simplificado e naturalmente um único bit (SEL_ALU(0)) não pode alimentar um XOR de 8 bits (que é um conjunto de 8 XORs de 1 bit cada).

Os displays que ligam à saída da ALU são apenas para mais ficilmente se ver o valor desta saída. O valor de qualquer sinal pode ser visto, em modo de simulação, colocando o cursor por cima de qualquer sinal, sem carregar no botão do rato.



3 – Simulação

Passe para modo de simulação (sem fazer START) e abra os painéis de controlo dos vários dispositivos de entrada e dos displays (estes lado a lado, com o que liga aos bits 3 a 0 da ALU do lado direito), para formarem um byte.

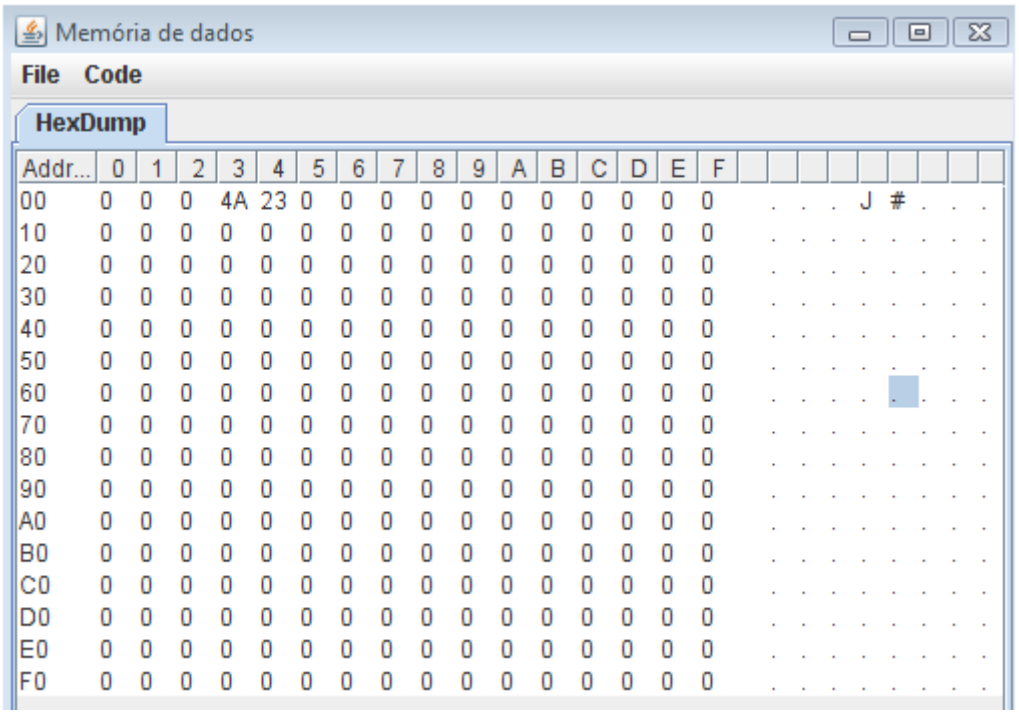
Os painéis dos botões precisam de ser abertos (arrastando o bordo direito da sua janela, por exemplo), para revelarem o seu nome.

Coloque o botão WR a 1, para não escrever um valor qualquer na memória. O sinal WR é activo a 0.

Comecemos por colocar dados na memória de dados. Abra o painel de controlo da memória e coloque os valores 4AH e 23H nos endereços 3 e 4, respectivamente. Estes são os valores (dados e endereços) usados nos exemplos, mas naturalmente pode usar quaisquer outros valores à sua vontade.

Note que um valor na memória só fica efectivo depois de retirar o cursor da célula onde escreveu o valor.

Se quiser, pode guardar estes valores num ficheiro (através do menu no painel) para mais tarde os recuperar.



Addr...	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0	0	0	4A	23	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Faça agora START e proceda às experiências seguintes.

Guardar no registo A o valor que se encontra no endereço 03H (A<-M[endereço])

1. Coloque no módulo Endereço o valor 3 (com OK). Note que na saída da memória aparece o valor 4AH
2. Coloque no botão SEL_A o valor 1, para que o valor lido da memória chegue à entrada do registo.
3. No botão ESCR_A mude o valor 0 para 1, para memorizar o valor no registo. Este sinal é activo no flanco ascendente, pelo que o valor 4AH aparece logo na saída do registo.
4. Passe ESCR_A para 0, de modo a ficar preprado para nova escrita.

Observar as diferentes operações da ALU (operandos A e M[endereço])

1. Coloque no módulo Endereço o valor 4 (com OK), de modo a fazer aparecer o valor 23 na saída da memória.
2. Observe que o valor que aparece nos displays é 6DH, o resultado da soma na ALU entre o valor do registo A (4AH) e da célula de memória com o endereço 4 (que contém o valor 23).
3. Coloque no botão SEL_ALU(0) a 1, o que selecciona a subtracção (por complementar para 2 o valor 23 à entrada do somador, através do XOR). A saída da ALU fica com o valor 27H (4AH-23H).
4. Coloque no botão SEL_ALU(0) a 0 e SEL_ALU(1) a 1, o que selecciona a operação AND. A saída da ALU fica com o valor 02H (4AH AND 23H).
5. Coloque no botão SEL_ALU(0) a 1e SEL_ALU(1) a 1, o que selecciona a operação OR. A saída da ALU fica com o valor 6BH (4AH OR 23H).

Guardar no registo A o resultado da soma entre o valor contido no registo A e valor contido no endereço 04H (A<-A+M[endereço])

1. Volte a colocar no botão SEL_ALU(0) a 0e SEL_ALU(1) a 0, o que selecciona a operação soma. A saída da ALU fica com o valor 6DH (4AH + 23H)
2. Coloque no botão SEL_A o valor 0, para que a saída da ALU chegue à entrada do registo A.
3. Passe o botão ESCR_A de 0 para 1 e para 0 novamente. O registo memoriza o valor 6DH

Guardar o valor do registo na célula de memória com o endereço 10H (M[endereço] <- A)

1. Coloque no módulo Endereço o valor 10H (com OK).
2. Passe o valor do botão WR de 1 para 0 e para 1 novamente. O valor 6DH do registo é memorizado na memória. Verifique-o no painel de controlo da memória.

Utilizar o valor do registo na operação seguinte

1. Repita a operação de guadar no registo o resultado de uma operação na ALU, mudando eventualmente a operação da ALU por meio dos sinais SEL_ALU. O valor actual do registo, que foi o resultado da última operação, é agora o novo operando.