

Guião de Laboratório de Arquitectura de Computadores

Simulação 4.16 – Máscara AND

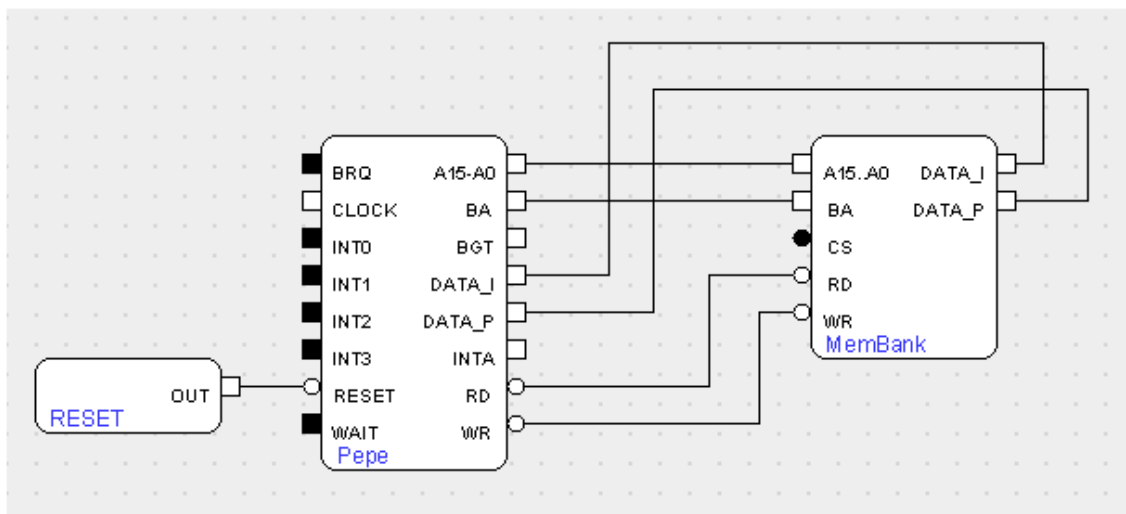
1 – Objectivos

Esta simulação ilustra o funcionamento das instruções de manipulação de um só bit, tendo por base os programas 4.18. Os aspectos cobertos incluem os seguintes:

- Execução passo a passo e com pontos de paragem do programa;
- Inicialização do conteúdo da memória (cadeia de caracteres);
- Verificação da evolução dos registos relevantes e da memória, iteração a iteração.

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 4.18


Carregue este circuito no simulador e passe para Simulação.

Abra o painel da memória e carregue (com File->Load) o ficheiro “setenta.dat”. Este ficheiro contém alguns números a partir do endereço 1000H (é preciso ir até lá no painel da memória):

70H, F0H, 7FH, 00H, 71H, 81H, 91H, A0H, 0AH

Os restantes são zero.

Nota – Deve carregar o ficheiro de dados antes do programa.

Abra o painel do PEPE e compile e carregue () o ficheiro “programa4-18.asm”.



Execute as instruções passo a passo e vá vendo os registos, percebendo o que o programa vai fazendo. Note que há dois apontadores para as strings, origem e destino, que variam de 1 em 1 unidade porque cada byte é tratado à vez.

Imediatamente antes da execução da instrução AND R0, R3, anote o valor dos registos R0 e R3 e verifique que, após a execução desta instrução, R0 contém o resultado correcto da operação AND sobre os dois registos.

Note também o funcionamento da instrução CMP R0, R4 (compare), que faz a subtracção R0-R4 sem alterar nenhum dos dois registos, apenas para afectar os bits de estado e permitir saltar ou não de acordo com os valores dos dois registos.

Ao longo da execução do programa, pode verificar a evolução do R1, que endereça na memória o byte a testar, e R5, que é incrementado sempre que se descobre um byte cujo valor esteja no intervalo 70H a 7FH.

Como resultado final, R5 fica com 3, o que está correcto em relação à lista de valores indicada atrás (o resto dos valores é zero).

Quando perceber o funcionamento, correr o programa em execução contínua, carregando no botão . Para parar o programa, coloque primeiro um ponto de paragem na instrução “fim:” (clicando lá, o que coloca lá uma barra roxa) ou páre manualmente o programa (botão ).

NOTA – o programa demora alguns segundos a executar pois testa todos os bytes entre 1000H e 1FFFH.

Pode também colocar pontos de paragem (fazendo clique nas instruções onde quiser parar).