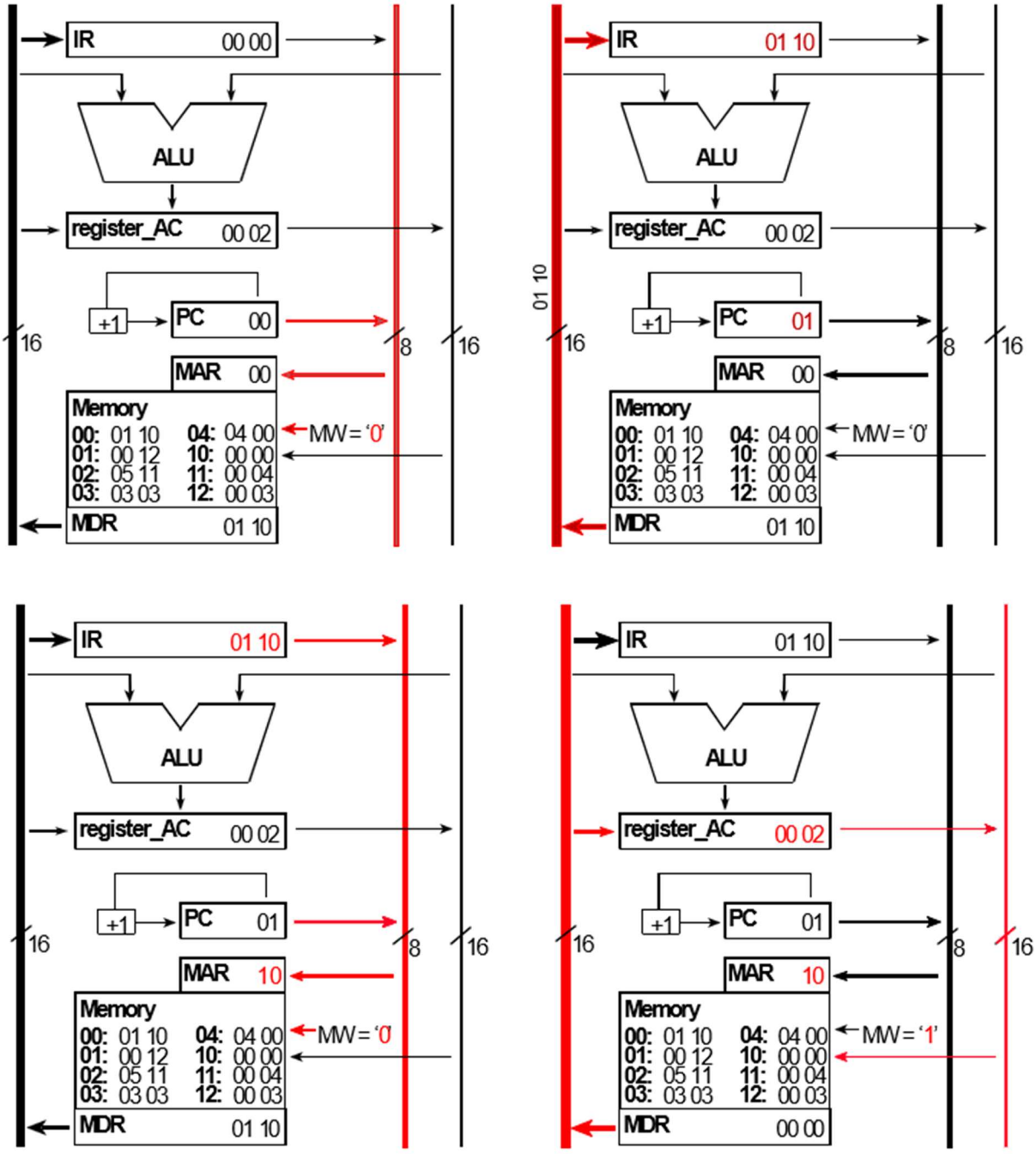
Fluxo de Dados – Nanoprocessador Sequencial:

Grupo 13: Eline Vieira, Giovana Maciel, João Averaldo, Rafael Mori, Renan Ueda.

- **Descrição de funcionamento das instruções:**

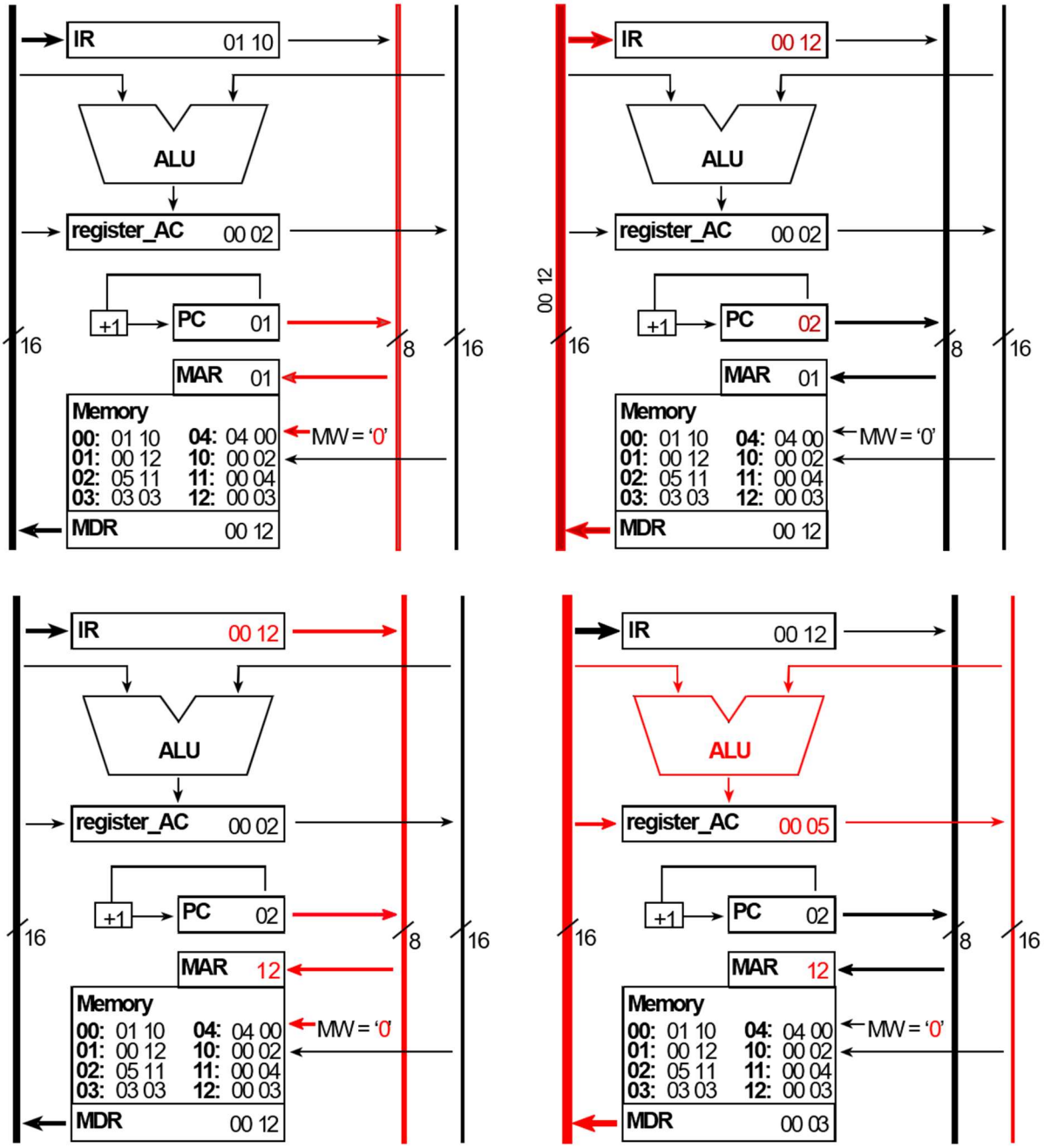
1. STORE:



- ✓ LER: Iniciamos com o conteúdo de endereço de memória igual a zero, o qual é repassado para o barramento de dados sinalizando quais bits devem ser armazenado em MDR (no caso, 0110);
- ✓ SALVAR: O registrador IR é ativado pelo pulso de *clock* e salva a instrução antes localizada em MDR (ou seja, 0110);
- ✓ INCREMENTAR: O registrador PC é ativado através do pulso de *clock* e salva 01 (00 + 01);

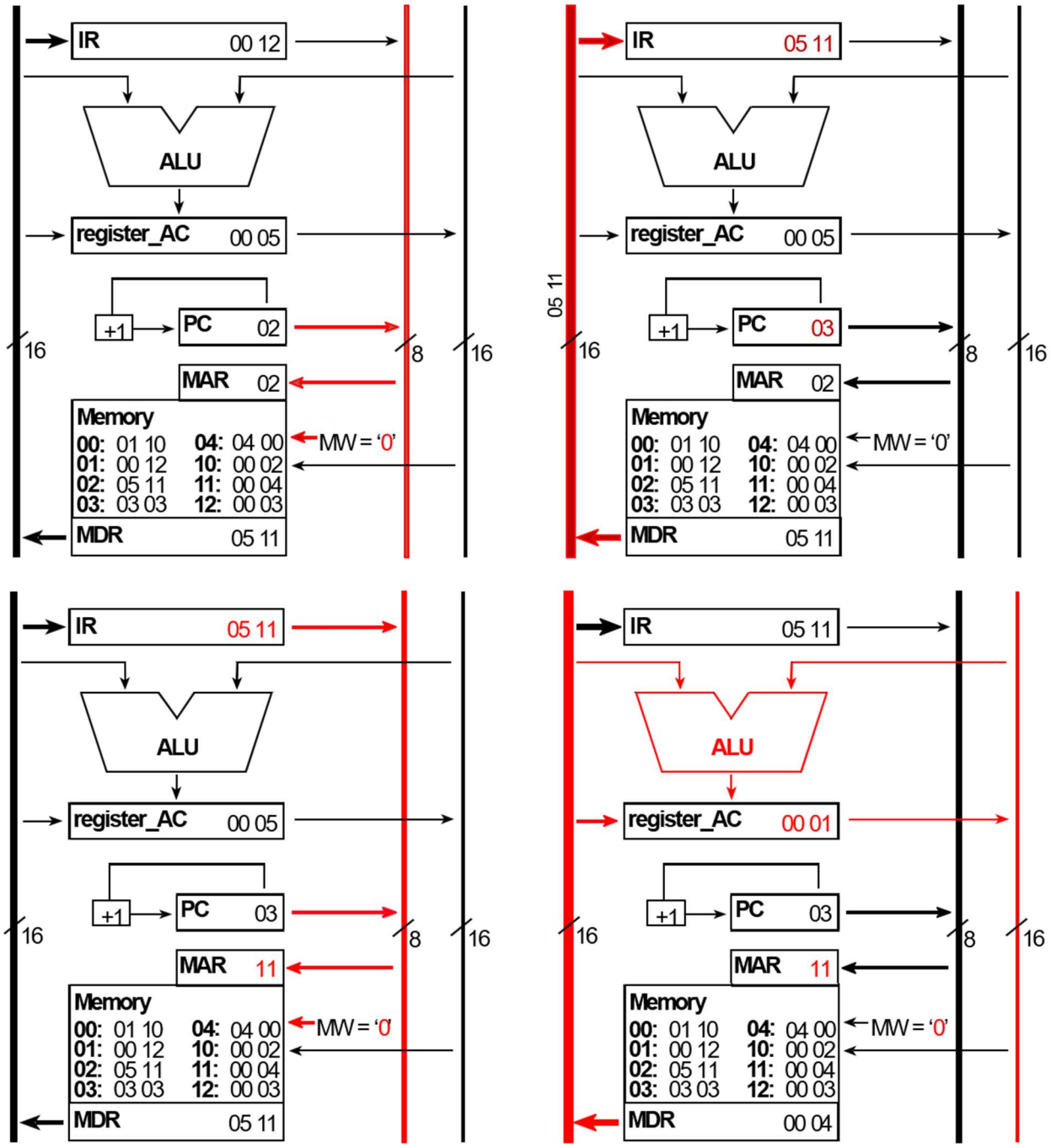
- ✓ DECODIFICAR: A instrução STORE com *address* igual a 10 é interpretada e lançada para o circuito. Note que isso é possível pelo auxilio do barramento de endereço;
- ✓ EXECUTAR: Por fim, ao encontrar o endereço de memória que deverá ser manipulado executa-se a operação e o conteúdo de AC é alocado no mesmo (é colocada a sequência 0002 em 10, como fora instruído pelo addrBus anteriormente).

2. SUM:



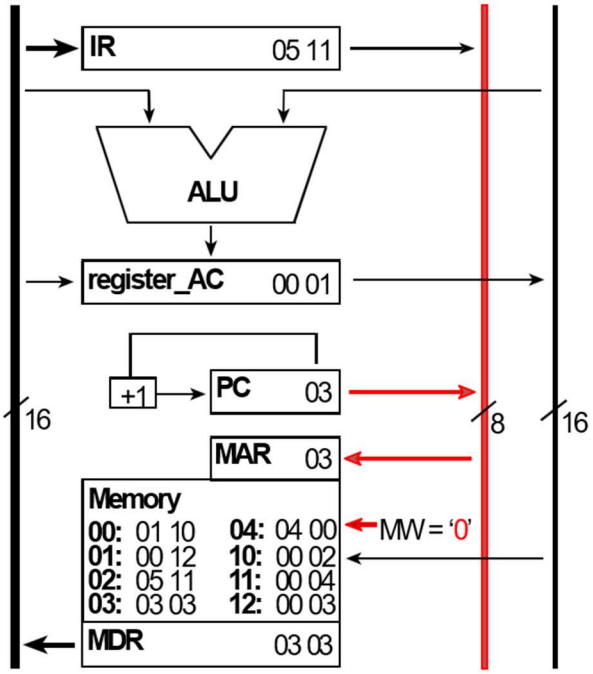
- ✓ LER: O conteúdo do endereço de memória é "inicializado" por PC e é colocado no barramento de dados, e através deste alcança MDR (com valor 0012);
- ✓ SALVAR: O IR é ativado pelo pulso de *clock* e salva a instrução 0012;
- ✓ INCREMENTAR: O PC é ativado pelo *clock* e salva PC+1, ou seja, 01+01 = 02;
- ✓ DECODIFICAR: Agora, o endereço da instrução (SUM 12) aloca-se no barramento de endereço e é lançado no circuito.
- ✓ EXECUTAR: Finalmente, com MAR em 12, os barramentos de dados são ativados e buscam as informações presentes em MDR e em AC, visando como resultado a soma dos valores encontrados nos respectivos locais. Ao final desta operação o valor retorna obtido retorna para AC (obtendo AC = 0005).

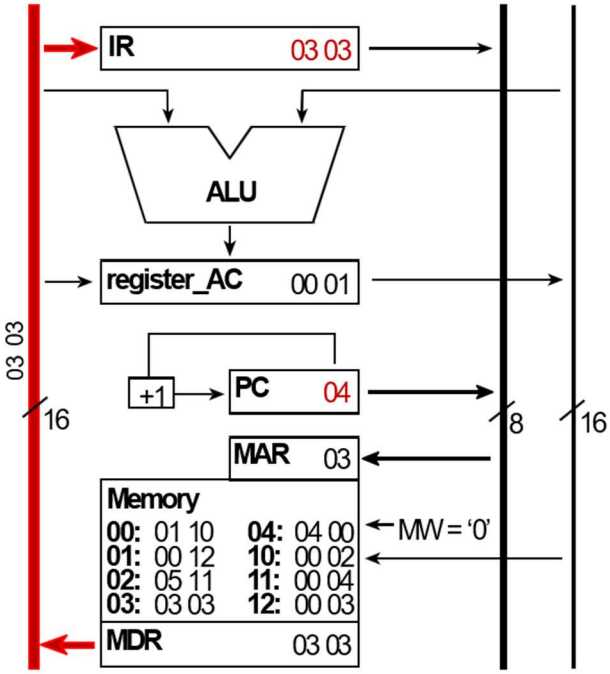
3. SUB:

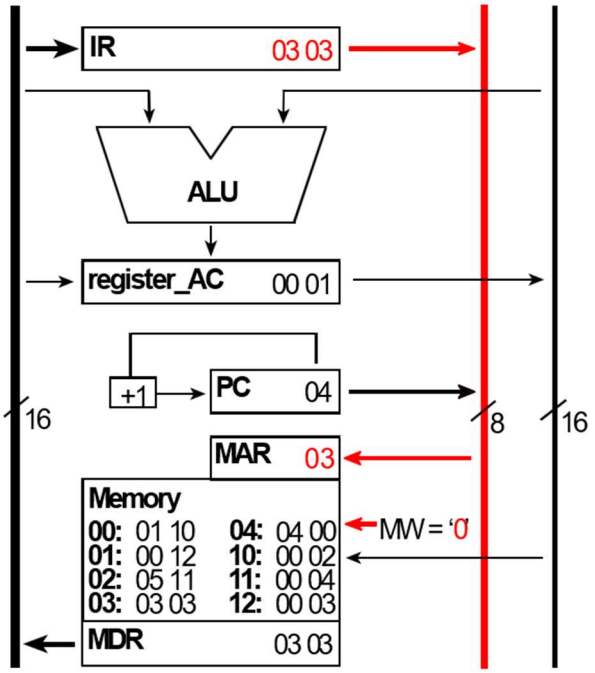


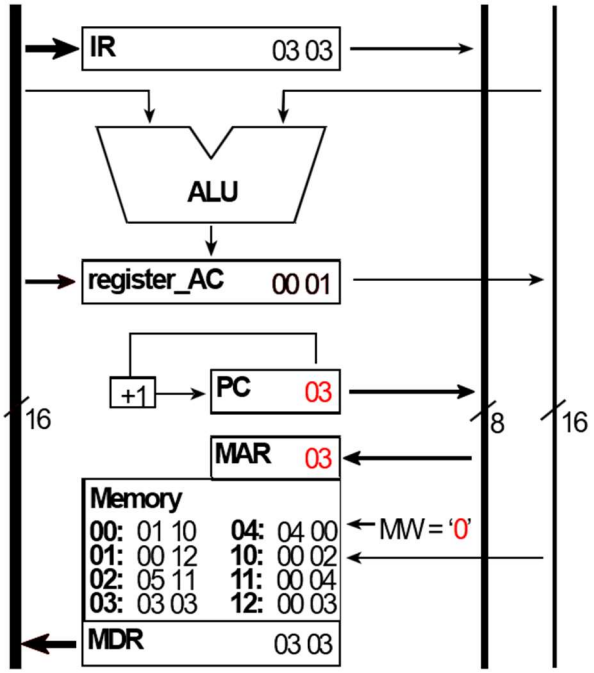
- ✓ LER: Nesta etapa começamos com o endereço de memória dois, o qual é repassado para o barramento de dados e alimenta o registrador de saída de memória MDR com 0511;
- ✓ SALVAR: Logo após, o IR é ativado pelo *clock* e a instrução 0511 é salva;
- ✓ INCREMENTAR: O registrador PC é ativado e salva PC+1, ou seja, 02+01 = 03;
- ✓ DECODIFICAR: A instrução é interpretada e é colocada no barramento de endereço para ser fornecida ao circuito (SUB 11), atualizando MAR e iniciando a leitura no componente de memória (em busca do endereço que está em MAR);
- ✓ EXECUTAR: Para finalizar, MDR recebe o valor que estava atrelado ao endereço em MAR (11) e o valor armazenado anteriormente pelo circuito é utilizado para realizar a operação de subtração entre a quantia presente em AC e o novo valor de MDR (obtendo AC = 0001).

4. JUMP:



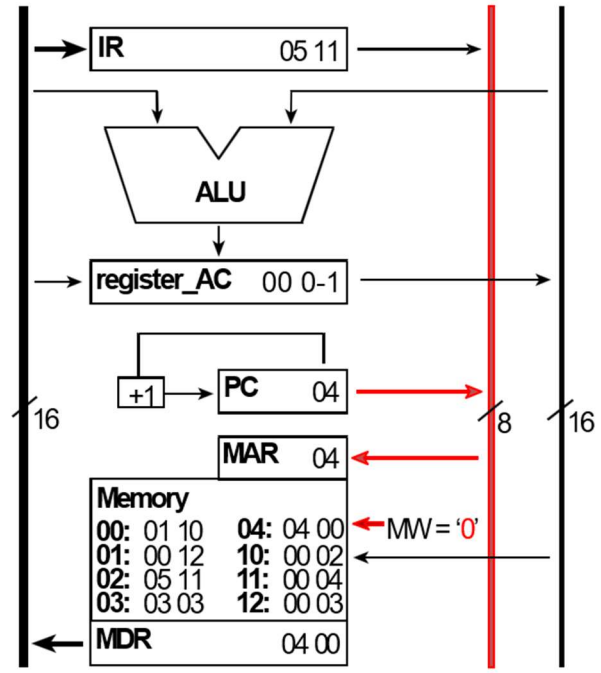


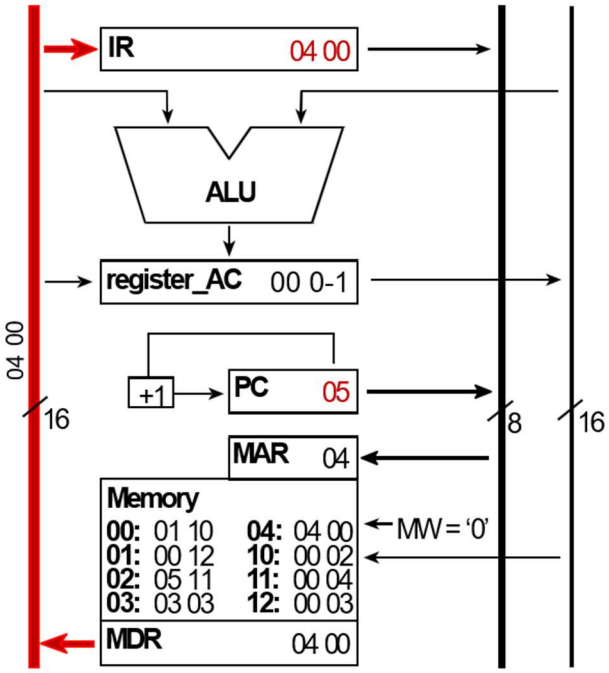


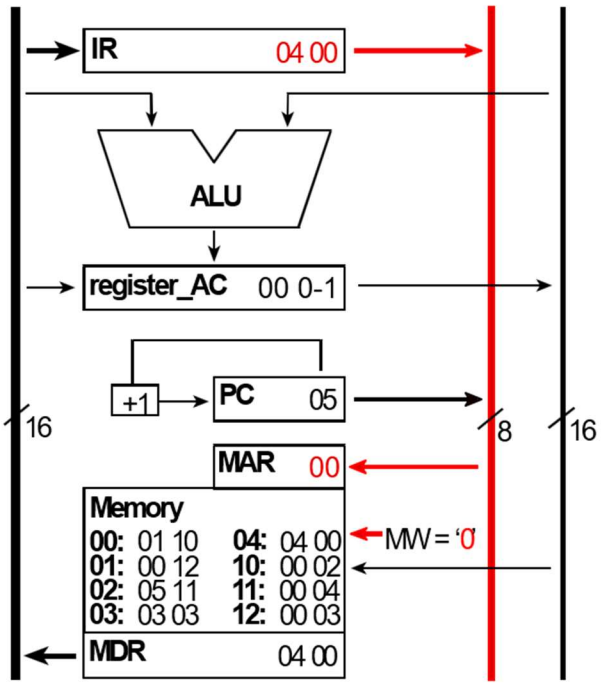


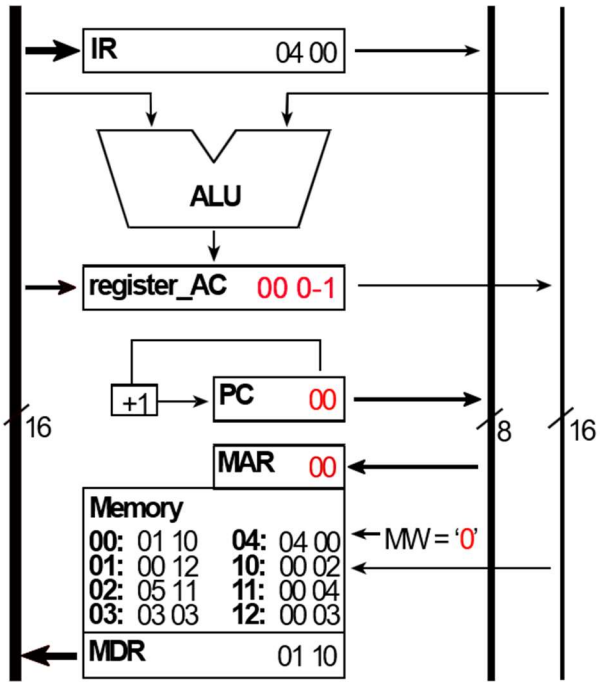
- ✓ LER: Durante essa operação, começamos com o endereço de memória igual a três, este por sua vez é colocado DataBus e registrado no MDR (0303);
- ✓ SALVAR: Com a ativação de IR é salva a instrução presente anteriormente em MDR (0303);
- ✓ INCREMENTAR: Mais uma vez, PC é incrementado e recebe PC + 1, ou seja, 03 + 01 = 04;
- ✓ DECODIFICAR: A instrução é captada e lida e colocada no barramento de endereço, atribuindo valor à MAR e varrendo a memória em busca da informação presente no endereço 03. Note que a interpretação gerada é JUMP 03;
- ✓ EXECUTAR: Concluindo, é retornado a PC o valor presente no *address*, fechando a execução do JUMP.

5. JNEG:









- ✓ LER: Nessa operação, começaremos com o endereço de memória quatro inserido no barramento de dados e com a instrução 0400 alocada no MDR;
- ✓ SALVAR: Com o pulso de *clock*, IR salva a instrução 0400;
- ✓ INCREMENTAR: Como ordinário, PC recebe seu incremento e aloca 04+01 = 05;
- ✓ DECODIFICAR: Durante a decodificação desta operação é preciso que o valor de MAR seja corretamente atribuído - através do barramento de endereço – e haja uma verificação com relação ao AC (que no caso deve ser menor que zero para passar para execução).
- ✓ EXECUTAR: Deste modo, é verificável que no registrador do exemplo há um número negativo para iniciar a execução da operação, portanto, PC recebe o *address* presente em MAR.

PS: As imagens estão dispostas da seguinte maneira: ler, salva e incrementar na "primeira imagem" e decodificar e executar na "segunda imagem". Note que cada operação possui duas imagens, as quais estão dispostas de modo a representarem a primeira e segunda imagem, respectivamente.