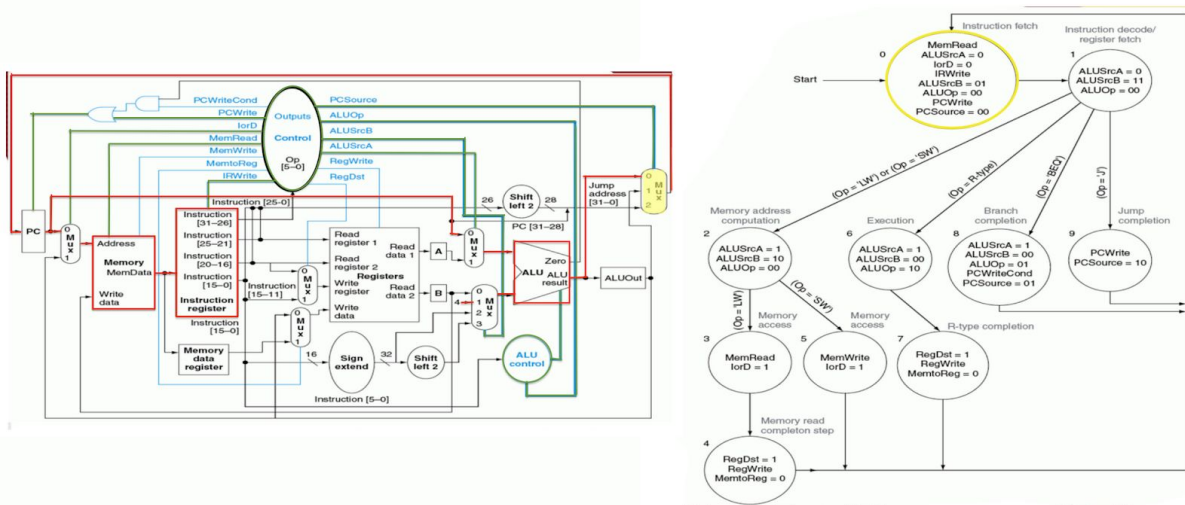


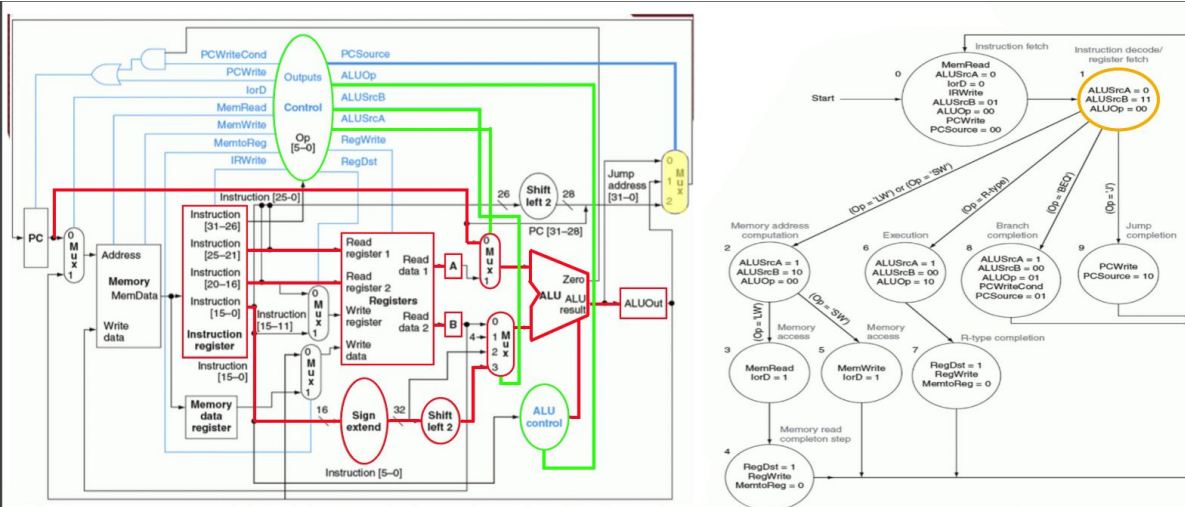
FETCH E DECODE

São os primeiros dois estados de qualquer ciclo de instrução, iguais em todas elas. Por isto, não repetiremos mais as imagens, e todas as instruções começarão como se estas duas etapas já tiverem sido cumpridas.

1. Fetch

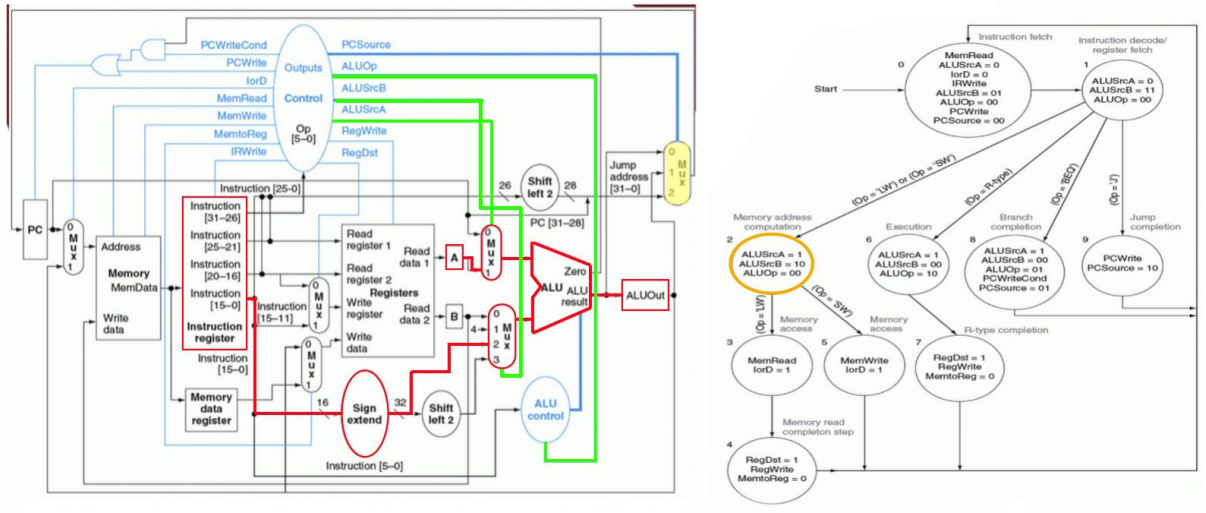


2. Decode

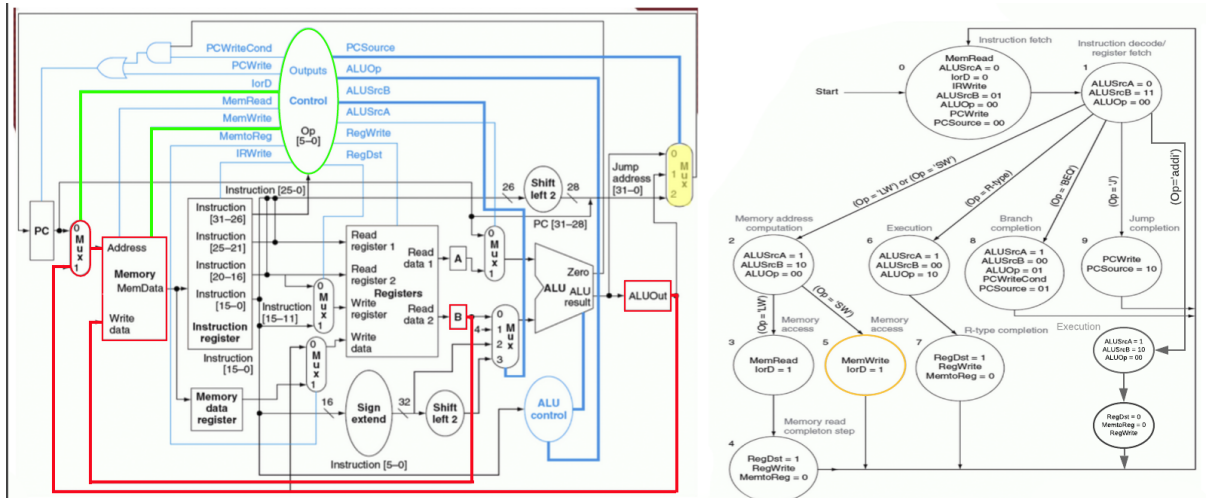


SW - Store Word

1. Fetch
2. Decode.
3. Memory Address Computation:

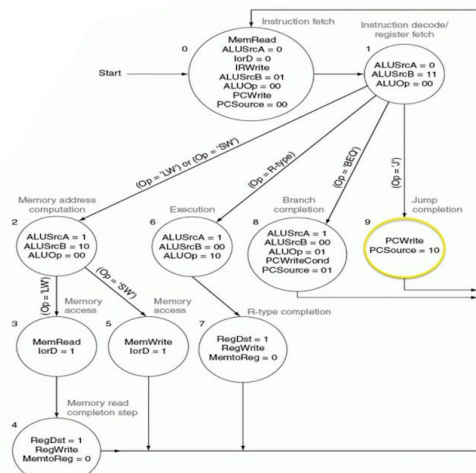
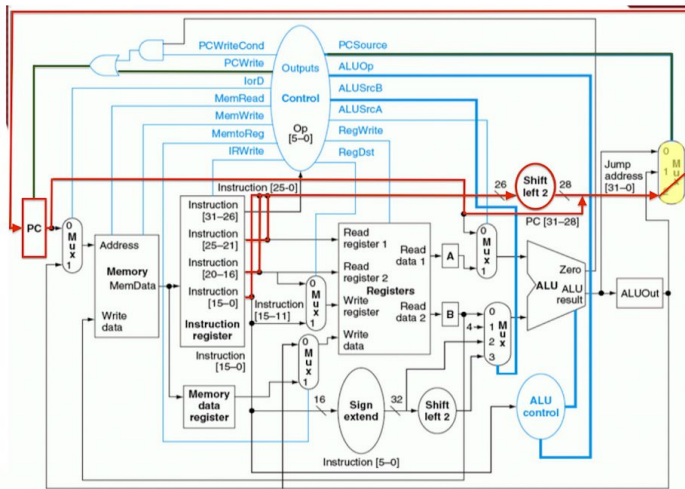


4. Memory Access:



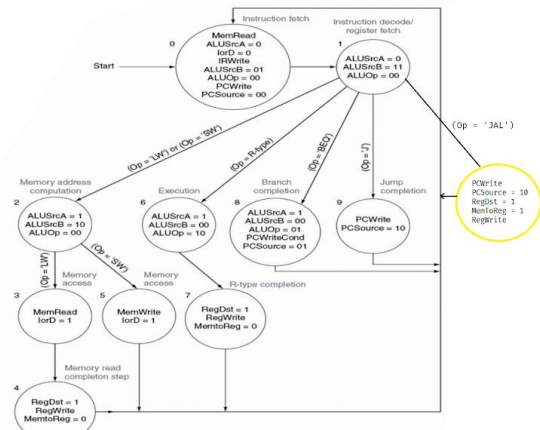
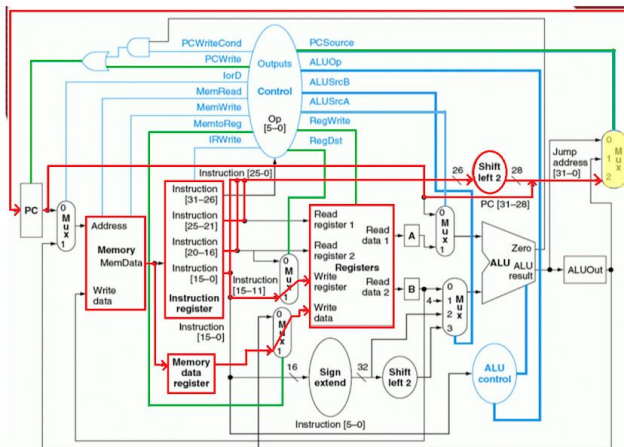
J - Jump

1. Fetch.
2. Decode.
3. Jump Completion:



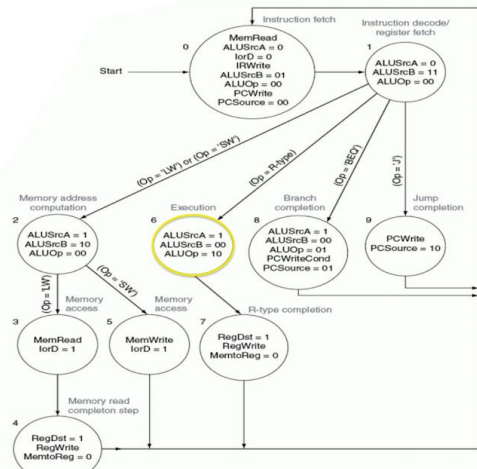
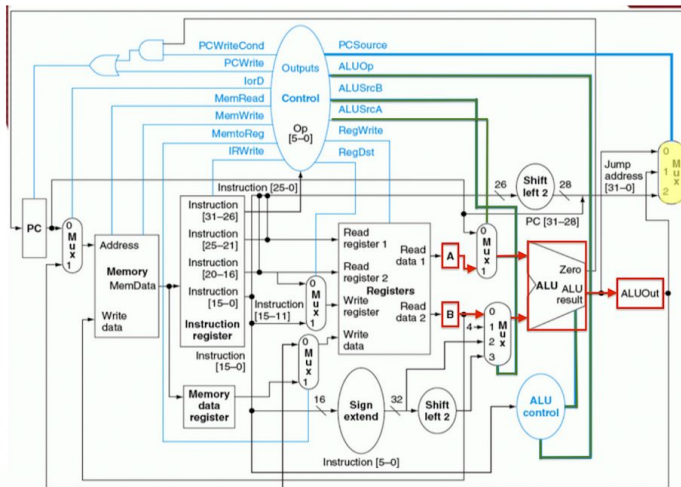
JAL - Jump and Link

1. Fetch.
2. Decode.
3. Jal Completion (estado extra, similar ao Jump completion, mas com escrita em registrador):

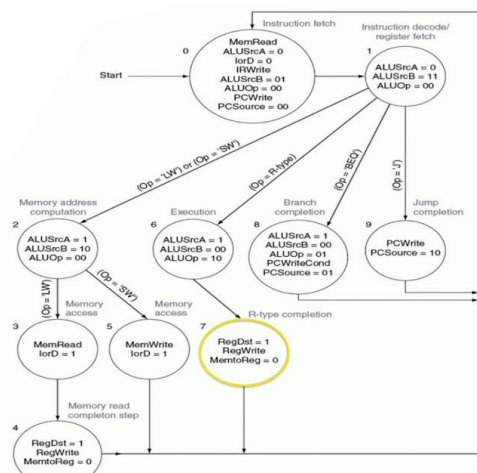
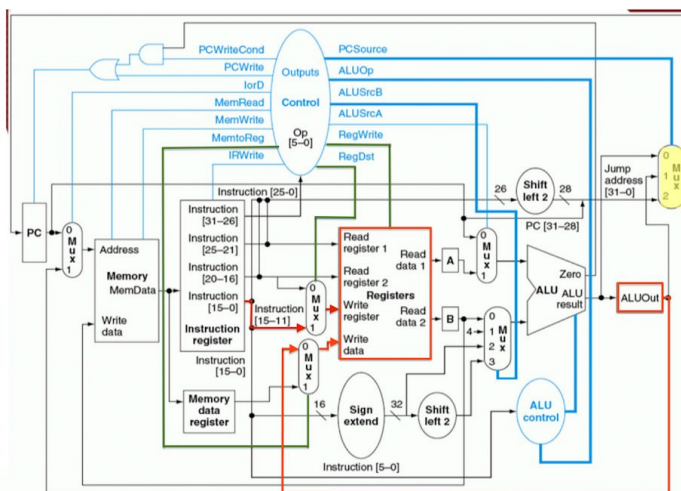


ADD

1. Fetch.
2. Decode.
3. Execution:



4. R-type Completion:



Instrução ADD. Explique a diferença da execução da instrução ADD para as instruções SUB, AND e OR.

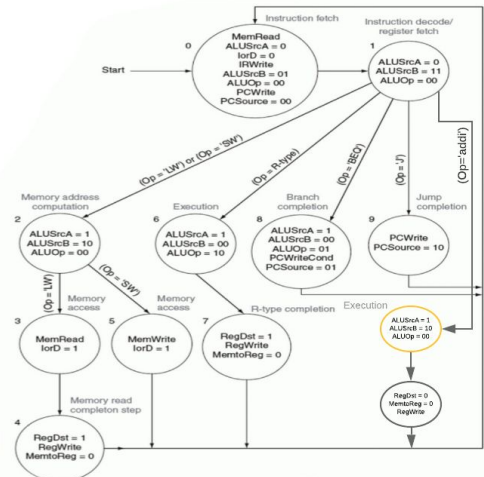
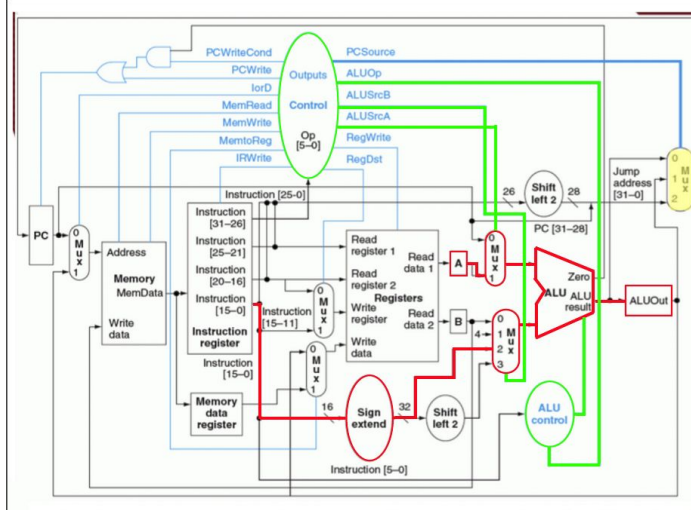
O ADD e o SUB são operações que podem gerar overflow, enquanto o AND e o OR são operações bitwise (cujo resultado de um determinado bit não altera o resultado de outro).

Entre si, cada uma destas quatro operações possui um código diferente (ALU Control Input), que realiza uma operação diferente, mas provém do mesmo ALUOp (10).

ADDI

1. Fetch.

3. Execution:



4. Write-back (Registra o resultado da soma):

