

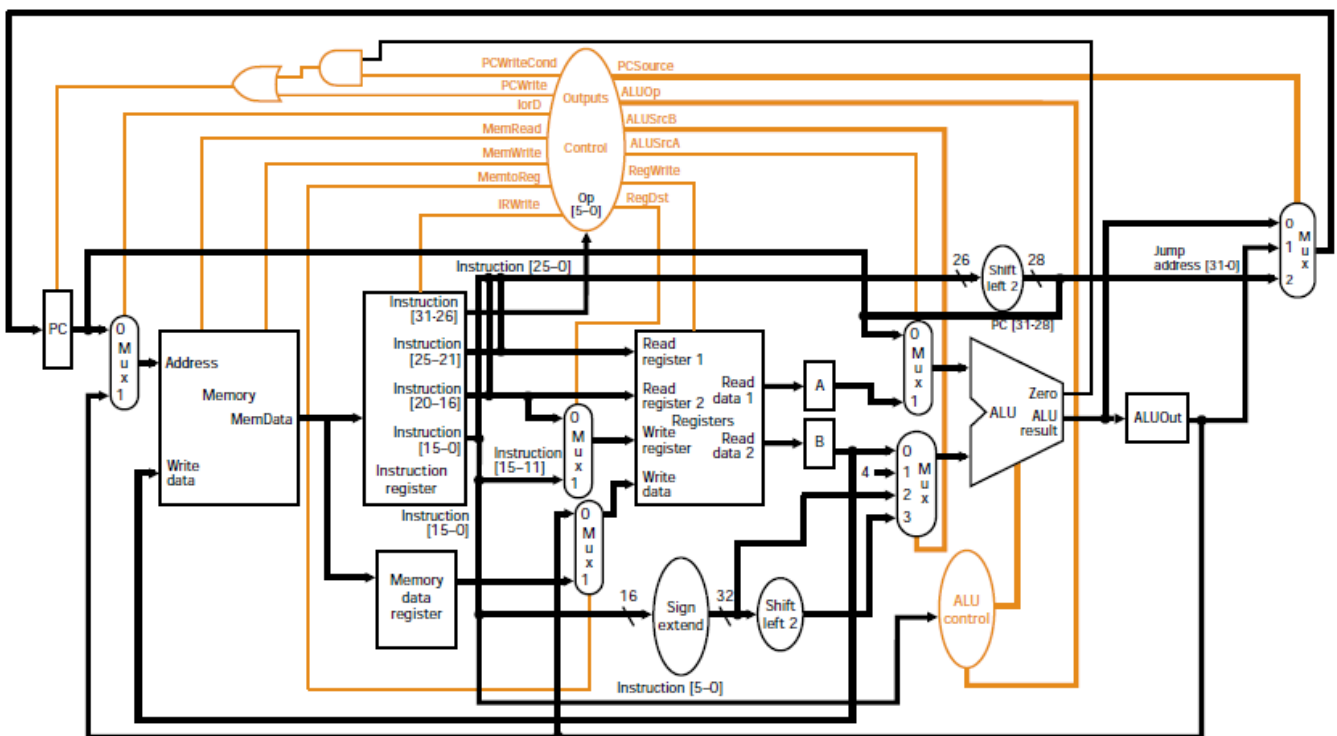
Desenvolva uma aplicação concorrente em C/PThreads no Linux, a qual simule o comportamento da CPU MIPS multiciclo [1]. Esta CPU deve executar estas nove instruções MIPS: *add*, *sub*, *and*, *or*, *slt* (*set-on-less-than*), *lw* (*load word*), *sw* (*store word*), *beq* (*branch-on-equal* - desvio condicional) e *j* (*jump* – desvio incondicional) [1]. Use como base o caminho de dados completo (vide figura abaixo) e os sinais de controle da CPU MIPS multiciclo, ambos descritos em [1].

Os registradores *PC*, *IR*, *MDR*, *A*, *B*, *ALUOut* e do banco de registradores são variáveis globais em memória e são compartilhados entre as **threads**. A memória RAM também é global.

As unidades funcionais executam em todos os ciclos. Cabe apenas aos sinais de controle emitidos pela UC impedir que algo “errado” seja feito e/ou considerado. A geração dos sinais de controle pela UC é o primeiro ponto que requer sincronização após o ciclo ter começado. Outros pontos de sincronização ocorrem em algumas das entradas dos multiplexadores. Em outras palavras, observe o fluxo de execução do ciclo de instrução para garantir que as dependências serão respeitadas, caso contrário não será atingida a semântica correta da CPU simulada. O início de um novo ciclo só pode ocorrer após se atingir toda a funcionalidade esperada para o ciclo atual.

O trabalho será entregue via *moodle*, um por grupo. **IMPORTANTE:** destaque como comentário no início do código o número do grupo e os nomes dos integrantes do grupo que de fato participaram do desenvolvimento do trabalho. Os nomes que não aparecerem nessa relação ficarão com nota "zero". A correção considerará a funcionalidade e a qualidade do programa desenvolvido.

Diagrama de blocos da CPU MIPS multiciclo a ser considerada neste trabalho prático [1].



[1] Patterson, D.A., Hennessy, J.L. Computer organization and design : the hardware/software interface, 3<sup>a</sup> ed., Elsevier, Amsterdam, 2005.