

Organização de Computadores I

DCC006

Professor: Omar Paranaíba Vilela Neto

Lista de Exercício 3

1 – Desejamos incluir uma variante da instrução ld (load double), que incrementa o registrador de índice após ler da memória. Essa instrução (l_inc) corresponde a estas duas instruções:

```
lw $rs, L($rt)
addi $rt, $rt, 1
```

2 – Descreva o efeito de que teria uma falha “stuck-at-0” (ou seja, independente do que deveria ser, o sinal é sempre 0) para os sinais mostrados a seguir, no caminho de dados multiciclo da Figura 5.27. Considere as seguintes falhas:

- a – EscreveReg = 0;
- b – LeMem = 0;
- c – EscreveMem = 0;
- d – EscreveIR = 0;
- e – EscrevePC = 0;
- f – EscrevePCCond = 0.

3 – Este exercício é semelhante ao exercício 2, mas, desta vez, considere falhas “stuck-at-1” (o sinal é sempre 1).

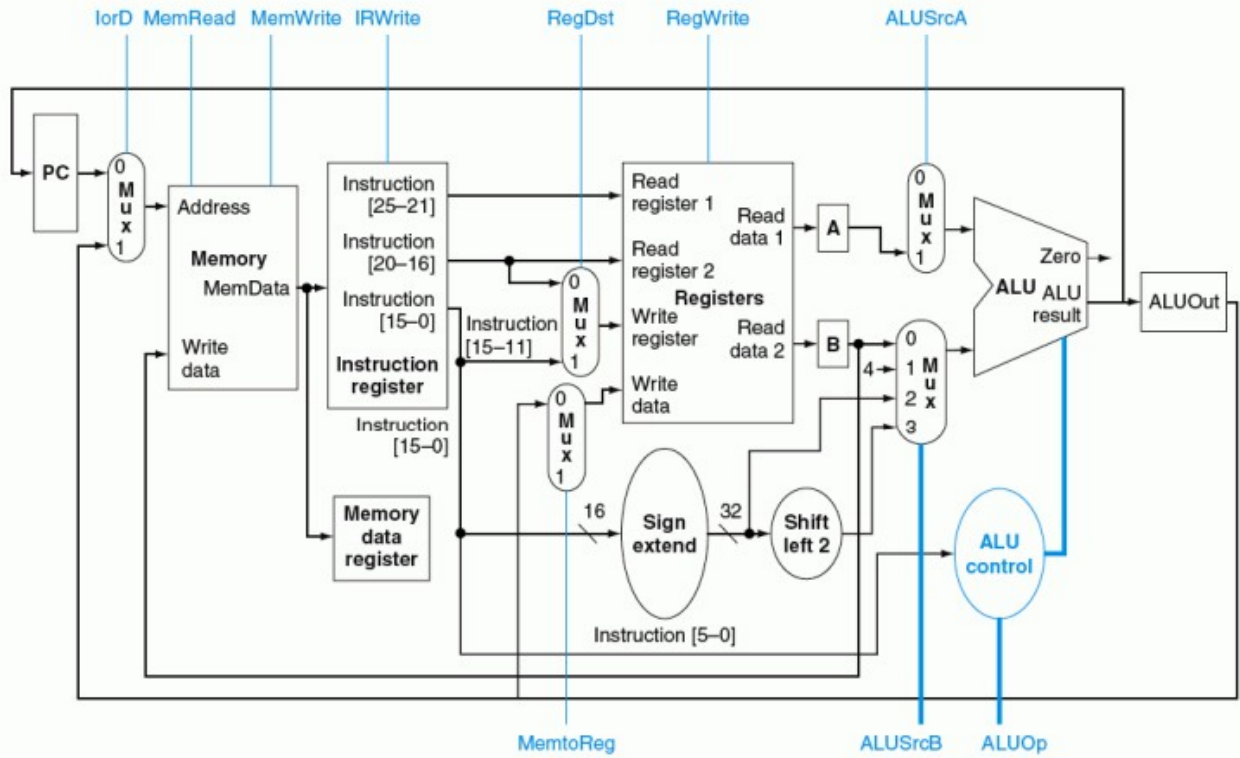
4 – Queremos acrescentar a instrução lui (load upper immediate). Use a mesma arquitetura do caminho de dados multiciclo da figura 5.28. Você pode achar útil examinar as etapas de execução e considerar as etapas que precisarão ser realizadas para executar a nova instrução. Quantos ciclos são necessários para implementar a instrução?

```
lui $S1, 100          -          Carrega constante nos 16 bits mais altos
```

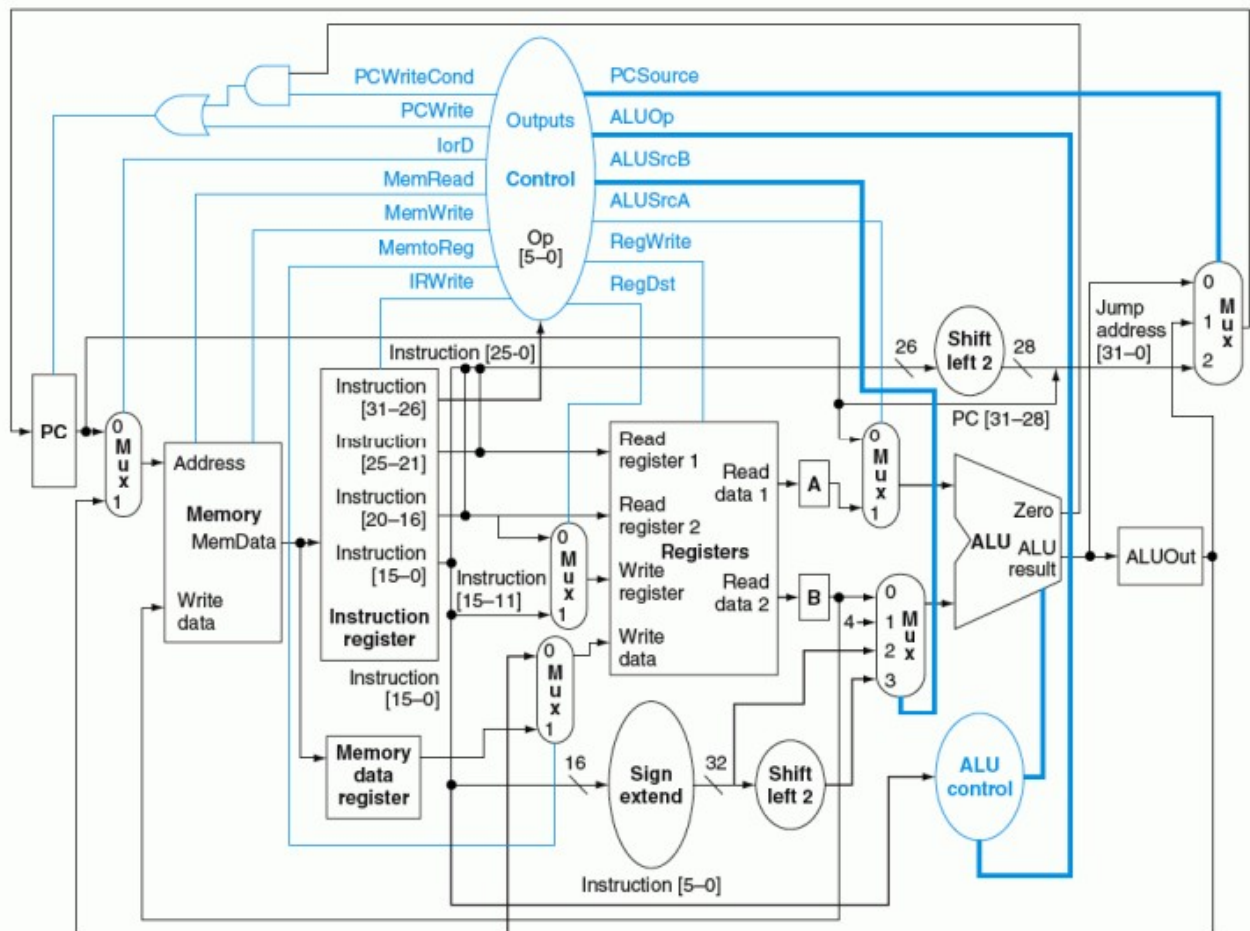
5 – Você foi solicitado a modificar a implementação do lui no exemplo anterior para reduzir o tempo de execução em 1 ciclo. Inclua quaisquer caminhos de dados e sinais de controle necessários no caminho de dados multiciclo da figura 5.28. Você precisa manter a suposição de que não sabe qual é a instrução antes do final do estado 1 (fim do segundo ciclo). Diga explicitamente quantos ciclos são necessários para executar a nova instrução em seu caminho de dados e desenhe a máquina de estados finitos modificada.

6 – Desenhe o controle da máquina e informe sobre a sua implementação.

Figuras



5.27



5.28

