



Laboratório 3 **- CPU *femto*RISC-V UNICICLO -**

Objetivos:

- Implementar uma CPU Uniciclo compatível com a ISA RV32I no Software de Simulação Deeds;
- Analisar o desempenho do processador construído;

A partir dos conceitos e diagramas apresentados em aula construa um processador *femto*RISC-V com a ISA RV32I usando o software de simulação Deeds, que seja capaz de executar a ISA reduzida composta pelas instruções: `add`, `sub`, `and`, `or`, `slt`, `xor`, `lw`, `sw`, `addi`, `slli`, `lui`, `jal`, `beq` e `jalr`.

1.1) Construa uma Unidade Lógico Aritmética, puramente combinacional de 32 bits, capaz de dar suporte a essas instruções. Defina o sinal de controle `ALUCtrl` de acordo com a tabela vista em aula e um sinal de saída 'zero'.

1.2) Construa as memórias de Instruções (ROM 1Ki x 32 bits), e de Dados (RAM Síncrona 1Ki x 32 bits) que possuam barramentos de 32 bits e os sinais de controle `EscreveMem` e `LeMem`. Considere que ambas as memórias começam no endereço 0x00000000.

Note que a Memória de Dados não é inicializável nesta versão do Deeds 😞

1.3) Construa um banco de registradores de 32 registradores de 32 bits cada de acordo com os requerimentos vistos em aula. Defina o registrador `sp` com o valor default 0x000003FC (último endereço da memória de dados);

1.4) Construa o caminho de dados completo e identifique os sinais de controle. Defina a tabela verdade do Bloco de Controle e projete um circuito que o implemente;

1.5) Construa o processador *femto*RISC-V, incluindo uma entrada de clock e um sinal de reset que deve resetar os valores dos registrados do BR e voltar `PC=0x00000000`. Qual a máxima frequência utilizável no seu processador?

1.6) Adicione ao seu processador os registradores do CSR `cycle`, `time` e `instret`, apenas como registradores de monitoramento (não acessíveis ao processador).

1.7) Implemente um circuito onde vc escolhe 2 registradores (por 5+5 chaves) e seus valores sejam apresentados em 2 conjuntos de 8 displays de 7 segmentos. Mostre também o registrador PC, a instrução lida da memória e os registradores `cycle`, `timer` e `instret` em displays;

1.8) Escreva um programa `TestBech.s` que verifique se todas as instruções foram implementadas corretamente, e filme a sua execução.

Dica: Considere se o programa não detectar nenhum erro mostre ao final da execução 0xCCCCCCCC no registrador `a0` e caso ocorra qualquer erro mostre 0xEEEEEEEE.

1.9) Faça a simulação e forma de onda na maior frequência possível e filme a execução do programa `Teste.s` no seu processador. Qual o tempo necessário à sua execução? Verifique se $t_{exec} = I \times CPI \times T$ justifique.

No arquivo GrupoX_Lab3.zip a ser enviado no Moodle coloque:

- (i) o arquivo GrupoX_Lab3.pdf do relatório;
- (ii) os arquivos do processador e com o programa Teste.s carregado na ROM;