



Laboratório 3 **- CPU μ 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 μ RISC-V com a ISA RV32I usando o software de simulação Deeds, que seja capaz de executar a ISA RV32I composta pelas instruções: `add`, `sub`, `and`, `or`, `xor`, `slt`, `sltu`, `lw`, `sw`, `addi`, `andi`, `ori`, `xori`, `slti`, `sltiu`, `sll`, `slli`, `lui`, `auipc`, `beq`, `bne`, `bge`, `bgeu`, `blt`, `bltu`, `jal` e `jalr`.

1.1) (1.0) Construa uma Unidade Lógica Aritmética (ULA), 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.

1.2) (1.0) 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 a memória de dados começa no endereço `0x10010000` e a memória de programa em `0x00400000`.

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

Dica2: <https://github.com/pinguimdeasadelta/montador>

1.3) (1.0) 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 `0x100103FC` (último endereço da memória de dados).

1.4) (1.0) 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) (1.0) Construa o processador μ RISC-V, incluindo uma entrada de clock e um sinal de reset que deve resetar os valores dos registradores do BR e voltar `PC=0x00400000`. Qual a máxima frequência utilizável no seu processador?

1.6) (1.0) Adicione ao seu processador os registradores do CSR `cycle`, `time` e `instret`, apenas como registradores de monitoramento (não acessíveis ao processador). Dica: Criem um controle do clock desses contadores por uma condição Ex.: Instrução = FIM: j FIM `0x0000006f`

1.7) (1.0) 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) (1.0) 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` e o endereço da instrução que o causou em `a1`.

1.9) (2.0) Faça a simulação e forma de onda na maior frequência possível e filme a execução do programa (com uma frequência menor) `Teste2.s` no seu processador. 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 Teste2.s carregado na ROM;