



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

Objetivos:

- Treinar o aluno com a Linguagem de Descrição de Hardware (HDL) Verilog;
- Familiarizar o aluno com a plataforma de desenvolvimento FPGA DE1-SoC da Intel e o software QUARTUS Prime;
- Desenvolver a capacidade de análise e síntese de sistemas digitais usando uma HDL;
- Implementar uma CPU Uniciclo compatível com a ISA RV32IMF com chamadas às rotinas do sistema;

PARTE A: Apresentação do ambiente de desenvolvimento, ferramentas e interface do processador

1) (0.0) Apresentação da plataforma de desenvolvimento.

Faça o download do arquivo Lab3.zip. Abra e leia o arquivo doc/RISCV-v2.0.docx.

Abra o projeto do processador RISCV-v2.0 e carregue o arquivo TopDE.sof no kit DE1-SoC.

1.1)(0.0) Com as chaves SW[9:0]=10'b0000000000, execute o programa *default* já presente na memória do processador.

Com as chaves SW[9:0]=10'b0000000010, execute novamente o programa default. O que ocorreu?

1.2)(0.0) Carregue o programa focafofa.s (Doc/testes) no Ras12_Custom3, analise e execute. Gere os arquivos .mif e carregue-os na DE1-SoC usando no menu Tools/In-system Memory Content Editor e importando os respectivos arquivos. Execute em uma frequência baixa (slow), visualizando os registradores, e em passo a passo. Defina um breakpoint no endereço 0x0040001c e execute novamente o programa no FPGA.

1.3)(0.0) Compile o TopDE.V e faça a simulação por forma de onda do processador executando o programa focafofa.s, usando o arquivo Waveform2.vwf, com o divisor de frequências definido como 12 e 1. Dica: Para alterar o programa a ser simulado por forma de onda, basta copiar os arquivos mif para o diretório Core com os nomes de1_data.mif e de1_text.mif e executar no menu Processing/Update Memory Initialization File.

1.4)(1.0) Modificando as seleções no arquivo Parametros.v e recompilando o processador, faça uma tabela comparativa dos requerimentos físicos e temporais das CPUs UNICICLO com as ISAs RV32I, RV32IM e RV32IMF.

Requerimentos físicos:	Requerimentos temporais:
Número de ALMs	maior atraso tpd
Número de Registradores	maiores tempos th, tco, tsu
Quantidade de bits de memória	máxima frequência de clock utilizável
Número de DSP	Requerimentos não atendidos (slacks)

Comente os resultados obtidos.

PARTE B: Processador RISC-V Uniciclo

2) (2.0) Seguindo o Cap. 10 do livro: [Guia Prático RISC-V: Atlas de Uma Arquitetura Aberta](#) Acrescente ao processador o banco de registradores CSR (Control and Status Registers) definidos no RARS: `ustatus(0)`, `fflags(1)`, `frm(2)`, `fcsr(3)`, `uie(4)`, `utvect(5)`, `uscratch(64)`, `uepc(65)`, `ucause(66)`, `utval(67)`, `uip(68)`, de forma a dar suporte à detecção de exceções cujos códigos (UCAUSE) são:


- 0 : Instruction address misaligned – endereço da instrução desalinhado , UVAL=endereço
- 1 : Instruction access fault - endereço fora do segmento .text, UVAL=endereço
- 2 : Illegal Instruction – Instrução não reconhecida, UVAL=instrução
- 4 : Load address misaligned – endereço de load desalinhado (obs.: lw, lh, lhu), UVAL=endereço
- 5 : Load access fault – endereço fora do segmento .data , UVAL=endereço
- 6 : Store address misaligned – endereço de store desalinhado (obs.: sw, sh) , UVAL=endereço
- 7 : Store access fault – endereço fora do segmento .data , UVAL=endereço
- 8 : environment call – chamada de ecall

Para todos: UEPC=PC e PC=UTVECT

Indique as modificações necessárias ao caminho de dados e no bloco de controle para que as exceções listadas sejam corretamente detectadas.

O tratamento das exceções deve ser feito no SYSTEMv14.s como uma mensagem de erro na “tela azul da morte” seguido do travamento do processador.

Escreva os programas que demonstrem a correta detecção e tratamento das exceções.



```
Error: 4 Load address misaligned
PC: 0x00400028
```

3) (3.0) Dado o processador Uniciclo RISC-V v2.0 ISA RV32IMF, acrescente as instruções listadas abaixo em conformidade com o RARS e a ISA RISC-V

- 3.1) (1.0) Descreva o projeto realizado, indicando as alterações e inclusões realizadas;
- 3.2) (1.0) Desenhe o Caminho de Dados completo (conforme diagrama visto em aula);
- 3.3) (0.5) Escreva a tabela verdade completa do Bloco de Controle;
- 3.4) (0.5) Analise e comente as dificuldades técnicas enfrentadas e as soluções propostas.

- 1) `csrrc t0, 0xFF, t1` Read/Clear CSR: read from the CSR into t0 and clear bits of the CSR according to t1
- 2) `csrrci t0, 0xFF, 10` Read/Clear CSR Immediate: read from the CSR into t0 and clear bits of the CSR according to a constant
- 3) `csrrs t0, 0xFF, t1` Read/Set CSR: read from the CSR into t0 and logical or t1 into the CSR
- 4) `csrrsi t0, 0xFF, 10` Read/Set CSR Immediate: read from the CSR into t0 and logical or a constant into the CSR
- 5) `csrrw t0, 0xFF, t1` Read/Write CSR: read from the CSR into t0 and write t1 into the CSR
- 6) `csrrwi t0, 0xFF, 10` Read/Write CSR Immediate: read from the CSR into t0 and write a constant into the CSR
- 7) `ebreak` Pause execution (dica: use a Break_Interface já implementada para parar o clock)
- 8) `ecall` Issue a system call : UCAUSE=8, UEPC=PC, PC=UTVECT
- 9) `uret` Return from handling an interrupt or exception: PC=UEPC

4) (1.0) Crie um programa `testbech.s` que verifique a corretude de cada uma das instruções implementadas. Analise a execução deste programa através da simulação por forma de onda. Filme a execução do seu programa no Rars e na DE1SoC. Analise os resultados obtidos e comente as dificuldades enfrentadas.

5) (1.0) Atualize o programa `testeECALLv13.s` e o *exception handler* `SYSTEMv13.s`, criando as versões v14 que não usem as macros `M_SetEcall`, `M_Ecall` e `M_Uret`. Filme o funcionamento no seu processador RISC-V Uniciclo RV32IMF na DE1-SoC comprovando seu correto funcionamento. Analise os resultados obtidos e comente as dificuldades enfrentadas.

6) (1.0) Execute o seu programa do Entregador de Pizza desenvolvido no Laboratório 1 no seu processador, com $N=10$ e filme a execução na DE1-SoC. Qual foi o mais rápido (t_{exec}), na DE1SoC ou no Rars? Por quê?

7) (1.0) Inclua no projeto um módulo de controle dos conversores Analógicos-Digitais (ADC) através da criação do IP do Quartus “ADC Controller for DE-series Boards”. Escreva a interface com o processador de acordo com o MMIO indicado na documentação.

```
// ***** Analog-Digital Converter Interface *****
ADC_Interface ADCI0 (
    .iCLK_50(oCLK_50),
    .iCLK(CLK),
    .Reset(Reset),
    .ADC_CS_N(ADC_CS_N),
    .ADC_DIN(ADC_DIN),
    .ADC_DOUT(ADC_DOUT),
    .ADC_SCLK(ADC_SCLK),
    // Barramento
    .wReadEnable(DReadEnable),
    .wWriteEnable(DWriteEnable),
    .wByteEnable(DByteEnable),
    .wAddress(DAddress),
    .wWriteData(DWriteData),
    .wReadData(DReadData)
);
```



Faça um programa que movimente um pixel na tela com velocidade e direção dadas a partir da leitura (x,y) do joystick analógico. Filme e coloque o link no relatório.

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

- (i) o arquivo GrupoX_Lab3.pdf do relatório;
- (ii) o arquivo TopDE.sof (com o programa do item 6 como default);
- (iii) o arquivo Core.qar do projeto realizado;
- (iv) os arquivos testbench.s, testeECALLv14.s e SYSTEMv14.s.