



Laboratório 4 **- CPU RISC-V MULTICICLO -**

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 Multiciclo compatível com a ISA RV32IMF com detecção de exceções;

1) (1.0) Modificando as seleções no arquivo Parametros.v e recompilando o processador, faça uma tabela comparativa dos requerimentos físicos e temporais das CPUs MULTICICLO 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.

2) (3.0) Seguindo o Cap. 10 do livro: [Guia Prático RISC-V: Atlas de Uma Arquitetura Aberta](#) Acrescente ao processador MULTICICLO 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 e do segmento de MMIO, UVAL=endereço
- 6 : Store address misaligned – endereço de store desalinhado (obs.: sw, sh) , UVAL=endereço
- 7 : Store access fault – endereço fora do segmento .data e do segmento de MMIO, UVAL=endereço
- 8 : environment call – chamada de ecall UVAL=0

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 já deve estar feito no SYSTEMv14.s elaborado no Lab. 3, 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 Multiciclo 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) (0.5) Desenhe o Caminho de Dados completo (conforme diagrama visto em aula);

3.3) (1.0) Desenhe a Máquina de Estados 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, UCAUSE=0 |

4) (1.0) Use o seu programa testbech.s do Lab 3 para verificar 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) Execute o seu programa testeECALLv14.s juntamente com seu *exception handler* SYSTEMv14.s, criados no Lab. 3. Filme o funcionamento no seu processador RISC-V Multiciclo RV32IMF na DE1-SoC comprovando seu correto funcionamento. Analise os resultados obtidos e comente as dificuldades enfrentadas.

6) (0.5) Execute o seu programa do Entregador de Pizza desenvolvido no Laboratório 1 no seu processador multiciclo, 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) (0.5) Execute o seu programa que movimenta 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_Lab4.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;