



Laboratório 5

- CPU RISC-V PIPELINE -

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 Pipeline compatível com a ISA RV32I reduzida;

1) (0.0) Faça o download do arquivo Lab3.zip . Restaure o projeto do processador RISC-V v24, o programa focafoca.s deve estar como default (de1_data.mif e de_text.mif), ajuste o Parametros.v para PIPELINE RV32IMF, compile e realize a simulação com a Wave1_Pipett.vwf.

1.1) (0.0) Carregue na DE1 o processador compilado.

Com as chaves SW[9:0]=10'b1000000100, execute o programa. Explique.

Com as chaves SW[9:0]=10'b1000000010, execute novamente o programa. Explique.

Com as chaves SW[9:0]=10'b1000000001, execute novamente o programa. Explique.

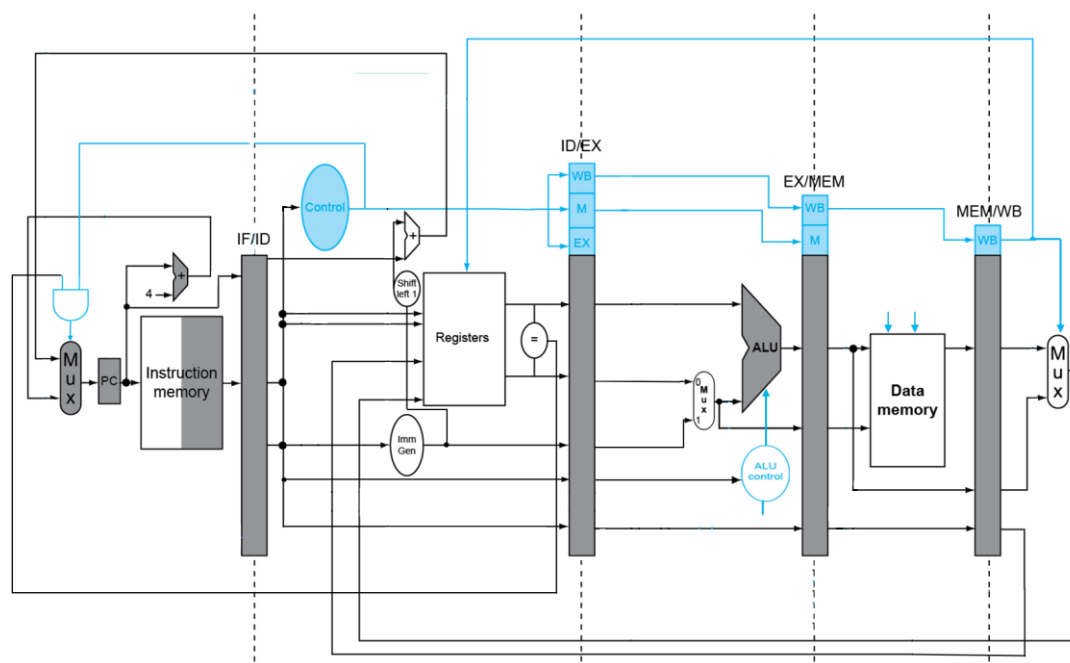
1.2) (2.0) Modificando as seleções no arquivo Parametros.v e recompilando o processador, faça duas tabelas comparativas dos requerimentos físicos e temporais das CPUs PIPELINE com as ISAs RV32I, RV32IM, RV32IMF e a Reduzida vista em aula RV32Red, gerando os arquivos Pipeline_RV32I.sof, Pipeline_RV32IM.sof, Pipeline_RV32IMF.sof e Pipeline_RV32IRed.sof.

ISA	Número de ALMs	Número de Registradores	Quantidade de bits de memória	Número de DSPs
RV32I				
RV32IM				
RV32IMF				
RV32Red				

ISA	maior atraso tpd	maiores tempos tco	maiores tempos th	maiores tempos tsu	máxima frequência de clock utilizável
RV32I					
RV32IM					
RV32IMF					
RV32Red					

Comente os resultados obtidos.

2) (8.0) Implemente o processador Pipeline com ISA reduzida apresentado em aula (ISA32Red), com as instruções: add, sub, and, or, slt, lw, sw, beq, jal, e ainda as instruções **jalr** e **addi**.



2.1) (3.0) Os blocos de Memória, Bloco de Controle, Banco de Registradores, Gerador de Imediatos, ULA, controlador da ULA são os mesmos do processador Uniciclo.

a) (1.0) Acrescente os registradores de pipeline

b) (2.0) Implemente o Processador Pipeline completo com os registradores de pipeline. Obtenha o netlist RTL view. E faça a simulação por forma de onda. Levante os requisitos funcionais e temporais do seu processador.

2.2) (1.0) Use o TopDE.v do Processador Pipeline:

- Use o fdiv.v com as chaves SW[3:0] para dividir o clock de 50MHz para o processador.
- De acordo com as chaves SW[8:4] mostre nos displays hexadecimais decoder7.v o valor do respectivo registrador. SW[9] define qual Banco de registradores é observado.
- Caso $\sim\text{KEY}[1] = 1$ mostra o valor de PC e caso $\sim\text{KEY}[2] = 1$ mostra a instrução (IR) e $\sim\text{KEY}[0] = 1$ faz o reset do processador (PC=0x00400000 e todos os registradores como 0, menos o sp e o gp).
- Adicione ao TopDE.v sinal que deixe piscando os LEDRs não usados.

2.3) (1.0) Com o mesmo programa testeRed.s, corrija os hazards apenas com a inserção de bolhas (nop), e teste a corretude de todas as 9+2 instruções, execute na maior frequência possível pelas divisões das SW[3:0], filme e explique a execução.

2.4)(1.5) Faça uma comparação dos requerimentos físicos e temporais dos seus 3 processadores (Uni, Multi e Pipe). Qual o tempo que cada um executa o programa testeRed.s ? Dica: Faça a simulação por forma de onda do TopDE para medir e colocar no relatório.

2.5)(1.5) Faça uma comparação dos requerimentos físicos e temporais dos 3 processadores RISCv-v24 (UNICICLO, MULTICICLO e PIPELINE). Carregue o programa testeECALLv24.s, visualize a tela de dados e veja qual o texec de cada processador. Tire fotos dessas telas para colocar no relatório.

No arquivo GrupoX_Lab5.zip a ser enviado no Moodle coloque apenas:

- o arquivo GrupoX_Lab5.pdf do relatório;
- o arquivo Pipeline_RV32Red.qar, com o programa do item 2.3) como default nos arquivos de1_data.mif e de1_text.mif ;