

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMATICA
Trabalho 1 - Sistemas Digitais
Prof. Fernanda Kastensmidt

TRABALHO Processador Neander
Pontuação: 10 pontos (vale 1/4 da nota do semestre)

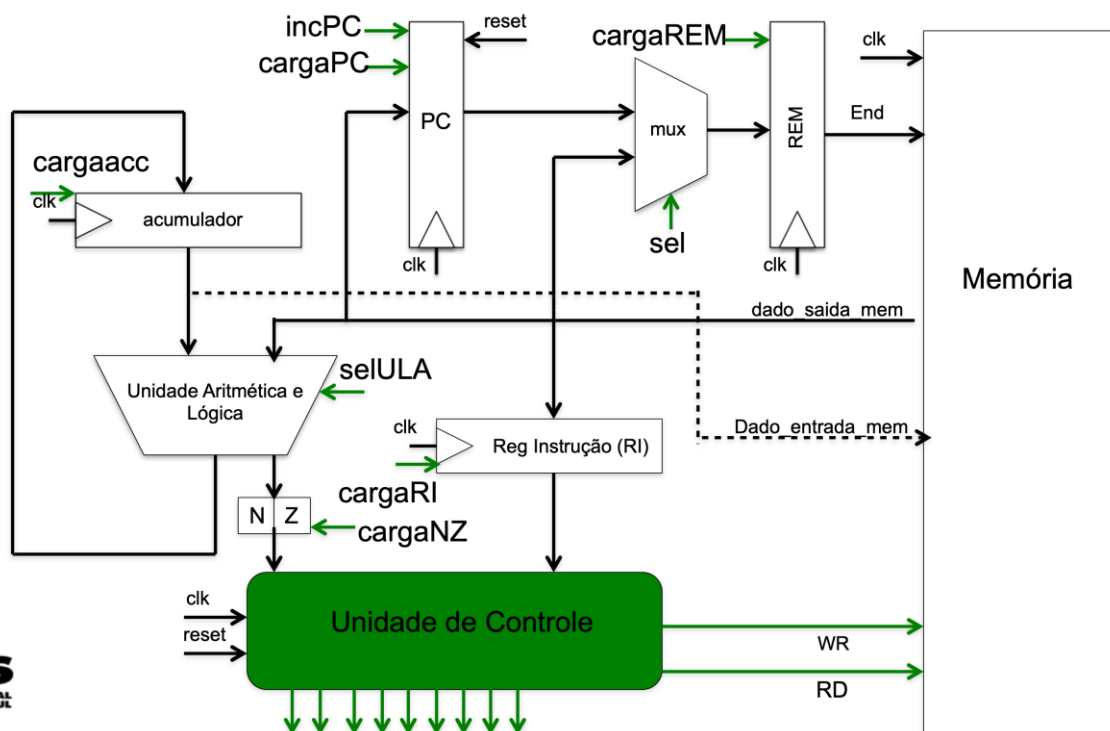
Nome: _____ matricula: _____

Objetivo: projetar e descrever em VHDL o processador Neander, implementar 2 programas em sua memória e mostrar através de simulação lógica sem e com atraso o funcionamento.

Pontuação extra: programar na placa de prototipação o Neander e mostrar funcionando com os displays 7 segmentos.

PASSO 1: 3 pontos

Descrever o DATAPATH do processador Neander em VHDL em uma entidade apenas chamada de datapath_neander.



Cole aqui o código completo em VHDL do datapath

Qual componente FPGA escolheste para a síntese? _____

Quantos registradores tem o datapath do Neander? _____

Quantas operações diferentes tem a ULA? _____

A área do DATAPATH em # LUTs: _____ e #ffps: _____

PASSO 2: 3 pontos

Descrever a parte de controle do Neander em VHDL como uma máquina de estados usando 2 PROCESS, um process(clk, rst) e outro process(estados, entradas).

Dada as tabelas com as instruções do Neander por estado da máquina de estados

tempo	STA	LDA	ADD	OR	AND	NOT
t0	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM
t1	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC
t2	carga RI	carga RI	carga RI	carga RI	carga RI	carga RI
t3	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	UAL(NOT), carga AC, carga NZ, goto t0
t4	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	
t5	sel=1, carga REM	sel=1, carga REM	sel=1, carga REM	sel=1, carga REM	sel=1, carga REM	
t6	carga RDM	Read	Read	Read	Read	
t7	Write, goto t0	UAL(Y), carga AC, carga NZ, goto t0	UAL(ADD), carga AC, carga NZ, goto t0	UAL(OR), carga AC, carga NZ, goto t0	UAL(AND), carga AC, carga NZ, goto t0	

tempo	JMP	JN, N=1	JN, N=0	JZ, Z=1	JZ, Z=0	NOP	HLT
t0	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM
t1	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC
t2	carga RI	carga RI	carga RI	carga RI	carga RI	carga RI	carga RI
t3	sel=0, carga REM	sel=0, carga REM	incrementa PC, goto t0	sel=0, carga REM	incrementa PC, goto t0	goto t0	Halt
t4	Read	Read		Read			
t5	carga PC, goto t0	carga PC, goto t0		carga PC, goto t0			
t6							
t7							

Cole aqui o VHDL da parte de controle usando FSM com dois process.

Entity

....

Begin

Process (reset, clock)

Begin

End process;

process(estado, sinais)

Begin

....

CASE is

END CASE;

End process;

end;

PASSO 3: 1 ponto

Descrever o programa em Assembly do Neander que realize a multiplicação de dois números inteiros positivos de 8 bits por soma sucessiva e colocar no arquivo .COE na memória BRAM.

Inserir aqui o programa em Assembly com explicação.

Inserir aqui o .coe

PASSO 4: 3 pontos

Simular sem atraso o Neander com o programa teste a ser feito pelo aluno e depois que testado e funcionando, simular com o programa do passo 3. Depois de tudo funcionando, simular também com atraso.

Lembrem-se que deve ser feito um testbench para a simulação.

Colar aqui o programa teste e simulações (.JPG)

Colar aqui as simulações do programa do passo 3:

Dados do Neander completo:

A área do DATAPTH em # LUTs: _____ e #ffps: _____ e #
BRAM _____

Quantos ciclos de relógio foram necessários para a execução do programa de
multiplicação no Neander? _____ = # instrucoes x # cc por instrucao

Qual frequência de operação o Neander atingiu? _____ Como tu
fizeste o teste para saber que ele não consegue rodar mais rápido que essa
frequência?

_____.

PONTO EXTRA:

Link para video mostrando o funcionamento da placa de prototipação.

