UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DISCIPLINA: SISTEMAS DIGITAIS
Prof. Fernanda Lima Kastensmidt

2018-2

Projeto do Processador Neander em VHDL

O computador NEANDER foi criado com intenções didáticas pelo prof. Raul Weber da UFRGS. Neste site há referencias e link para o simulador: http://www.dcc.ufrj.br/~gabriel/neander.php

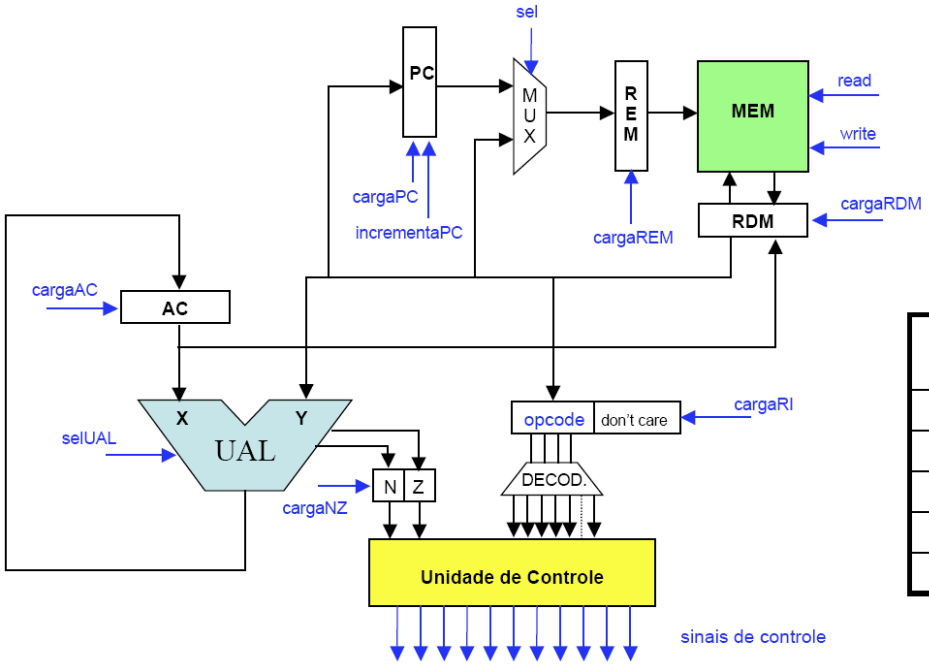
O objetivo deste trabalho de SD é implementar o NEANDER usando a linguagem de descrição de hardware VHDL, simular esse circuito em um simulador lógico (ISE) sem e com modelo de atraso.

Deve-se inserir a instrução de Subtração (SUB) conforme os modo de operandos da instrução ADD. E inserir mais uma instrução a definir.

Programas a serem implementado no NEANDER na memoria embarcada BRAM.

1) Soma de duas matrizes A e B 2x2 com dados de 4 bits, onde os dados das matrizes podem ser entrados pela interface da placa em endereços de memoria pre-determinados. A matriz resultado estará em endereços pré-determinados.
2) Multiplicação de duas matrizes A e B 2x2 com dados de 4 bits, onde os dados das matrizes podem ser entrados pela interface da placa em endereços de memoria pre-determinados. A matriz resultado estará em endereços pré-determinados.
3) Programa a ser definido pelo aluno que use as instruções de subtração e a nova instrução definida pelo aluno.

O computador NEANDER tem as seguintes características:
• Largura de dados e endereços de 8 bits
• Dados representados em complemento de dois
• 1 acumulador de 8 bits (AC)
• 1 apontador de programa de 8 bits (PC)
• 1 registrador de estado com 2 códigos de condição: negativo (N) e zero (Z)



operações da UAL	selUAL
X + Y	000
X and Y	001
X or Y	010
Not X	011
Y	100

Projeto do Datapath

Passo 1: Projeto dos circuitos combinacionais

A) Multiplexador 2:1 de largura de 8 bits.

B) Unidade Aritmética e Lógica (UAL): conforme a seleção da UAL (selUAL), 5 operações diferentes podem ocorrer na UAL. A largura dos dados é de 8 bits. Note que a UAL é capaz de identificar quando o resultado é ZERO (Z) ou NEGATIVO (N).

C) Decodificador de instruções: na tabela a seguir AC é o acumulador, MEM(end) significa conteúdo da posição end de memória, N e Z são os códigos de condição e ← representa uma atribuição.

Instrução	Comentário
NOP	nenhuma operação
STA end	MEM(end) ← AC
LDA end	AC← MEM(end)
ADD end	AC← MEM(end) + AC
OR end	AC← MEM(end) OR AC
AND end	AC← MEM(end) AND AC
NOT	AC← NOT AC
JMP end	PC← end
JN end	IF N=1 THEN PC ← end
JZ end	IF Z=1 THEN PC ← end

Código	Instrução	Comentário
0000	NOP	nenhuma operação
0001	STA end	armazena acumulador - (store)
0010	LDA end	carrega acumulador - (load)
0011	ADD end	soma
0100	OR end	"ou" lógico
0101	AND end	"e" lógico
0110	NOT	inverte (complementa) acumulador
1000	JMP end	desvio incondicional - (jump)
1001	JN end	desvio condicional - (jump on negative)
1010	JZ end	desvio condicional - (jump on zero)
1111	HLT	término de execução - (halt)

Passo 2: Projeto dos circuitos sequenciais

A) Registradores de 8-bits **ACC, REM, RDM e INST(opcode)** com carga paralela. Notem que todos esses registradores são iguais. Registrador **NZ** de 2 bits com carga paralela. *Onde N - (negativo) : indica sinal do resultado, 1 - resultado é negativo e 0 - resultado é positivo. Z - (zero) : indica resultado igual a zero, 1 - resultado é igual a zero e 0 - resultado é diferente de zero.*

B) Contador de 8-bits **PC** com carga paralela e sinal de incremento.

C) Memória **RAM** para programa e dados. USE BRAM dual port para usar o DUMP de memoria e visualizar o endereço de memoria e dado correspondente no vídeo e display de 7 segmentos na placa.

IMP: o endereço 0 da BRAM deve ter a instrução NOP. Logo a primeira instrução do programa estará no endereço 01 de BRAM.

Passo 3: Projeto da Unidade de Controle

A unidade de controle é uma maquina de estados finita (FSM) que controla a leitura e escrita da memoria e os elementos do Datapath conforme os sinais do decodificador de instrução e a temporização do processador.

Passo 4: Projeto do programa

A memória projetada ao ser inicializada com o arquivo .coe que contem o programa projetado.

ENTREGA E APRESENTAÇÃO:

- Apresentação oral com slides do tipo powerpoint/pdf/ou similiar onde cada aluno apresenta o trabalho, projeto, simulações, explica os programas e o testbench. Dados de área, desempenho em frequência e tempo de execução em ciclos de relógio e tempo em segundos deve ser apresentado dado um determinado clock usado. O programa que roda no Neander deve ser apresentado e os resultados esperados.
-

Sugestão:

Programa	Numero de Instruções Executadas	Tempo de execução em # de ciclos de relógio (c.c.)	Em Segundos (Neander operando a 50 MHz)
Soma de matrizes			
Multiplicação de matrizes			
A definir pelo grupo			
Outro ?			

Dados de Area do Neander
FPGA device: SPARTAN3E-100
Numero de 4-LUTs:
Numero de ffps:
Numero de BRAM:
Numero de MULT e ADD DSP

Máxima frequência de operação estimada pela ferramenta ISE:
Máxima frequência de operação simulada no simulador ISE (correto funcionamento):

- Upload no Moodle: Todos os códigos devem ser enviados ao link do Moodle para avaliação do professor (projeto e arquivo pdf da apresentação).

AVALIAÇÃO:

2 pontos: apresentação em slides (organização, clareza)
4 pontos: qualidade dos resultados apresentados (simulação **sem atraso**, dados de área em numero de 4-LUTs e flip-flops, desempenho em MHz)
2 pontos: novas instruções (SUB e outra) e funcionamento correto das mesmas
2 pontos: Simulação **com atraso** com explicação nos gráficos