

## **Projeto do Processador Neander em: VHDL**

O computador NEANDER foi criado com intenções didáticas pelo prof. Raul Weber da UFRGS. Neste site há referências 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 e implementar no FPGA da placa de prototipação fornecida em aula. O Neander deve utilizar as interfaces da placa tais como:

**- botões, chaves, leds, display 7 segmentos e saída de vídeo.**

**Deve-se inserir a instrução de multiplicação conforme os modo de operandos da instrução ADD. A parte menos significativa do resultado estará a disposição no registrador ACC e a parte alta em um endereço de memória de dados 250d (padrão).**

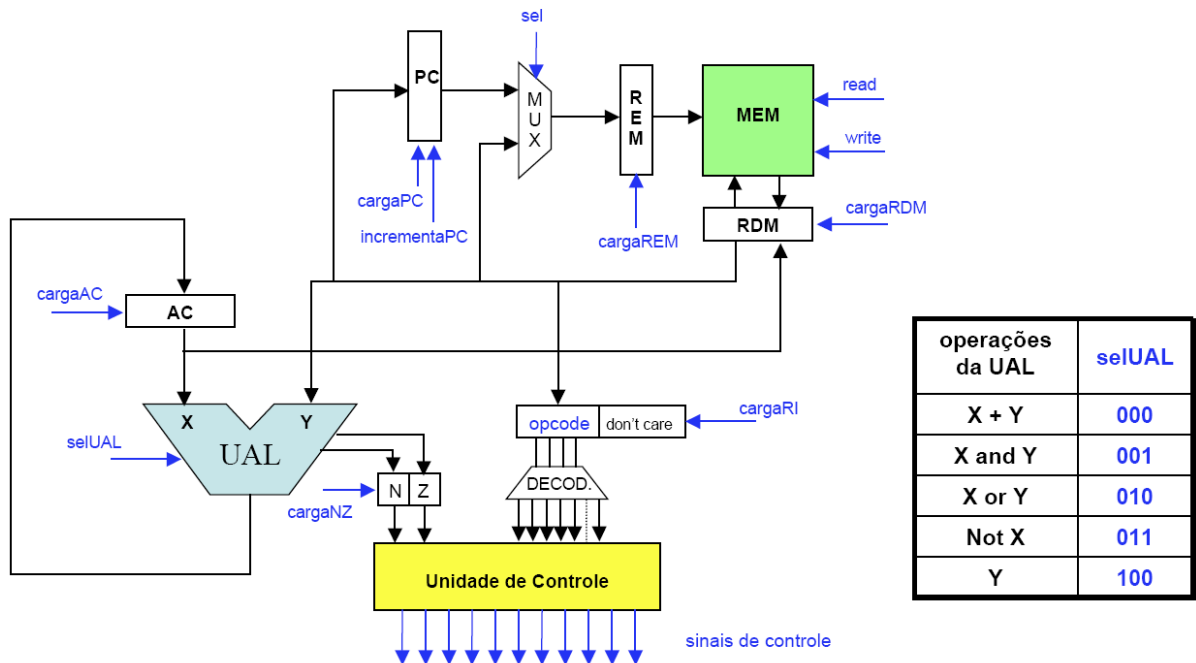
**Deve-se inserir mais uma instrução a definir pelo GRUPO.**

Programas a serem implementado no NEANDER na memória embarcada BRAM

- 1) contador de 0 a 36 que conta de 6 em 6 e vai mostrando os valores no display de 7 segmentos
- 2) 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 memória pre-determinados. A matriz resultado estará em endereços pré-determinados e podem ser mostrados nos visores 7 seg da placa ou no monitor de vídeo.
- 3) Programa a ser definido pelo grupo que use as instruções de multiplicação e a nova instrução definida pelo grupo.

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)



## 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	$MEM(end) \leftarrow AC$
LDA end	$AC \leftarrow MEM(end)$
ADD end	$AC \leftarrow MEM(end) + AC$
OR end	$AC \leftarrow MEM(end) \text{ OR } AC$
AND end	$AC \leftarrow MEM(end) \text{ AND } AC$
NOT	$AC \leftarrow \text{NOT } AC$
JMP end	$PC \leftarrow end$
JN end	IF N=1 THEN $PC \leftarrow end$
JZ end	IF Z=1 THEN $PC \leftarrow end$

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 memória e visualizar o endereço de memória e dado correspondente no vídeo e display de 7 segmentos na placa.

**IMP: o endereço 0 da BRAM deve ter a instrução NOP.**

## Passo 3: Projeto da Unidade de Controle

A unidade de controle é uma máquina de estados finita (FSM) que controla a leitura e escrita da memória 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 similar onde a dupla ou trio apresenta o trabalho, projeto, simulações, verificação, vídeo funcionando na placa e nas interfaces utilizadas. 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. **IMPORTANTE:** todos os integrantes do grupo devem estar presentes na apresentação do trabalho.

Sugestão:

Programa	Numero de Instruções Executadas	Tempo de execução em # de ciclos de relógio	Em Segundos (Neander operando a 50 MHz)
contador			
Soma 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

4 pontos: qualidade dos resultados apresentados (simulação sem atraso, simulação com atraso, dados de área em numero de 4-LUTs e flip-flops, desempenho em MHz)

2 pontos: novas instruções (multiplicação e outra) e funcionamento correto das mesmas

2 pontos: prototipação na placa com o seu correto funcionamento e uso das interfaces de vídeo e display 7 segmentos para debug.