

2020-2

Trabalho 2 - Individual

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.

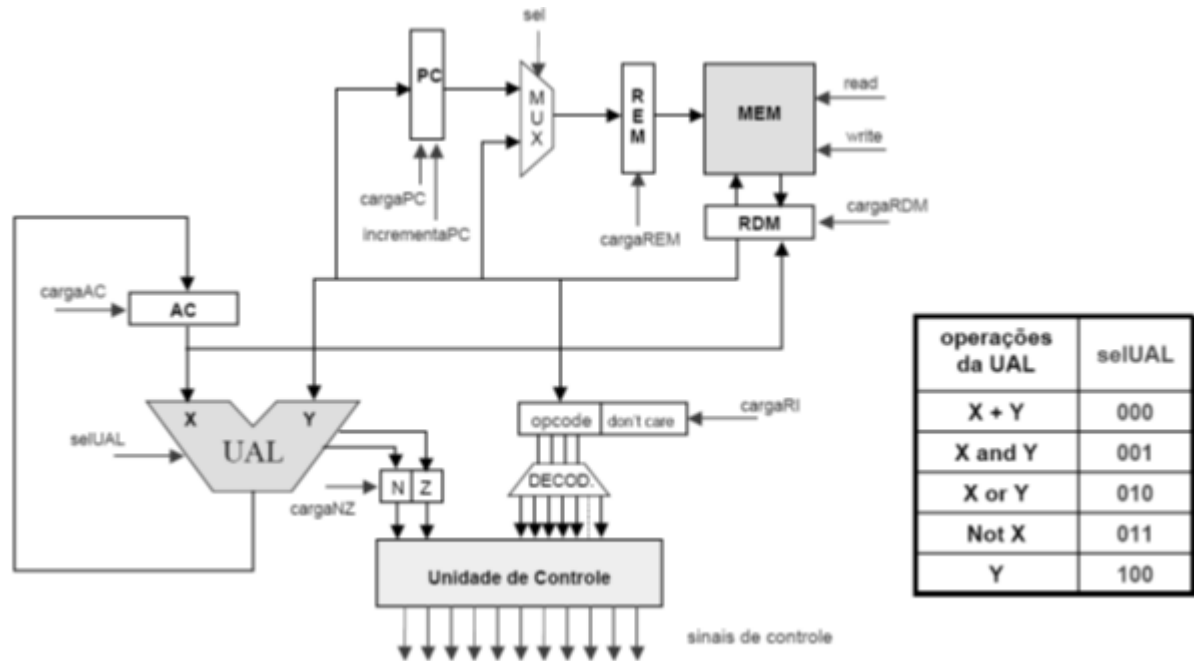
Deve-se inserir a instrução de Subtração (SUB) conforme os modos de operandos da instrução ADD e XOR conforme os modos de operandos da OR.

Programas a serem implementado no NEANDER na memória embarcada BRAM.

- 1) Soma **de duas matrizes A e B 2x2** com dados de 8 bits, onde os dados das matrizes estão armazenados em memória (use o .coe para inicializar a memória).
- 2) Programa a ser definido pelo aluno que use as instruções de soma e subtração (com no mínimo 10 instruções ao total).
- 3) Programa que calcule a **paridade par** de um número de 8 bits com portas lógica XOR e mascaras. Resultado da paridade (nono bit) deve ser armazenado no endereço 150d da memória.
- 4) Programa que use a instrução XOR (com no mínimo 10 instruções ao total).

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

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

- 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 single PORT e o crie o arquivo .coe para inicialização.

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 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:

- 1) Apresentação pelo MS-TEAMS dos slides:** com slides do tipo powerpoint/pdf/ou similar onde o aluno apresenta:
- 2) Descrição do trabalho, a seguir sugestão de tópicos que devem conter no relatório de entrega do trabalho**
 - Descrição do trabalho,
 - VHDL completo do neander
 - Explicação do testbench e VHDL completo
 - Explicação das aplicações em Assembly
 - Simulações sem e com atraso com detalhes e flechas mostrando inicio meio e final do programa e resultados,
 - Dados de área, 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			
Programa 2			
Programa 3			
Programa 4			

Dados de Area do Neander

FPGA device:

Numero de 4-LUTs:

Numero de ffps:

Numero de BRAM:

Numero de MULT e ADD DSP

- **Upload no MS_teams dos slides e do relatório**