

Projeto do Processador Neander em: VHDL

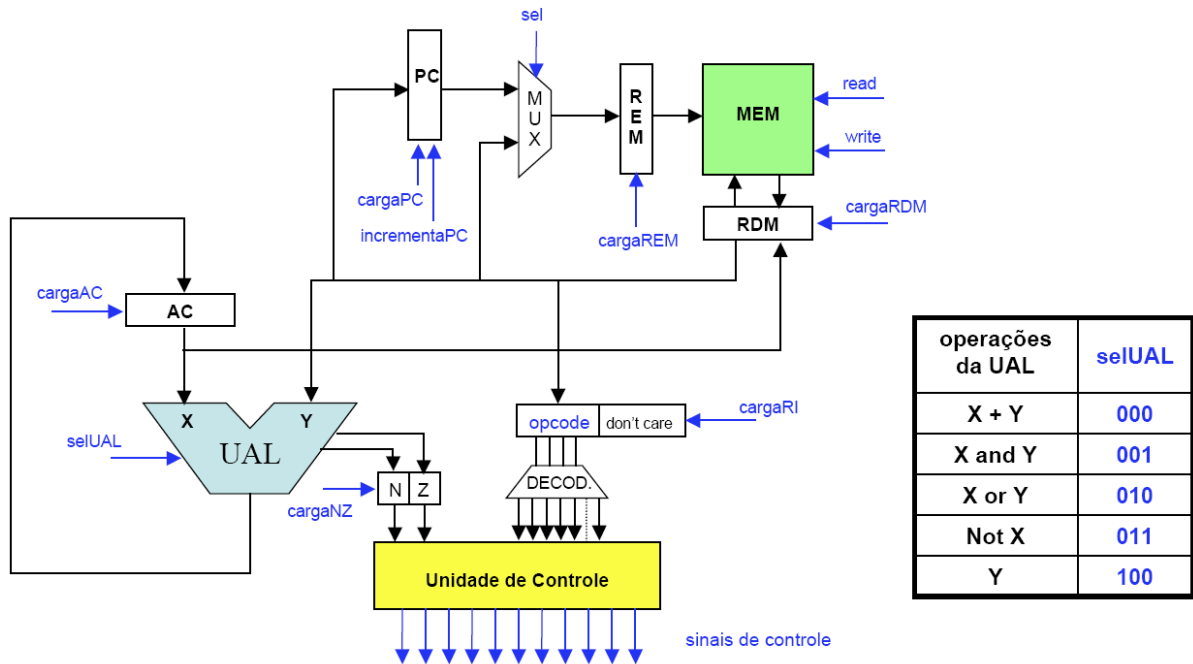
O computador NEANDER foi criado com intenções didáticas¹. O objetivo deste trabalho é implementar o NEANDER usando a linguagem de descrição de hardware VHDL, simular esse circuito em um simulador lógico sem e com atraso e implementar no FPGA da placa de prototipação fornecida em aula. Para verificar o funcionamento do circuito realizar um programa que utilize pelo menos todas as instruções do NEANDER.

Desafio: Insira as instruções de deslocamento de registrador (ACC) para esquerda e direita e 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 50d (padrão).

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)

¹ Esta pseudo-máquina foi criada pelos Profs. Raul Weber e Taisy Weber para a disciplina Arquitetura de Computadores I da UFRGS.



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 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.

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 .core que contem o programa projetado. Para futuras modificações no programa pode-se usar o arquivo MIF e altera-lo.

ENTREGA:

- Apresentação oral com slides do tipo powerpoint onde a dupla 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.

Todos os códigos devem ser enviados ao link do Moodle para avaliação do professor (projeto e arquivo pdf da apresentação).

AVALIAÇÃO:

Até:

2 pontos: apresentação

4 pontos: qualidade dos resultados apresentados (simulação sem atraso, com atraso, dados de área, desempenho)

2 pontos: novas intruções e funcionamento

2 pontos: prototipação na placa com o seu correto funcionamento