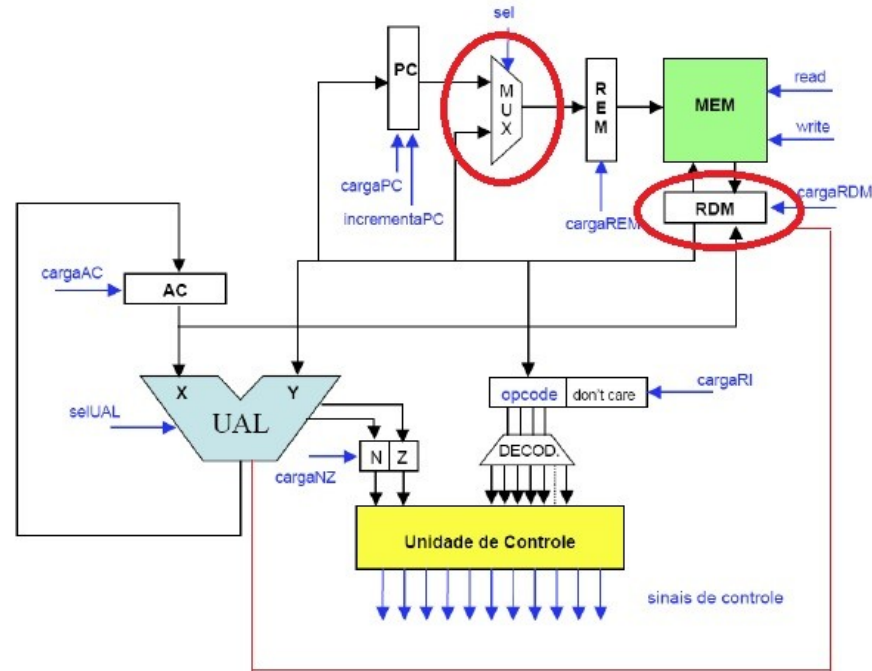


Neander

Mathaus Corrêa Huber

Diagrama



Descrição

- O computador hipotético NEANDER é uma máquina muito simples, desenvolvida para fins didáticos. Seu conjunto de instruções é limitado a NOP (no operation), STA (store), LDA(load), ADD (add), OR (logical or), AND (logica and), NOT (logical not), JMP (jump toaddress), JN (jump if negative), JZ (jump if zero) e HLT (halt).
- Sua arquitetura possui endereços e dados de 8 bits, com esses dados sendo representados em complemento de 2.
- O computador possui um acumulador de 8 bits (AC), um apontador de programa de 8 bits (PC), um registrador para endereços de memória de 8 bits (REM), um registrador para dados da memória de 8 bits (RDM), um registrador de instruções de 8 bits (RI) e um registrador de estado com 2 códigos de condição : negativo (N) e zero (Z).
- Todos os sinais de controles utilizados pela máquina são produzidos pela Unidade de Controle

Decodificador

Também temos uma saída que é um vetor de onze posições, onde cada posição correlaciona com uma variável da UC, sendo:

- 1a: NOP (nenhuma operação);
- 2a: STA (armazena acumulador);
- 3a: LDA (carrega o acumulador);
- 4a: ADD (soma);
- 5a: OR (ou lógico);
- 6a: AND (e lógico);
- 7a: NOT (inverte acumulador);
- 8a: JMP (desvio incondicional);
- 9a: JN (desvio condicional a negativo);
- 10a: JZ (desvio condicional ao zero);
- 11a: HLT (termina a execução).

Memória

A memória do processador abriga dados recebidos do bloco RDM. Os dados recebidos do bloco RDM ficam armazenados em uma célula específica dentro da memória e também ficam expostos para leitura assim que solicitado. A memória implementada no projeto conta com as seguintes portas:

- Entrada do clock externo
- Entrada habilitadora para leitura/escrita na memória
- Endereço e operação recebido do bloco operacional REM
- Saída de dados (Retorno)

Célula

A memória implementada possui dezesseis células enumeradas de zero a quinze. Cada célula possui internamente um componente contendo uma lógica capaz de gerar oito flip flops do tipo JK, e portanto, capaz de armazenar uma informação de oito bits. A principal funcionalidade implementada dentro da entidade célula é a seleção de uma célula específica dentro de um conjunto de dezesseis células. Para tal feito, foi necessário desenvolver uma lógica que pudesse manipular o endereço recebido pelo bloco operacional REM e comparar com uma "chave" exclusiva pertencente à célula.

Decodificador

O decodificador serve para receber um código recebido de opcode em uma instrução para a Unidade de Controle (UC), tivemos uma entrada que se chama instrução que contém oito bits de memória, porque os quatro primeiros bits foram reservados para memória de dados e os quatro últimos são reservados para as instruções que ativarão as variáveis que contém na UC, que é o cérebro de todo o programa.

PC (contador de Programa)

- Há quatro entradas, sendo o clk (clock); o R representa o endereço de memória atual onde os dígitos menos significativos (da posição 7 para 4) representam a instrução e os mais significativos (da posição 3 para 0) servem para dados recebidos pelo usuário; carga serve para escolher o dado recebido da entrada R ou o valor que esteja "rodando" dentro do PC, sendo assim ele desabilita o incremento; a variável incrementa serve para escolher entre o valor que esteja "rodando" dentro do PC ou se escolhe o valor incrementado por um somador.
- A saída só tem a E que é a memória que atualizará o Endereço de memória.

ULA e AC

- A ULA é a parte do processador que realiza cálculos de lógica aritmética e lógicas. A ULA deste trabalho possui duas entradas de 8 bits, 1 seletor de 3 bits, 1 saída 8 bits e outra saída de 2 bits. Além disso, possui um componente chamado Soma para que, dessa forma, esse bloco possa realizar a operação de adição. A ULA realiza quatro tipos de operação, para isso foi necessário fazer o uso da estrutura with select para codificar com 3 bits cada uma, o resultado foi armazenado em y.

Multiplexador

O multiplexador, neste trabalho, é um dispositivo que seleciona entre duas entradas de 8 bits, o endereço de memória do PC ou o endereço de memória da RDM. Para isso, existe uma chave seletora que varia entre 0 ou 1. Ainda, existe uma saída de 8 bits que vai servir como entrada para o REM.

REM

REM tem a função de endereçar a memória. Ele possui dois ports principais: `endereco_in` que é endereço de memória de 8 bits proveniente do multiplexador, e a saída `endereco_out` de 8 bits, que é o endereço de memória que vai acionar determinada célula de memória. Para obter essa saída, foi necessário fazer o uso de um conjunto de FlipFlops JK, então foi usado o FlipFlop JK como component e elaborado um laço com o for no port map. Como resultado, foi obtido um uma saída de oito bits no formato XXXX YYYY em que XXXX se refere a operação a ser realizada e Y é o endereço de memória. As 4 primeiras posições da saída foram preenchidas com 0 e as quatro restantes com o endereço de memória.

RDM

O RDM tem a função de armazenar o dado que vem da memória. No momento em que o endereço entra no REM, a memória libera o conteúdo na saída, esse conteúdo não entra automaticamente na RDM, ele entra a partir do momento que é dado um clock nele.

Simulação

