UNIVERSIDADE FEDERAL DO PIAUÍ

José Emanuel Rêgo de Souza

Isaac Silva Santos Ramos

IMPLEMENTAÇÃO DO MIC-1 DE TANENBAUM

Teresina – Piauí

2019

José Emanuel Rêgo de Souza

Isaac Silva Santos Ramos

IMPLEMENTAÇÃO DO MIC-1 DE TANENBAUM

Relatório final, apresentado

ao professor Ivan Saraiva Silva,

como parte da exigência feita ao

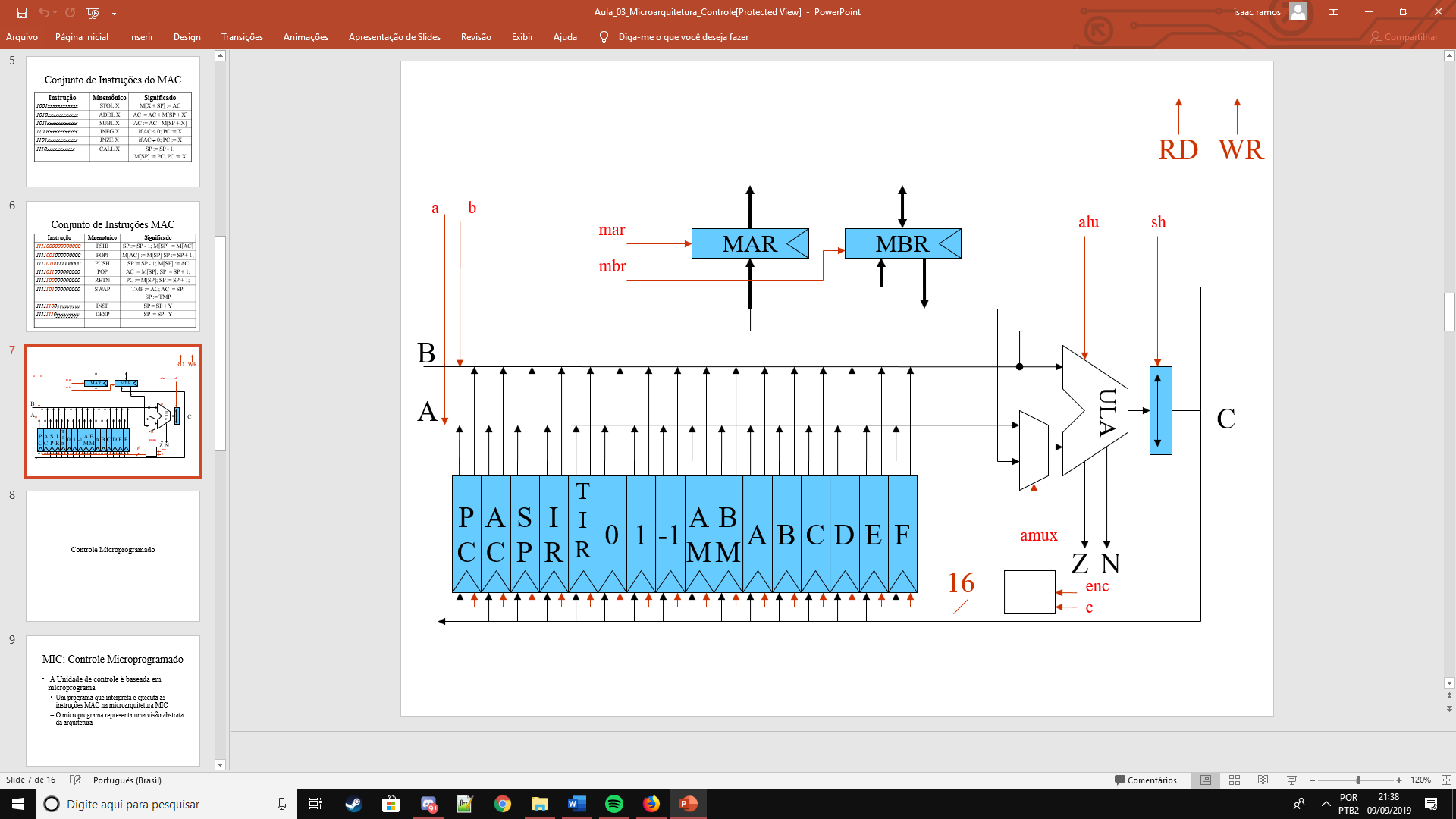
termino do exercício.

INTRODUÇÃO:

Em um nível mais baixo de codificação, o MIC-1 apresenta todos os componentes básicos com os quais se constrói o nível de microprogramação. A parte envolvida no projeto base do MIC feito por Tanenbaum e neste relatório foi a Parte Operativa (PO).

O **caminho de dados** é a parte do MIC que contém a Unidade Lógica e Aritmética (ULA), os barramentos A, B e C. Como apresentado na Figura 0.1.

Figura 0.1



Operação do esquema acima é explicado:

1. Os barramentos A e B alimentam uma ULA de largura de 16 bits, que pode realizar quatro funções: (A+ B), (A and B), (A), (not A). Cuja função é executada pela entrada de habilitação (amux). A ULA gera dois bits, com base na sua saída:
2. N, que é ligado quando a saída é negativa
3. Z, que é ligado quando a saída é composta por zeros.
4. A saída entra em um deslocador, que pode desloca-la 1 bit em qualquer direção ou não fazer nada, cabe ao sinal (SH), de 2 bits:
5. 00: não desloca.
6. 01: desloque 1 bit à esquerda.
7. 10: desloque 1 bit à direta.
8. 11: não usado.
9. Para comunicação com a memória, foi incluído um MAR e um MBR.
10. Nas leituras, no MBR, o dado lido da memória pode chegar à entrada esquerda da ULA através do multiplexador (amux).
11. Um banco de registradores de 16 bits cada.

Outrossim, é necessário:

1. 16 sinais para controlar a carga do barramento A e B, respectivamente, a partir da memória de rascunho.
2. 16 sinais para controlar da memória de rascunho a partir do barramento C.
3. 2 sinais para controlar as funções da ULA.
4. 2 sinais para controlar o deslocador.
5. 4 sinais para controlar o MAR e MBR.
6. 2 sinais para indicar leitura da memória e escrita na memória.
7. 1 sinal para controlar amux.

Um ciclo consiste em colocar valores nos barramentos A e B, passar os valores através da ULA e do deslocador e, finalmente, armazenar o resultado na memória de rascunho e/ou MBR. Outrossim, o MAR pode também ser carregado e um ciclo de memória iniciado.

O próximo passo do projeto foi implementar o formato de microinstrução contendo 22 bits. A microinstrução contém 13 campos, 11 dos quais são os seguintes:

1. amux: controla a entrada esquerda da ULA
2. ula: função da ULA
3. sh: função do deslocador
4. mbr: carrega MBR a partir do deslocamento
5. mar: carrega MAR a partir do barramento B
6. rd: requisita leitura da memória
7. wr: requisita escrever na memória
8. enc: controla armazenamento na memória de rascunho
9. c: seleciona registrador para armazenamento, dependendo de enc
10. b, a: seleciona fonte do barramento B e A, respectivamente.