Programa para Excelência em Microeletrônica

Módulo: Nivelamento

Matéria: Sistemas Digitais

Desenvolvimento de um Microcomputador

Henrique Martins Miranda

Sumário

[Introdução iii](#_Toc466277862)

[Teoria iii](#_Toc466277863)

[Desenvolvimento do Projeto iii](#_Toc466277864)

[Contador de Programa iv](#_Toc466277865)

[Unidade de Memoria iv](#_Toc466277866)

[Unidade lógica e aritmética iv](#_Toc466277867)

[Controlador v](#_Toc466277868)

[Registradores vi](#_Toc466277869)

[Conclusões viii](#_Toc466277870)

# Introdução

O objetivo principal desse projeto é a construção de um microcomputador SAP usando as ferramentas do Quartus Prime como forma de avaliação do nivelamento geral. O SAP por ser composto pelos principais circuitos e ter a arquitetura mais simples possível, facilita o entendimento dos detalhes relacionados a organização de computadores e sua aplicação.

# Teoria

O computador SAP (Simple As Possible) foi projetado com o objetivo de apresentar, de forma lúdica, a estrutura operacional de um computador eletrônico moderno, de maneira que fique mais simples abstrair a concepção usada nos computadores eletrônicos mais modernos.

# Desenvolvimento do Projeto

Foi dividido o microcomputador em seis partes principais para facilitar a sua construção, aproveitando que o Quartus tem um esquema de criação de circuitos parecido com programação orientada a objeto (criar classes, importa classes...), sendo elas: controlador de programa, unidade de memória, registrador de instruções, registrador de saída, controlador e a ULA (unidade lógica e aritmética).

*Figura 1. Microcomputador – uP.*

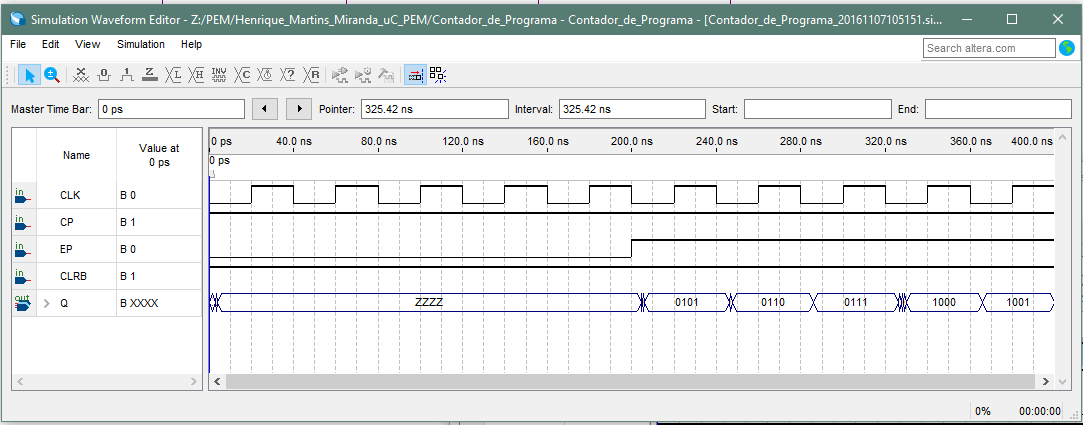


## Contador de Programa

Tem a funcionalidade de enviar à memória o endereço da próxima instrução a ser executada. A cada processamento, o contador é inicializado com *0000*.



*Figura 2a. Contador de programa – PC.*



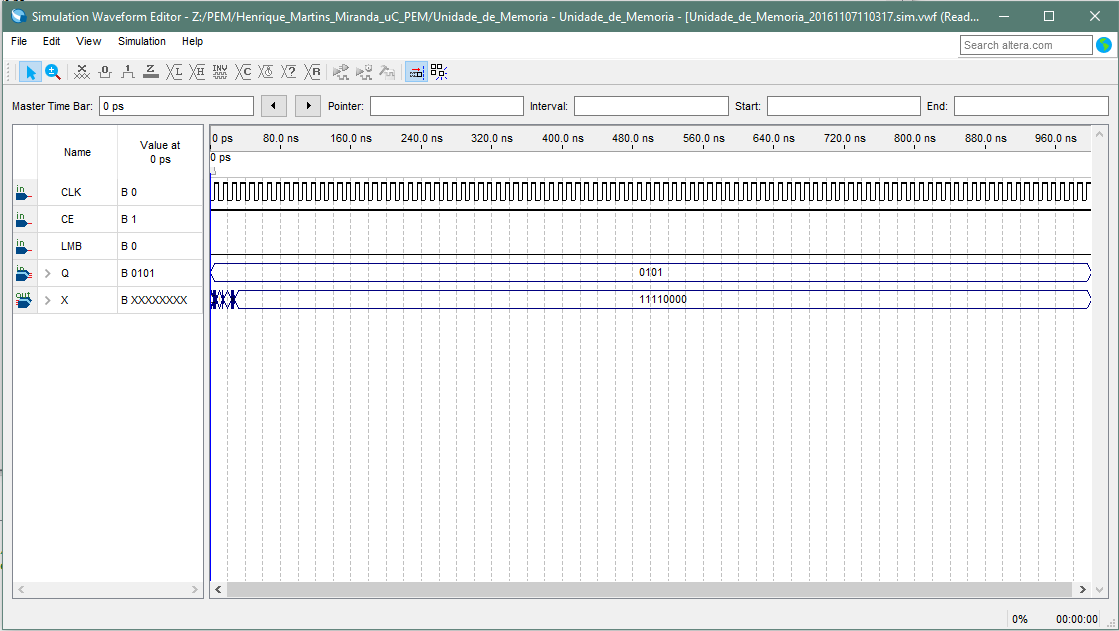
*Figura 2b.Diagrama de tempo, teste do PC.*

## Unidade de Memoria

É onde fica armazenado os dados. Pode ser dividida em duas partes: REM e ROM. A ROM é o armazenamento em si do microcomputador, a memória usada foi uma gerada automaticamente pelo Quartus com o endereçamento de 16bits e o tamanho de palavra de 8bits, já a REM, tem a função de registrar o ultimo endereço da memória acessada.



*Figura 3a. Unidade de memória.*



*Figura 3b. Diagrama de tempo, teste da unidade de memória.*

## Unidade lógica e aritmética

A ULA é composta por dois registradores, para armazena os valores inseridos e um somador/subtrator de 8 bits que é o núcleo dessa unidade.

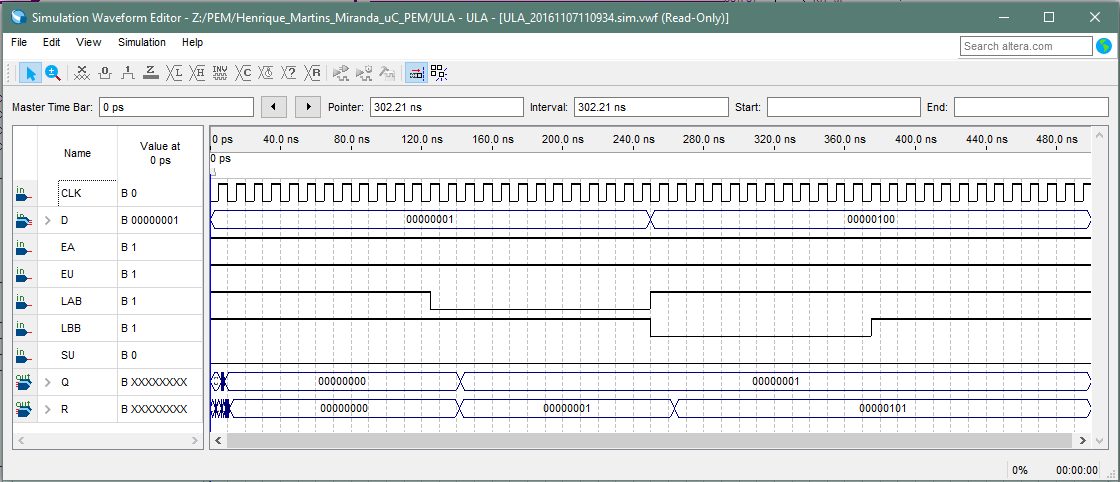
O somador/subtrator, como o nome já diz ele faz operações de soma e subtração entre dois valores (*X* e *Y*). A operação e decidida pelo *SU*, quando estiver baixo, a operação e de adição. Quando estiver alto, a operação e de subtração. Para construí-lo, usei de base o somador completo ensinado pelo tutorial “Como usar o Quartus Prime”, do professor Agripino.

Um dos registradores, o chamado *Acumulador A* serve para armazena os resultados intermediário calculados pelo *µP*. Ele é o *X* da unidade lógica.

O outro, chamado de *Registrador B*, fornece o número em operações aritméticas. Ele é o *Y* da unidade lógica.



*Figura 4a. ULA.*



*Figura 4b. Diagrama de tempo, teste do ULA.*

## Controlador

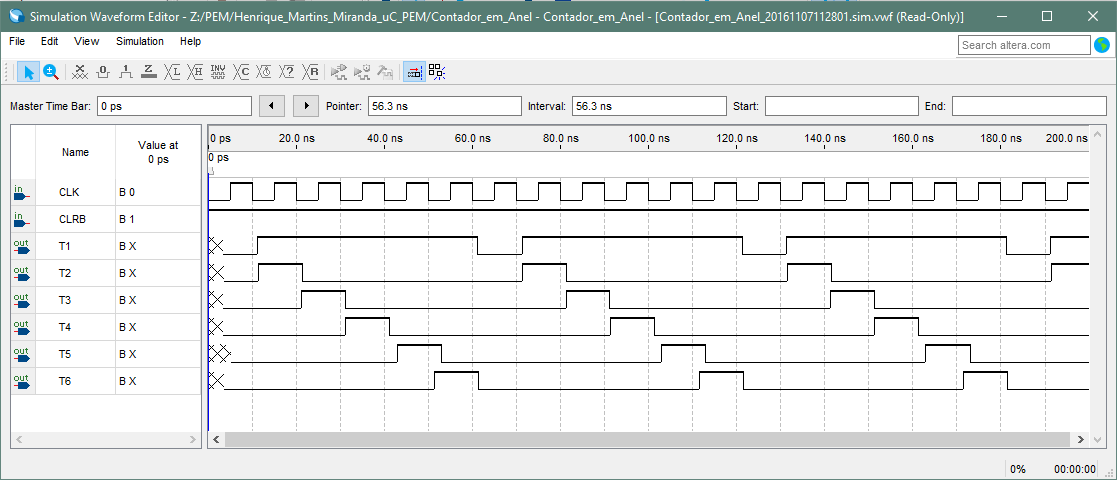
Tem a função de decodificar os 4 bits disponível para todas as instruções do µP e depois, decodificar as instruções para as entradas principais que vão controlar o microcomputador. Ele é dividido em 3 partes:

* Contador em anel, pode ser visto na figura 5. Ele tem a função parecida com a do *clock*;
* Decodificar de instruções, pode ser visto na figura 6;
* Matrix de controle, composta na figura 7, seve para decodificar as instruções nas principais entradas.

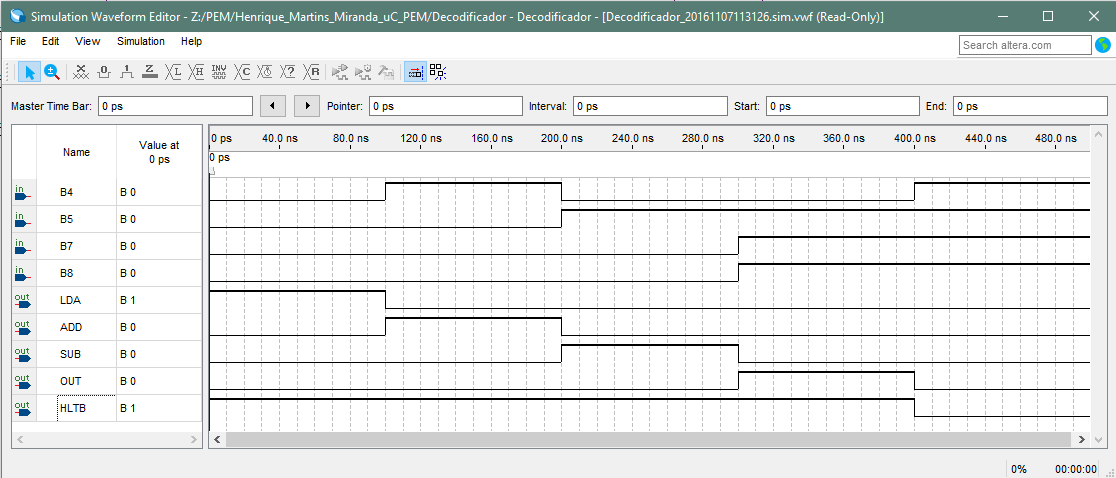


*Figura 5a. Contador em anel.*

*Figura 5b. Diagrama de tempo. Teste do contador em anel.*



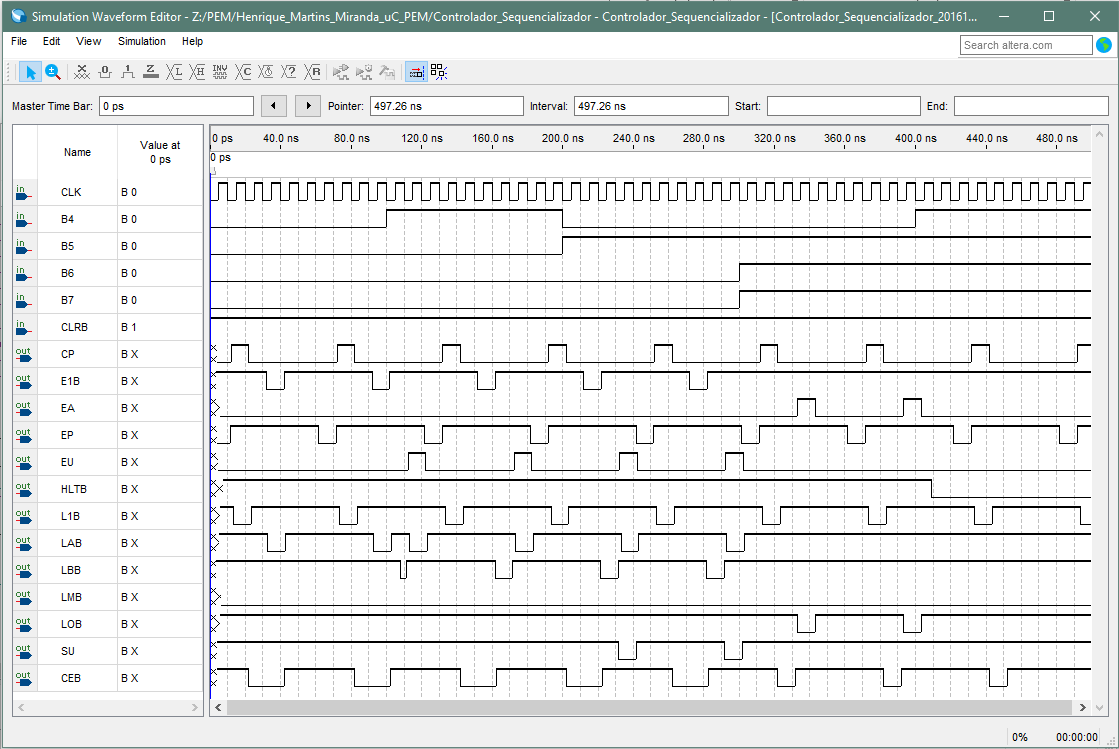
*Figura 6a. Decodificador de instruções.*



*Figura 6b. Diagrama de tempo. Teste do decodificador de instruções.*



*Figura 7a. Controlador/Sequencializador.*



*Figura 7b. Diagrama de tempo. Teste do Controlador/Sequencializador.*

## Registradores

Para todos os registradores usados no *uP,* foi utilizado o *74173* disponibilizado na biblioteca do próprio Quartus Prime. Esse modelo de registrador é bem parecido com o *74LS173* usado no circuito original, pode-se acessa o *datasheet* dele aqui <*bit.ly/2fTsr39*>. A única diferença entre eles é que no disponibilizado no Quartus não tem os pinos de *VCC* e de *GND*.



*Figura 8. Registrador de instruções.*



*Figura 9. Registrado de Saída.*

# Conclusões

Apesar de, não poder testa-lo, por diversos motivos (tempo, provas, Enem...), esse projeto me fez trabalhar muito, e com isso me deu uma esclarecida nas duvidas e no que não deu tempo ser ensinado, pelo curto tempo de um mês reservado para o nivelamento. Agora, não vou dizer que fixei *100%* do assunto, mas, pelo menos, não ficarei perdido nos projetos futuros.

Sobre o Git, não tive nem um problema, uso ele desde *2015.1,* por exigências dos professores; em relação ou Quartus Prime 16, foi fácil aprender a usá-lo, graças ao tutorial de Agripino. Único problema que tive foi nas simulações, por ter adicionado caracteres nos pinos que o Quartus interpreta como operações.