Programa para Excelência em Microeletrônica

Módulo: Nivelamento

Matéria: Sistemas Digitais

Desenvolvimento de um Microcomputador

Aluno: Aislan Jefferson de Souza Brito

Sumário

[Introdução iii](#_Toc466321597)

[Desenvolvimento do Projeto iii](#_Toc466321598)

[Contador de Programa (PC) iii](#_Toc466321599)

[Registrador de instrução (IR) iv](#_Toc466321600)

[ULA iv](#_Toc466321601)

[Acumulador A v](#_Toc466321602)

[Registrador B v](#_Toc466321603)

[Somador/Subtrator vi](#_Toc466321604)

[Controlador/Sequencializador vi](#_Toc466321605)

[RAM/REM vii](#_Toc466321606)

[Registrador de saída viii](#_Toc466321607)

[Conclusões viii](#_Toc466321608)

# Introdução

O projeto consiste em modelar um microcomputador SAP (Simple-As-Possible), utilizando os conceitos de diagrama esquemático e de bloco na ferramenta Altera® Quartus Prime.

# Desenvolvimento do Projeto

A implementação do projeto foi baseada no conceito bottom-up, no qual iniciou-se criando componentes mais básicos para reuso posterior. A seguir temos as descrições de cada bloco principal:

## Contador de Programa (PC)

O contador de programa é um contador de 4 bits que aponta o endereço da instrução a ser buscada e executada.



Figura – Visão geral do Contador de Programa

Em seu projeto original utiliza-se flip-flop JK sensíveis à descida de borda. Por a ferramenta Quartus não oferecer tal componente já pré-fabricado foi feito um componente de comportamento semelhante utilizando flip-flop JK sensíveis a subida de borda e um inversor em sua entrada de clock como visto na figura 2.

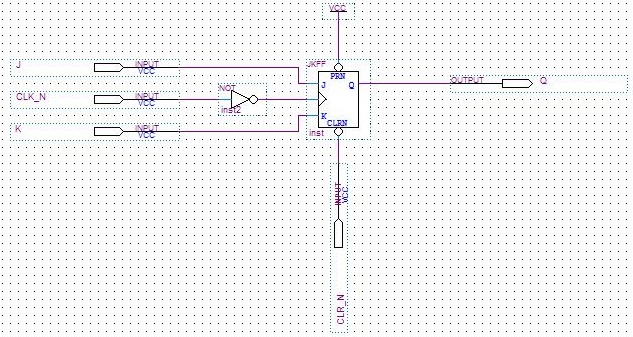


Figura 2 – Visão expandida componente FFJK\_CLKN

## Registrador de instrução (IR)

É responsável por armazenar a instrução endereçada na memória e que foi disponibilizada no barramento.

Consiste de dois registradores de 4 bits no qual um deles, que mantem comunicação com o barramento W, possui saída ligada a portas tristate como visto na figura 3.

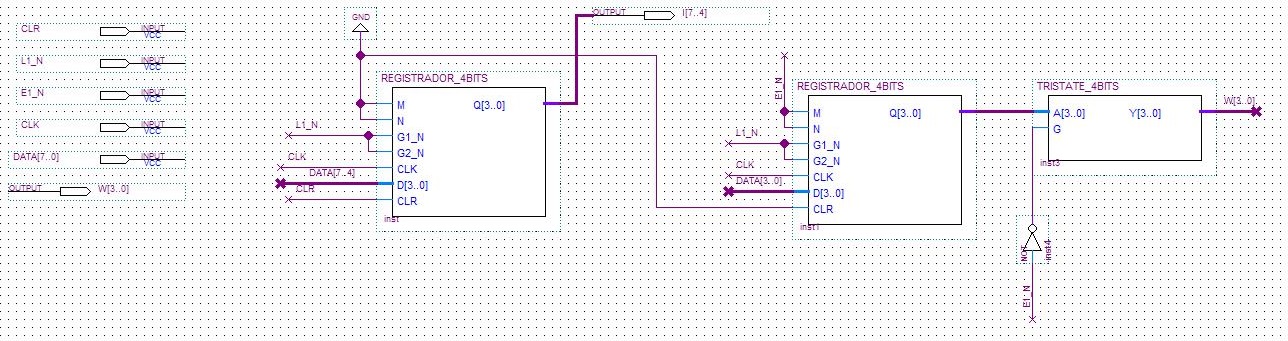


Figura 3 – Visão geral registrador de instrução (IR)

## ULA

Bloco responsável por realizar operações aritméticas, consiste de dois blocos de armazenamento o acumulador A e o registrador B, e também de um somador/subtrator no qual efetivamente realiza as operações.

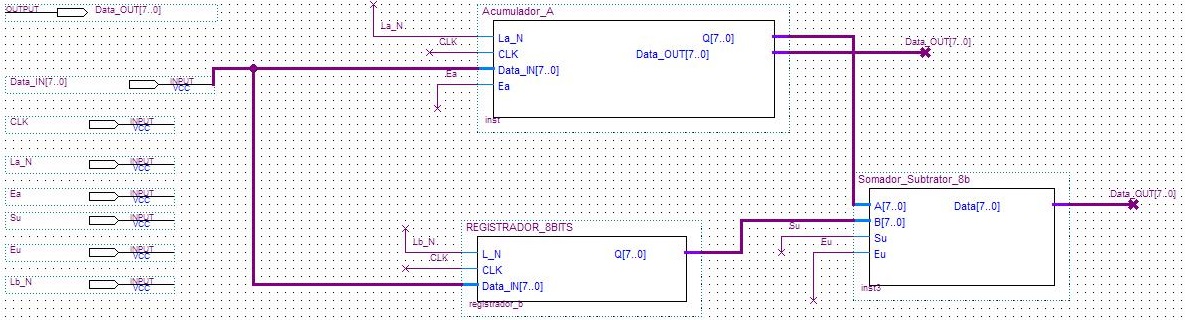


Figura 4 – Visão geral ULA

## Acumulador A

O acumulador é um registrador de 8 bits utilizado para armazenar um dos operandos para a operação aritmética.

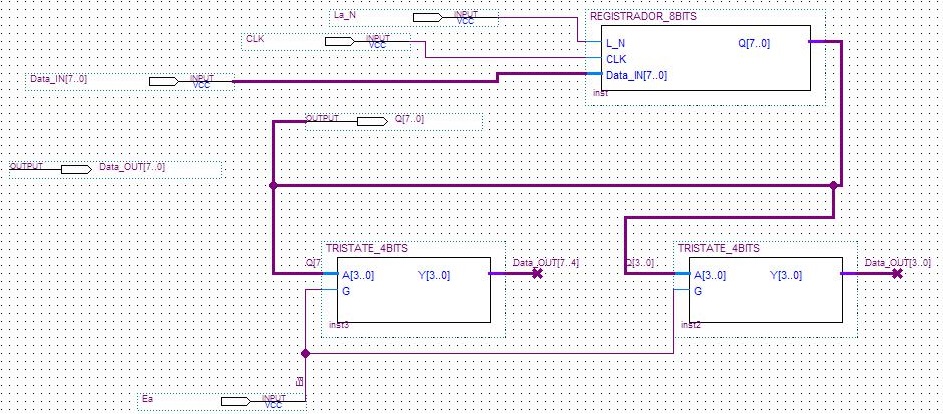


Figura 5 – Visão geral acumulador A

Onde o registrador de 8 bits é uma composição a partir do componente registrador 4 bits, como visto na figura 5.

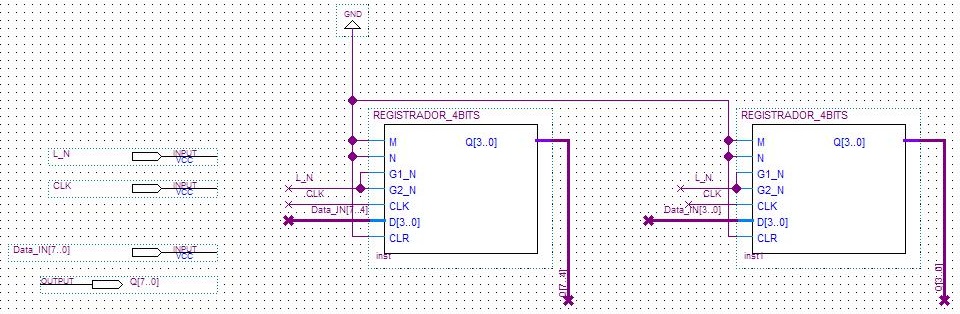


Figura 6 – Visão expandida registrador 8bits

## Registrador B

O registrador B interno ao diagrama da ULA é um registrador de 8bits, como visto na figura 6, que possui função de fornecer um dos operandos ao somador/subtrator

## Somador/Subtrator

Efetua soma/subtração de 8 bits, é estruturado como na figura 7 a seguir.

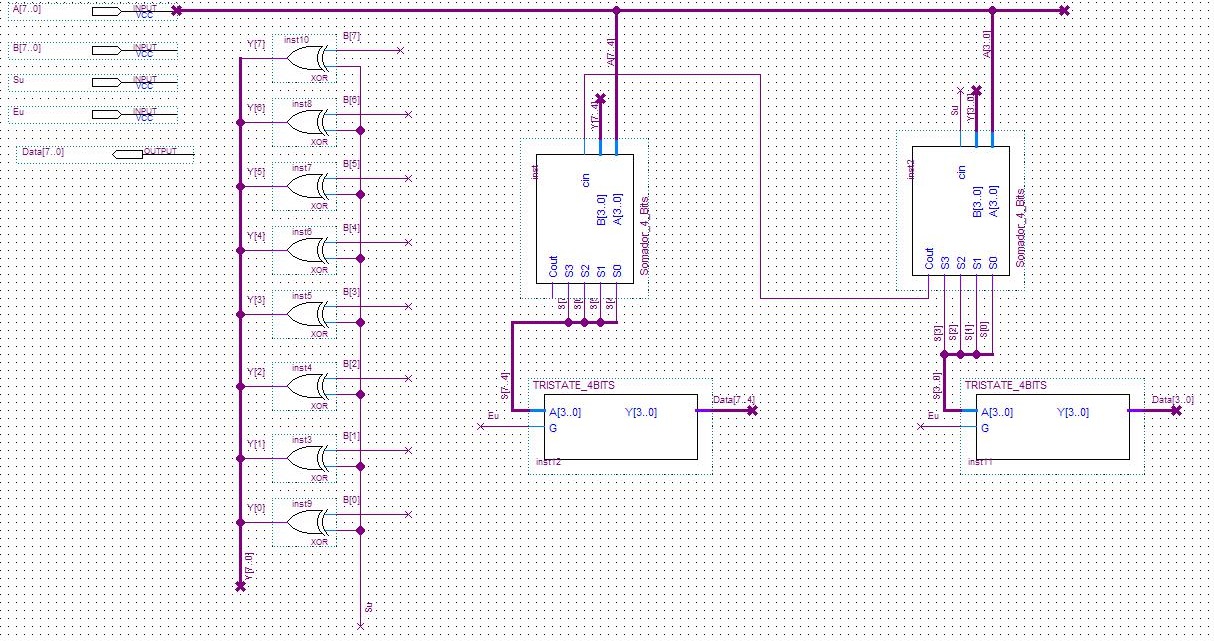


Figura 7 – Visão expandida somador/subtrator

## Controlador/Sequencializador

Consistido de um decodificador de instrução (figura 9), de um contador em anel (figura 10) e uma matriz de controle (figura 8). É responsável por mandar os sinais de controle para os outros componentes do microprocessador.

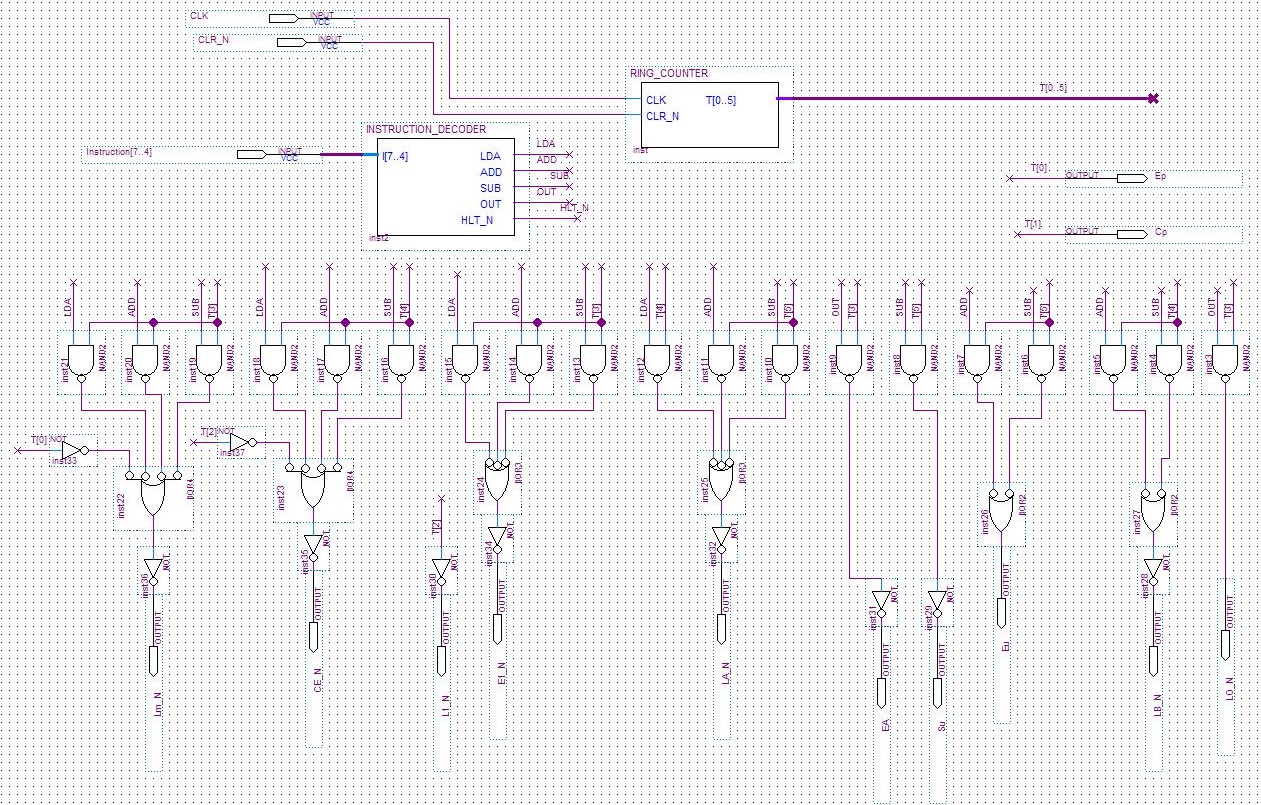


Figura 8 – Visão geral do bloco controlador/sequencializador

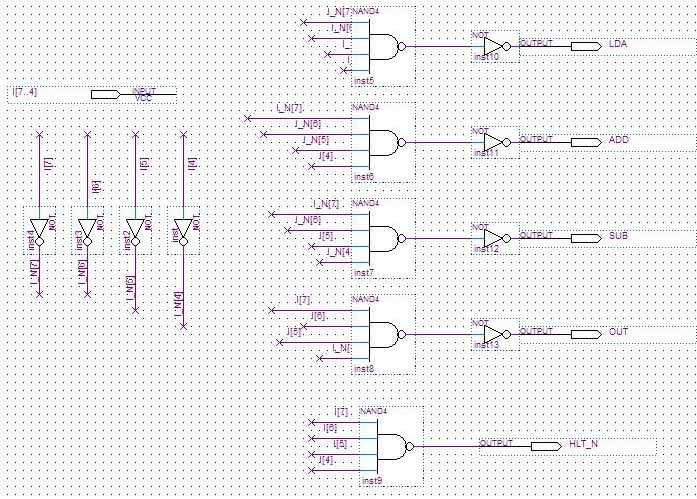


Figura 9 – Visão expandida decodificador de instrução

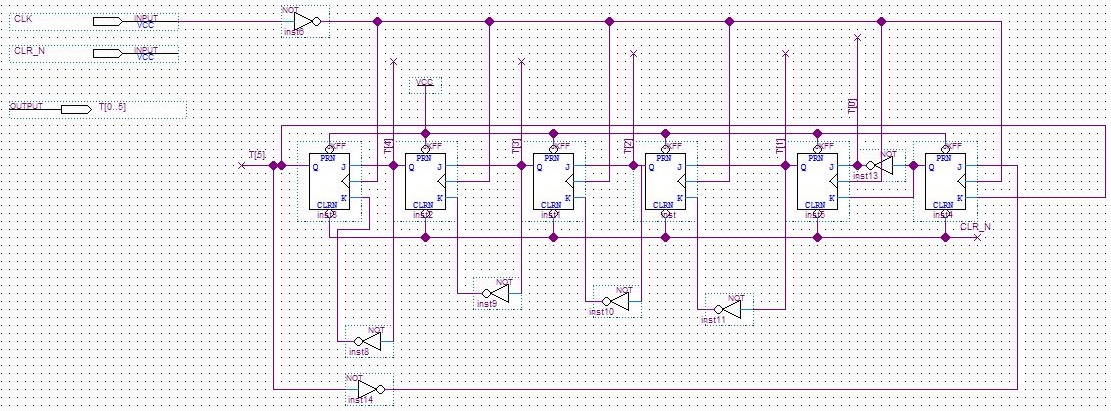


Figura 10 – Visão expandida contador em anel

## RAM/REM

O bloco de memorias foi desenhado de forma que integrasse a REM e a RAM em um único bloco. A RAM é carregada previamente com o arquivo uP\_Programa.hex

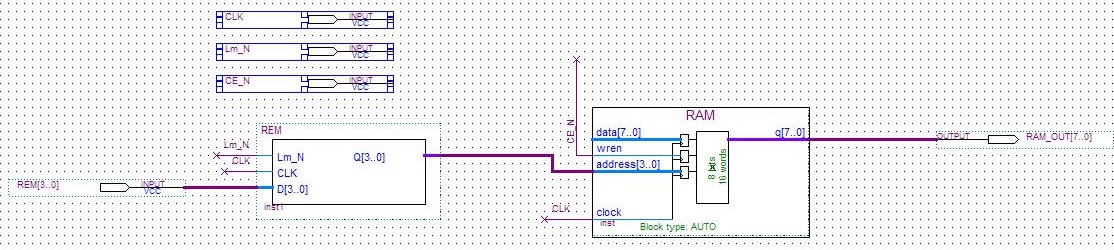


Figura 11 – Visão geral do bloco REM\_RAM

O bloco REM consiste de um registrador de 4 bits, como visto na figura 12.

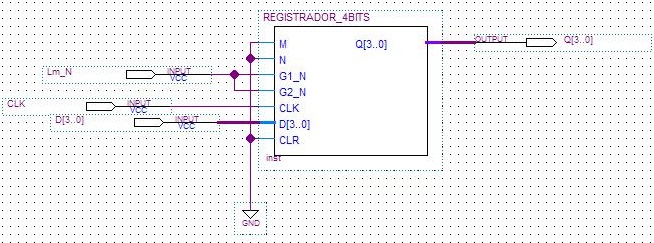


Figura 11 – Visão expandida do bloco REM

## Registrador de saída

Responsável por transferir o conteúdo do barramento para exibição (saída), consiste de um registrador de 8 bits como visto na figura 12.

# Conclusões

O projeto topo de nível ficou da seguinte forma:

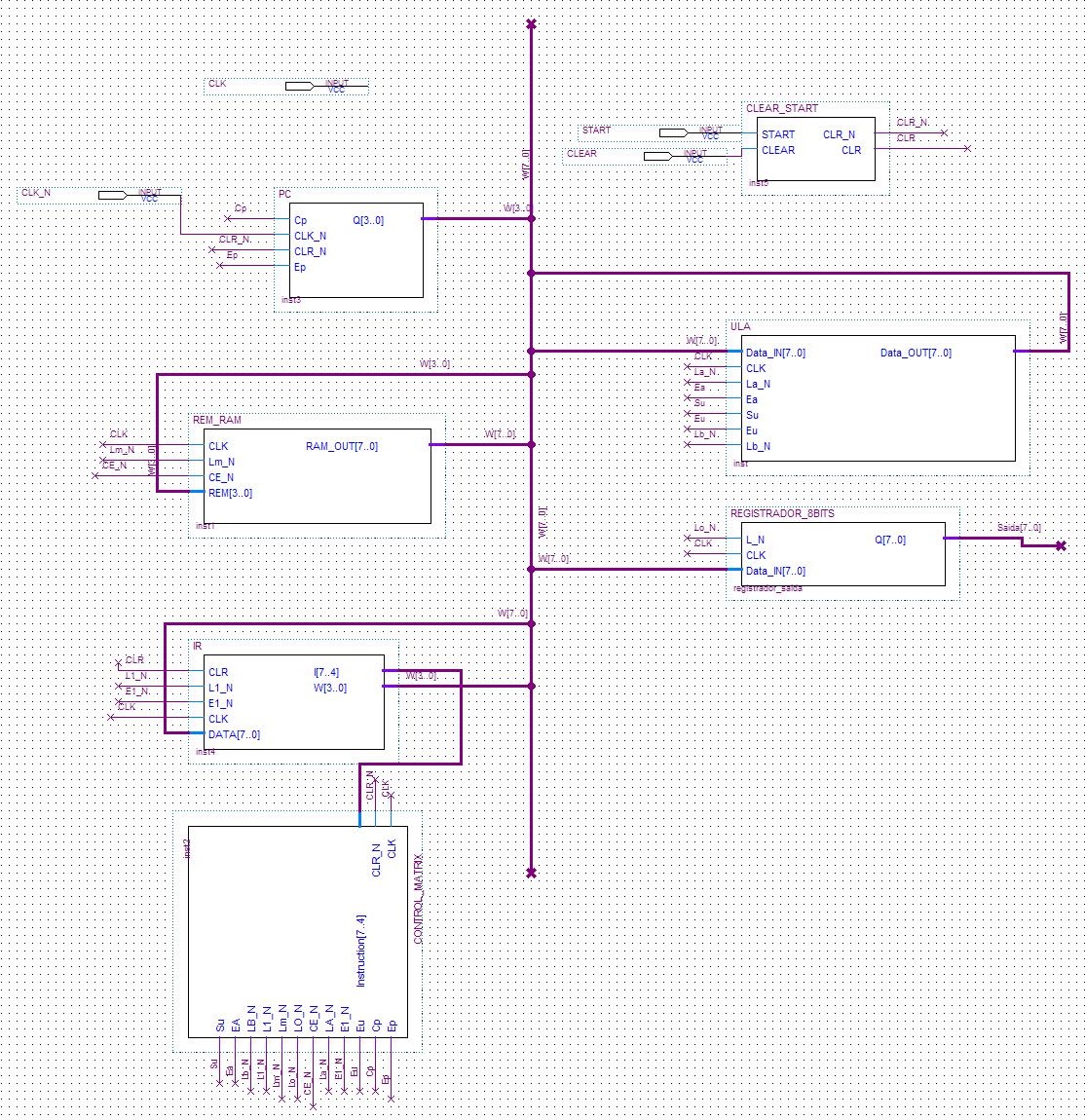


Figura 12 – Visão geral uPC