Programa para Excelência em Microeletrônica

Módulo: Nivelamento

Matéria: Sistemas Digitais

Desenvolvimento de um Microcomputador

Aluno: Filipe Cazuza Cavalcanti

Sumário  
  
capa 1  
sumário 2  
introdução 3  
teoria 3  
Desenvolvimento 4  
 1. Contador de programa (ring counter) 4

2. Acumulador 4

3. Registrador de saída 4

4. Registrador de instruções 4

5. Registrador-b 4

6. Somador/subtrator 4

7. Memória RAM 4

8. Controlador / sequencializador 5

Conclusões 5

Anexos[1...8] 6...10

**Introdução**

Este projeto tem como fins didáticos, o desenvolvimento do computador tipo SAP-1 (Simple as Possible) , a finalidade do SAP é introduzir todos os conceitos e ideias principais para criação e operação de um microprocessador básico, de forma a compreender o papel de seus principais componentes, instruções e entender a essência do sistema computacional.

Tendo objetivos:

* Uso da ferramenta QUARTUS PRIME.
* Entendimento da arquitetura básica de um processador.
* Entendimento de como as instruções funcionam.
* Uso do conhecimento ministrado nas aulas de nivelamento.

**Teoria**

A arquitetura **SAP-1** (*simple as possible*) é projetada para iniciar os estudos sobre microeletrônica, onde, toda sua comunicação é realizada através de um barramento principal W de 8 bits, ressaltando apenas alguns blocos que precisam se comunicar com áreas específicas do computador para isto tendo ligações com tais áreas. Este computador realiza apenas **5** instruções mostradas na **tabela-1** tal dispositivo realiza basicamente somas e subtrações, sua programação utilizada é básica, no software QUARTUS PRIME se dá por meio de um arquivo **\*.HEX** contendo instruções e dados podendo ser escrito em algumas unidades de medidas como binário, octal e hexadecimal, esse arquivo é carregado na RAM que armazena até 16 palavras de 8 bits cada, onde cada instrução é salva como os 4 bits mais significativos (Ex.: **0000 0101** aqui é carregado a instrução LDA seguido dos quatro bits menos significativos que são dados salvos junto com o código da instrução).

**Tabela-1 (**instruções e sua descrição seguida de seu código**)**

|  |  |  |
| --- | --- | --- |
| Mnemônico | código | Descrição |
| LDA | **0000** | Carrega um valor da memória principal no acumulador. |
| ADD | **0001** | Soma um valor da memória principal com o acumulador. |
| SUB | **0010** | Subtrai um valor da memória principal do acumulador. |
| OUT | **1110** | Exibe o valor do acumulador nas saídas. |
| HLT | **1111** | Interrompe a execução do programa. |

**Desenvolvimento** **do** **Projeto**

O desenvolvimento do projeto se deu através de blocos para facilitar sua implementação a seguir tem se uma descrição detalhada de cada parte do circuito e em anexo a imagem do esquemático de cada parte do uP.

1. **Contador de programa (ring counter)**

Sendo parte da unidade de controle o contador do programa (*ring counter*) é responsável por gerar o endereço da instrução à memória **RAM** através de sua contagem que inicia em **0000** e vai até **1111,** tendo a entrada **Cp** em nível lógico alto a contagem continua sequencialmente, e quando **Cp** entra em nível baixo a contagem se mantém atual até que a matriz de controle ative **Cp** novamente continuando assim a contagem a partir do momento em que parou.

**2. Acumulador**

O acumulador armazena o operando enquanto a operação de soma ou subtração não finalizou bastando **La'** estar em nível lógico baixo, e após finalizado armazena o resultado da soma/subtração bastando a entrada **Ea** ser alto para ter seu valor disponibilizado no barramento **W**

**3. Registrador de saída**

No momento que a soma/subtração finaliza e o acumulador disponibiliza seu resultado no barramento **W** esperando apenas mais uma transição do clock para armazenar o resultado que está ocioso do barramento dependendo apenas de **Ea** em alto e **Lo'** em nível lógico baixo, onde, essa configuração salva os dados disponíveis, entra em ação o registrador de saída que disponibilizará o resultado para o mundo externo através de seu conjunto de saídas .

**4. Registrador de instruções**

O registrador de instruções salva o conjunto de bits correspondentes a cada instrução que o processador realiza sendo esses bits os 4 mais significativos disponibilizando sua saída para o decodificador de instruções.

**5. Registrador-b**

O registrador-b fornece uma das palavras para o somador/subtrator dependendo **Lb’** ele carrega os dados que estão no barramento W para si, e os manda para o somador para ser feita a operação aritmética junto com os dados que estão o acumulador.

**6. Somador/subtrator**

O somador/subtrator fazendo parte da **ULA,** junto com o acumulador e o registrador-b este é um dos únicos circuitos combinacionais presente neste **uP** junto com a matriz de decisão e o decodificador de instruções. Seu esquemático é o de um somador completo de 8 bits fazendo a subtração a partir de portas **XOR** ligadas nas entradas de uma das palavras (2 palavras de 8 bits) tendo umas das entradas da porta **XOR** ligadas a saída **Su** da matriz de decisão quando **Su** está alta realiza uma subtração em complemento de 2, quando **Su** está baixo é realizada uma soma.

**7. Memória RAM**

É na memória RAM que está salva toda a programação do uP junto com os dados que serão processados pelo SAP-1 neste projeto o bloco correspondente a memoria ram ja esta incluso o registrador de endereço reduzindo assim um bloco do projeto original. Sendo uma memória 8 x 16 salva as instruções nos quatro bits mais significativos e nos primeiros endereços da memória, como este processador opera com clock de 1Mhz e funciona sequencialmente, se os dados estiverem nos últimos endereços da memória, mesmo que em uma memória a velocidade de acesso de um endereço é o mesmo para todos, ele iria acessar antes de chegar as instruções todos os outros endereços de memória aumento assim o tempo do processamento como não queremos isso colocamos todos os dados nos primeiros endereços da memória.

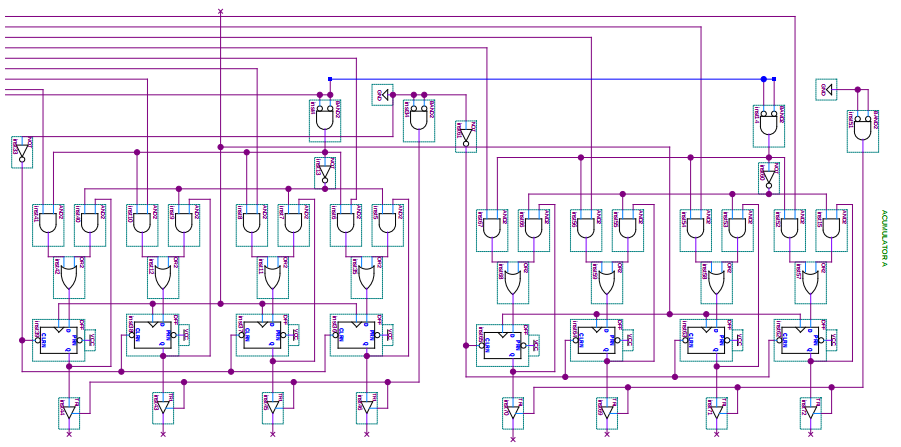
**8. Controlador / Sequencializador**

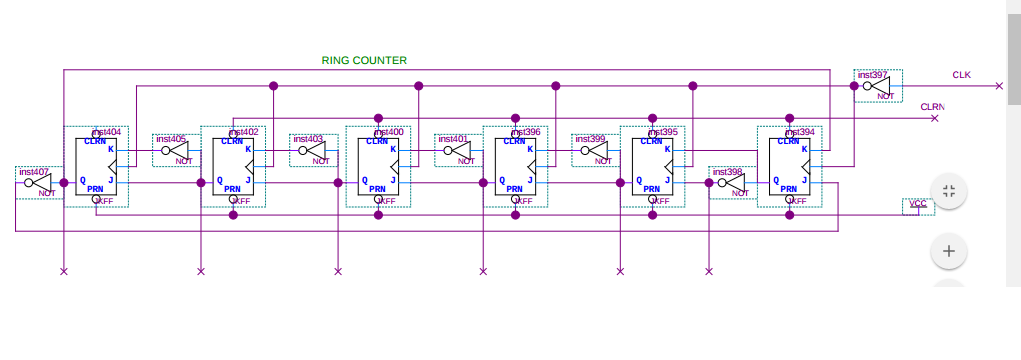
Fazendo parte da unidade de controle junto com o contador (*ring counter*) o controlador / sequencializador tem um conjunto de **12 bits** de saída que comandam o resto do computador essa palavra determina como os registradores agiram a próximo sinal de clock, determinando ainda a operação aritmética que será utilizada com sua saída **Su**, ela estando em nível lógico alto o somador/subtrator realizará uma soma e **Su** entrando em baixo será realizada a soma.

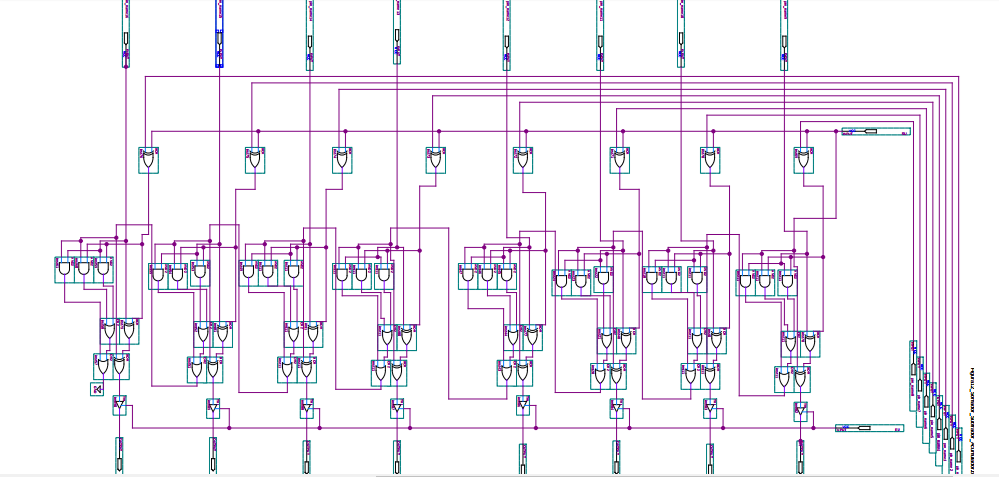
**Conclusões**

Ao fim deste projeto conseguimos entender o funcionamento básico de um microprocessador, sua comunicação com seus componentes internos, como as instruções são tratadas e executadas, o entendimento de cada parte assim como a importância das mesmas no sistema computacional. Aprofundamento de conhecimentos sobre memória, e sobre processamento em si, também foi possível conhecer sobre ciclos de trabalho e de máquina. conhecimento sobre arquitetura de barramento que sem bem entendidos podem facilitar a implementação de um uP dependendo de sua complexidade, ressalto aqui algumas observações; infelizmente por falta de conhecimento da ferramenta de criação QUARTUS PRIME ou por alguma falha na implementação do uP, não conseguimos chegar ao resultado esperado mesmo fazendo cada componente funcionar de maneira esperada separadamente, revisando cada conexão entre todos os blocos incansáveis vezes, mesmo depois de toda contemplação, o melhor diagnóstico que temos até o presente momento é que a falha está na memória RAM que implementamos no uP.

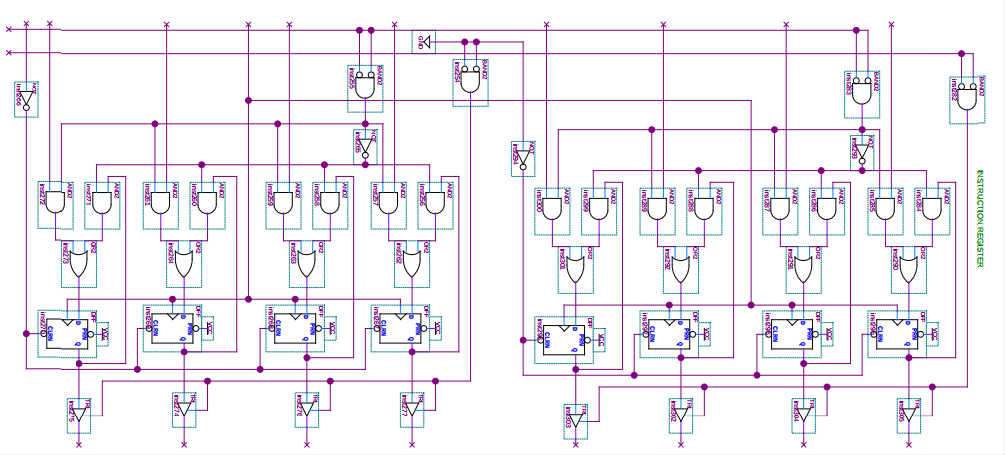
Anexo-1 Acumulador



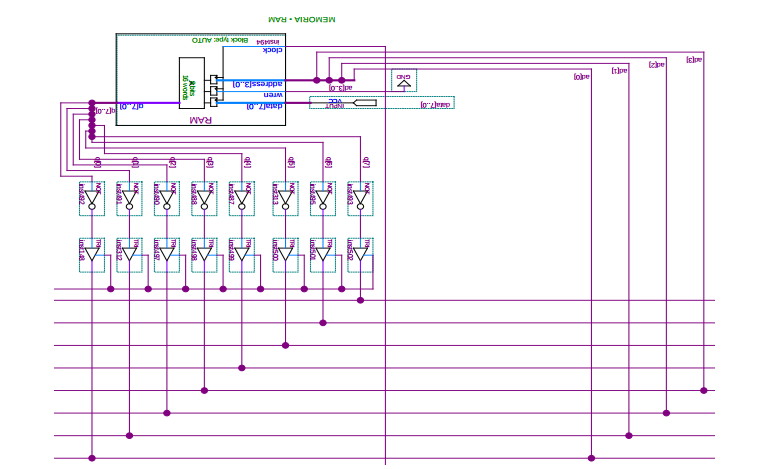
Anexo-2 contador de programa (ring counter)

Anexo-3 somador/subtrator

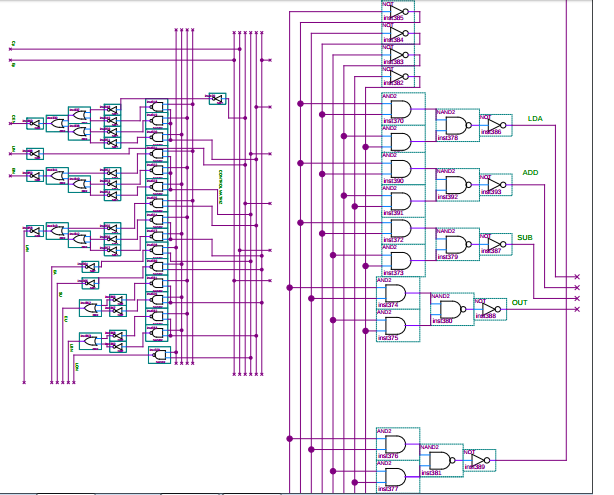
Anexo-4 registrador de instruções



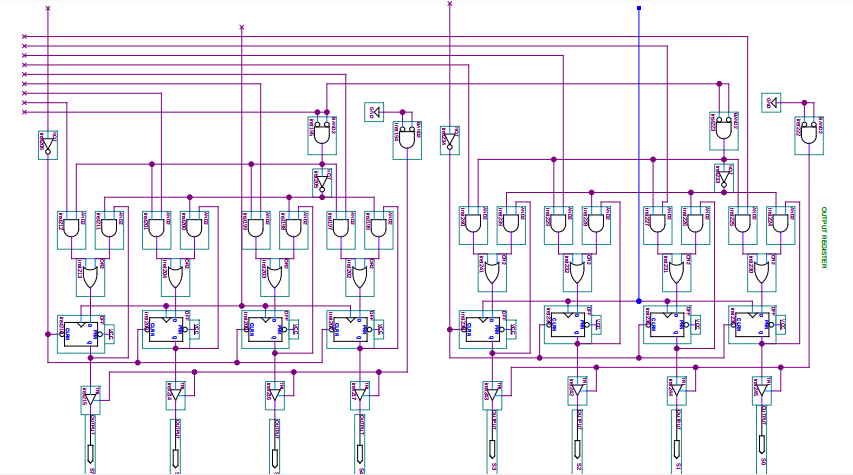
Anexo-5 memória RAM



Anexo-6 controlador sequencializador



Anexo-7 registrador de saída



Anexo-8 Registrador-b

