PODER EXECUTIVO

MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE RORAIMA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO DO PROJETO: PROCESSADOR 8-BITS

ALUNOS:

LUCAS DANIEL MIRANDA FREITAS - 2018005584

Maio de 2021

Boa Vista/Roraima

PODER EXECUTIVO

MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE RORAIMA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO DO PROJETO: PROCESSADOR 8-BITS

Maio de 2021

Boa Vista/Roraima

Resumo

Este trabalho aborda o projeto e implementação de um processador com arquitetura de 8(oito) bits, escrito na linguagem VHDL e executado na estrutura do programa Quartus II da Intel.

Será descrito com detalhes todos os componentes importantes do processador, que são necessários para o funcionamento adequado do mesmo.

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Especificação

Nesta seção é apresentado o conjunto de itens para o desenvolvimento do processador 8-BITS, bem como a descrição detalhada de cada etapa da construção do processador.

1.1 Plataforma de desenvolvimento

Para a implementação do processador 8-BITS foi utilizado a IDE Quartus Prime Versão 20.1.1 Build 720 11/11/2020 SJ Lite Edition, com o simulador ModelSim e a linguagem usada para a programação foi o VHDL.

Flow Status	Successful - Tue May 18 00:49:21 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	bancoregistrador
Top-level Entity Name	bancoregistrador
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	48
Total pins	30
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

Figura 1 - Especificações no Quartus

1.2 Conjunto de instruções

O processador 8-BITS possui 4 registradores: S0, S1, S2, S3. Assim como dois formatos de instruções de 8 bits cada, Instruções do **tipo** (do tipo **R** e **J**), seguem algumas considerações sobre as estruturas contidas nas instruções:

- **Opcode**: a operação básica a ser executada pelo processador, tradicionalmente chamado de código de operação;
- **Reg1**: o registrador contendo o primeiro operando fonte e adicionalmente para alguns tipos de instruções (ex. instruções do tipo R) é o registrador de destino;
- **Reg2**: o registrador contendo o segundo operando fonte;

Tipo de Instruções:

- Formato do tipo R: Este formatado aborda instruções de Load , Store e instruções baseadas em operações aritméticas e aritméticas imediatas.

Formato para escrita em código binário:

Opcode	Reg2	Reg1
4 bits	2 bits	2 bits
7-4	3-2	1-0

Tabela 1 – Formato para escrita em código Binário.

Visão geral das instruções do Processador 8-BITS:

O número de bits do campo **Opcode** das instruções é igual a quatro, sendo assim obtemos um total de 16 Opcodes **(0-15)** que são distribuídos entre as instruções, assim como é apresentado na Tabela 2.

Tabela 2 – Tabela que mostra a lista de Opcodes utilizadas pelo processador 8-BITS.

Opcode	Nome	Formato	Breve Descrição	Exemplo
0000	ADD	R	Soma	**add** $S0, $S1 , ou seja, $S0 := $S0+$S1
0001	ADDI	I	Soma Imediata	**addi** $S0, 22 , ou seja, $S0 := $S0+22

0010	SUB	R	Subtração	**sub** $S0, $S1 , ou seja, $S0 := $S0 - $S1
0011	SUBI	I	Subtração Imediata	**subi** $S0, 22 , ou seja, $S0 := $S0 - 22
0100	MULT	R	Multiplicação	**mult** $S0, $S1 , ou seja: , $S0 := $S0 * $S1
0101	MULTI	I	Multiplicação Imediata	**multi** $S0, 22 , ou seja: , $S0 := $S0 * 22
0110	MOVE	R	Troca	**move** $S0, $S1 , ou seja: $S0 := $S1
0111	MOVI	I	Troca Imediata	**movi** $S0, 22 , ou seja: $S0 := 22
1000	LW	I	Load Word	**lw** $S0, memoria (00) , ou seja, $S0 := Valor memoria (00)
1001	SW	I	Store Word	**sw** $S0, memoria (00) , ou seja, memoria (00) := $S0

1.3 Descrição do Hardware

Nesta seção são descritos os componentes do hardware que compõem o processador Quantum, incluindo uma descrição de suas funcionalidades, valores de entrada e saída.

1.3.1 ULA

O componente ULA (Unidade Lógica Aritmética) tem como principal objetivo efetuar as principais operações aritméticas, dentre elas: soma, subtração, multiplicação. Adicionalmente a ULA efetua operações de comparação de valor igual. O componente ULA recebe como entrada três valores: A – dado de 8bits para operação; B - dado de 8bits para operação e OP – identificador da operação que será realizada de 4bits. A ULA também possui uma saída: result – saída com o resultado das operações.

1.3.2 Banco de Registradores

O banco de registrador funciona de seguinte maneira: Funciona como um seletor de dados, caso o write_reg esteja em 0, é procurado os dados registrados no input_reg_1 e input_reg_2. Caso o write_reg esteja em 1, o banco de registrador irá guarda o dado contido no data_reg no input_reg_1.

1.3.3 Controle

O componente Control tem como objetivo realizar o controle de todos os componentes do processador de acordo com o opcode ... Esse controle é feito através das flags de saída abaixo:

§ **LerMem:** 1 Bit, manda o comando pra memória de dados, informando se vai ser lido da memória.

§ **MemParaReg**: 1 Bit, manda o comando ao multiplexador, para a escolha do dado a ser escrito no registrador.

§ **UlaOp**: 4 Bits, informa qual operação aritmética será realizado na ULA.

§ **EscMem:** 1 Bit, manda o comando pra memoria de dados, informando se vai ser escrito na memória.

§ **UlaFonte**: 1 Bit, manda o comando 0 ou 1 pro multiplexador anterior à ULA.

§ **Writw_reg** 1 Bit, manda o comando 1 pro bloco de registradores.

1.3.4 Memória de dados

Memória de dados responsável por armazenar temporariamente dados usados durante a execução das instruções.

1.3.5 Memória de Instruções

Se o clock está em nível alto, a saída output irá conter o valor da memória no endereço da entrada .

1.3.6 EXTENSOR DE SINAL DE 2 PARA 8

O extensor de sinal de 2 para 8 bits funciona da seguinte maneira: A entrada input com 2 bits será convertida para uma saída output com 8 bits.

1.3.7 Somador

Somador responsável por somar o resultado da instrução até o momento (instrução de 8 bits) com o número 1 em binário, recebendo assim duas entradas e apenas uma saída, com o resultado dessa soma.

1.3.8 Mux_2x1

O multiplexador de 2 entradas é um componente onde terá 2 entradas e apenas uma saída que será definida através do seletor dependendo do seu valor: caso o seletor seja "00", a saída do MUX terá o valor de E0. Caso o seletor seja "01", a saída do MUX terá o valor de E1.

1.3.9 PC

O componente responsável por armazenar uma instrução de 8 bits que será executada é responsável por mandar o endereço da instrução a ser executada.

1.4 Datapath

É a conexão entre as unidades funcionais formando um único caminho de dados e adicionando uma unidade de controle responsável pelo gerenciamento das ações que serão realizadas para diferentes tipos de instruções. Para o Processador 8-BITS foi decidido colocar a memória de dados e a memória de instruções, pois ambas possibilitam um gerenciamento melhor dos testes facilitando no momento da execução dos algoritmos que foram codificados.

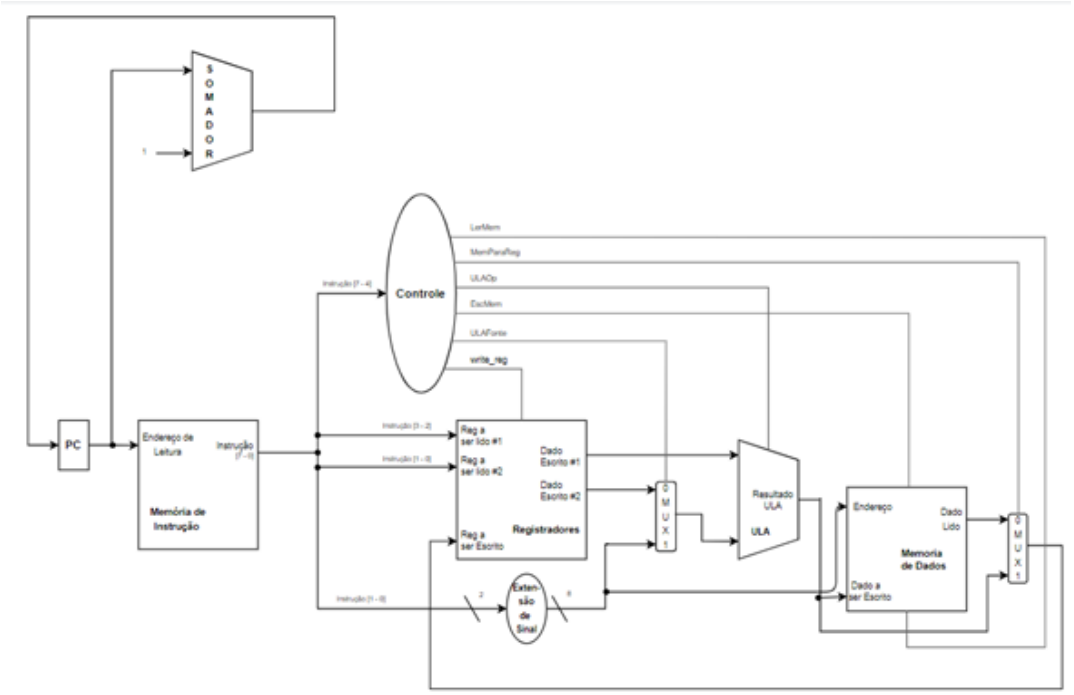

figura 2: Datapath 8 Bits

2 Considerações finais

Este trabalho apresentou o projeto e implementação do processador de 8 bits, que é a junção dos conhecimentos adquiridos durante o período de ensino da disciplina Arquitetura e Organização de Computadores. Os conhecimentos passados pelo professor Herbert.