



**PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE RORAIMA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO DO PROJETO: PROCESSADOR MENSIS

ALUNOS:

Robson Vinicius Pontes Silva - 2018015956

João Roberto Randel Costa Cunha - 2018015393

**Outubro de 2019
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 MENSIS

**Outubro de 2019
Boa Vista/Roraima**

Resumo

Neste trabalho será abordado os principais pontos da construção e implementação do processador multiciclo MENSIS de 8 bits baseado na arquitetura MIPS. Será descrito com detalhes cada etapa do processo de sua construção levando em conta todos os componentes necessários para o seu funcionamento e os testes realizados durante a implementação. O processador terá capacidade de executar 16 instruções, já incluso, a soma de ponto flutuante, dando a possibilidade bastante abrangente de executar algoritmos.

A implementação do processador foi feita integralmente com a linguagem de descrição de hardware VHDL e os teste foram analisados através simulador ModelSim que gera waveforms, demonstrando assim o comportamento do processador.

Conteúdo

1	Especificação	7
1.1	Plataforma de desenvolvimento	7
1.2	Conjunto de instruções	8
1.3	Descrição do Hardware	9
1.3.1	ALU ou ULA	9
1.3.2	BDRegister	9
1.3.3	Clock	9
1.3.4	Controle	9
1.3.5	Memória de dados	10
1.3.6	Memória de Instruções	10
1.3.7	Somador	10
1.3.8	And	10
1.3.9	Mux_2x1	10
1.3.10	PC	10
1.3.11	ZERO	11
1.4	Datapath	11
2	Simulações e Testes	13
3	Considerações finais	14

Lista de Figuras

FIGURA 1 - ESPECIFICAÇÕES NO QUARTUS	6
FIGURA 2 - BLOCO SIMBÓLICO DO COMPONENTE QALU GERADO PELO QUARTUS	8
FIGURA 19 - RESULTADO NA WAVEFORM.	13

Lista de Tabelas

TABELA 1 – TABELA QUE MOSTRA A LISTA DE OPCODES UTILIZADAS PELO PROCESSADOR XXXX.	7
TABELA 2 - DETALHES DAS FLAGS DE CONTROLE DO PROCESSADOR.	9
TABELA 3 - CÓDIGO FIBONACCI PARA O PROCESSADOR QUANTUM/EXEMPLO.	12

1 Especificação

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

1.1 Plataforma de desenvolvimento

Para a implementação do processador Mensis foi utilizado a IDE:

Flow Status Flow Failed - Tue Dec 03 01:47:59 2019
Quartus Prime Version 18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name Mensis
Top-level Entity Name Mensis
Family Cyclone V
Device 5CGXFC7C7F23C8
Timing Models Final
Logic utilization (in ALMs) N/A until Partition Merge
Total registers N/A until Partition Merge
Total pins N/A until Partition Merge
Total virtual pins N/A until Partition Merge
Total block memory bits N/A until Partition Merge
Total PLLs N/A until Partition Merge
Total DLLs N/A until Partition Merge

Figura 1 - Especificações no Quartus

1.2 Conjunto de instruções

O processador Mensis possui 4 registradores: S0, S1, S2 e S3. Assim como 3 formatos de instruções de 8 bits cada, Instruções 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 formato aborda instruções de Load, Store e instruções baseadas em operações aritméticas e instruções aritméticas imediatas.

Formato para escrita de código na linguagem Mensis:

Tipo da Instrução Reg1 Reg2

Formato para escrita em código binário:

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

- **Formato do tipo J:** Este formato aborda instruções do tipo jump condicional e incondicional.

-

Formato para escrita de código na linguagem Mensis:

Tipo da Instrução Label

2	10000
7-6	5-0

Visão geral das instruções do Processador Mensis:

O número de bits do campo **Opcode** das instruções é igual a quatro, sendo assim obtemos um total ($Bit(0 \text{ e } 1)^4 \therefore 2^4 = 16$) de 16 **Opcodes (0-15)** que são distribuídos entre as instruções, assim como é apresentado na Tabela 1.

Tabela 1 – Tabela que mostra a lista de Opcodes utilizadas pelo processador Mensis.

Tipo	OP	RS	RT	IMM	Exemplo	Instrução
R	0000	0-11	00-11	-	add \$s1, \$s2	add
R	0001	0-11	00-11	-	addi \$s1, WORD	addi
R	0010	0-11	00-11	-	sub \$s1, \$s2	sub
R	0011	0-11	00-11	-	subi \$s1, WORD	subi
R	0100	0-11	00-11	-	mult \$s1, \$s2	mult
R	0101	0-11	00-11	-	multi \$s1, WORD	multi
R	0110	0-11	00-11	-	eq \$s1, \$s2	eq
R	0111	0-11	00-11	-	eqi \$s1, WORD	eqi
R	1000	0-11	00-11	-	move \$s1, \$s2	move
R	1001	0-11	00-11	-	movi \$s1, WORD	movi
R	1010	0-11	00-11	-	lw \$s1, WORD	lw
R	1011	0-11	00-11	-	sw \$s1, WORD	sw
R	1100	0-11	00-11	-	addf \$s1, \$s2	addf
J	1101	-	-	0000-1111	bne Label	bne
J	1110	-	-	0000-1111	beq Label	beq
J	1111	-	-	0000-1111	j Label	j

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 ALU ou ULA

O componente ULA(Unidade Lógica Aritmética) tem como principal objetivo efetuar as principais operações aritméticas, dentre elas: soma, subtração, divisão (considerando apenas resultados inteiros) e multiplicação. Adicionalmente o ULA efetua operações de comparação de valor como maior ou igual, menor ou igual, somente maior, menor ou igual. O componente ULA recebe como entrada três valores: **A** – dado de 8 bits para operação; **B** - dado de 8 bits para operação e **OP** – identificador da operação que será realizada de 4 bits. O ULA também possui 2 saídas: **zero** – identificador de resultado (2bit) para comparações (1 se verdade e 0 caso contrário); e **result** – saída com o resultado das operações aritméticas.

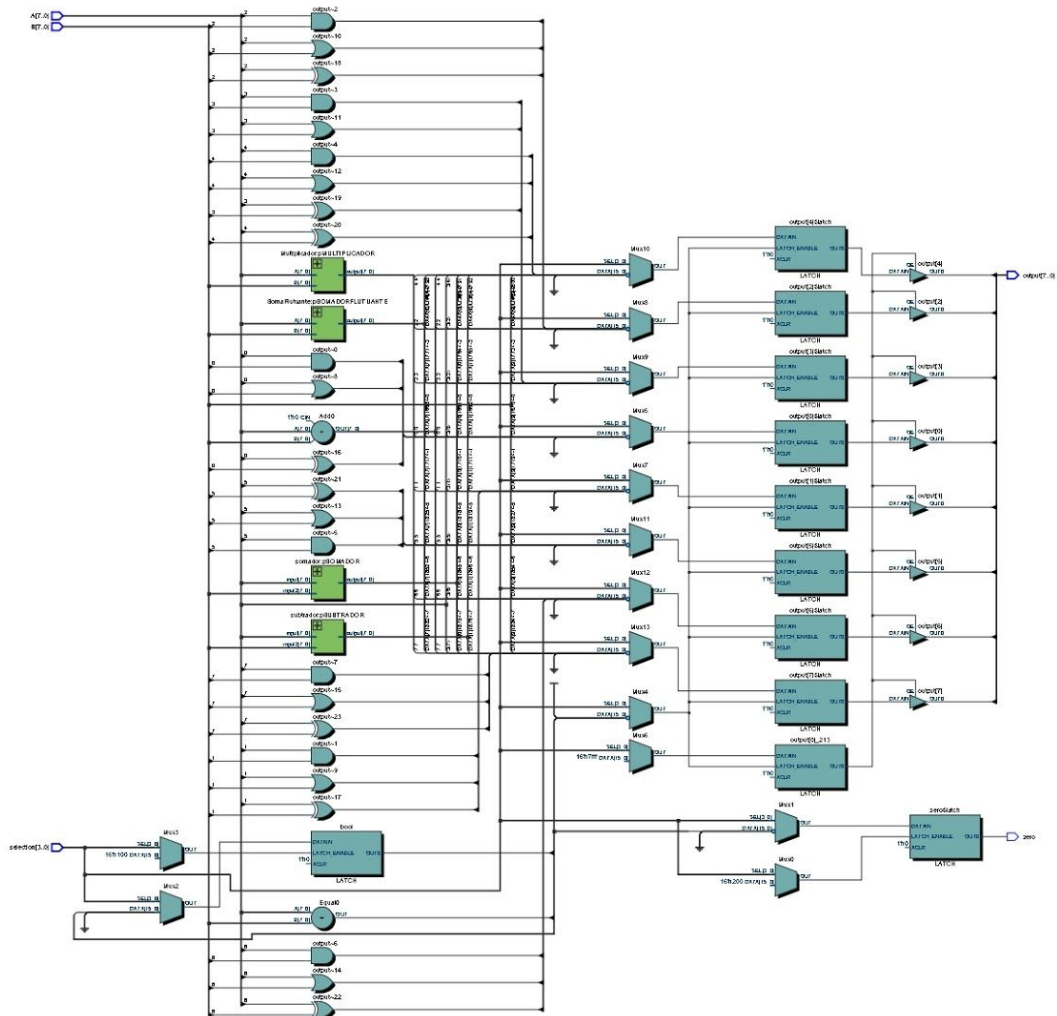


Figura 2 - Bloco simbólico do componente QALU gerado pelo Quartus

1.3.2 BRegister

O banco de registradores funciona da seguinte maneira: Funciona como um seletor de dados, caso o enableWrite esteja em 0, é procurado os dados registrados no endereço_A e endereço_B. Caso o enableWrite esteja em 1, o banco de registrador irá guardar o dado contido no datain no endereço endereço_A.

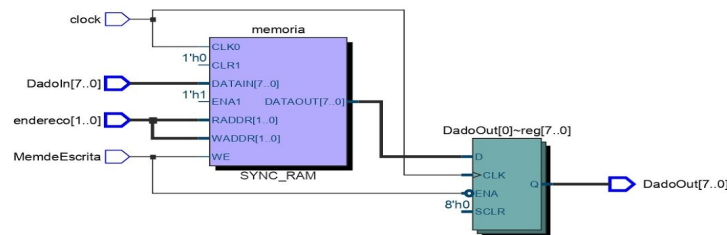


Figura 7 - RTL view do componente Banco de Registradores gerado pelo Quartus.

1.3.3 Multiplexador de 3 entrada

É um componente onde terá diversas entradas e apenas uma saída que será definida através do selector dependendo do seu valor: caso o selector 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. caso o seletor seja "10", a saída do MUX terá o valor de E2. Obs.: todos os casos necessitam do evento de borda alta no ciclo de clock

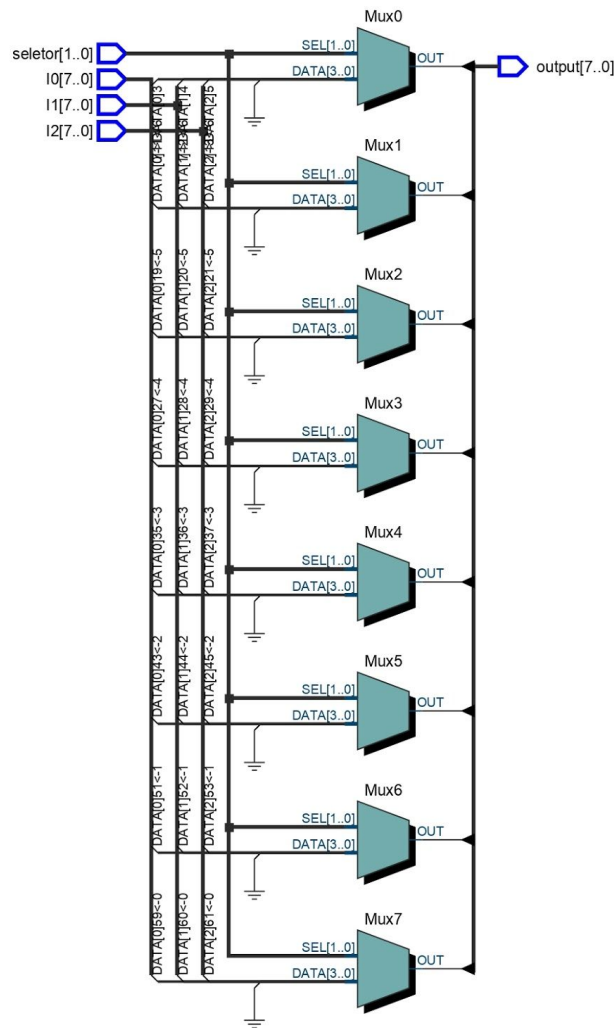


Figura 15 - RTL view do componente Multiplexador de 3 entradas gerado pelo Quartus.

1.3.4 Somador de Ponto Flutuante

O padrão IEEE 754 é uma forma de normalizar as operações com pontos flutuantes.

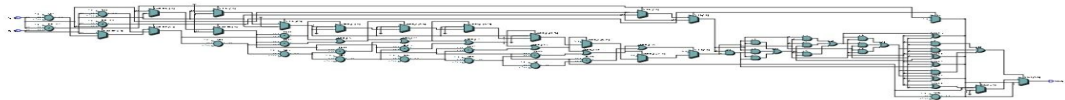


Figura 19 - RTL view do componente Somador de Ponto Flutuante gerado pelo Quartus.

1.3.5 Memória de dados

Basicamente ela funciona da seguinte maneira: quando houver um ciclo de borda alta no **clock** a escrita será habilitada, logo após o dado **datain** é armazenado no **endereço** estabelecido, no próximo ciclo de borda alta em que o **endereço** lido for o estabelecido obteremos o valor contido naquele endereço no **dataout**. Abaixo podemos ver a RTL view do componente especificado.

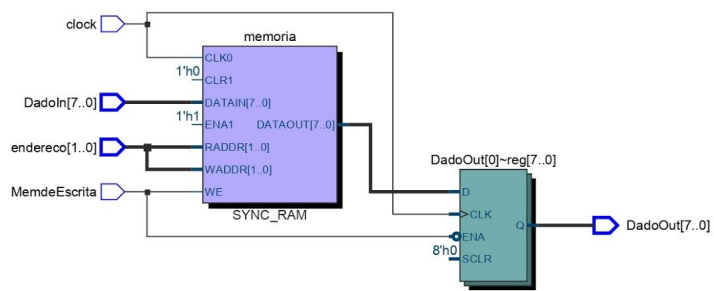


Figura 21 - RTL view do componente Memória de dados gerado pelo Quartus.

1.3.6 Somador

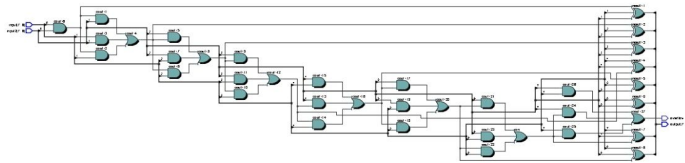


Figura 21 - RTL view do componente Somador gerado pelo Quartus.

1.3.7 Multiplicador

O componente utiliza do algoritmo de booth para executar uma multiplicação através de shifts.

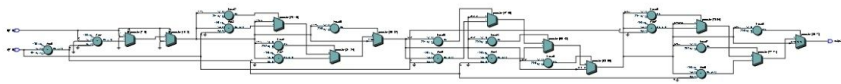


Figura 21 - RTL view do componente Multiplicador gerado pelo Quartus.

1.4 Datapath

É a conexão entre as unidades funcionais formando um único caminho de dados e acrescentando uma unidade de controle responsável pelo gerenciamento das ações que serão realizadas para diferentes tipos de instruções

2 Considerações finais

Este trabalho apresentou o projeto e implementação do processador de 8 bits denominado de MENSIS que é a junção dos conhecimentos adquiridos durante o período de ensino da disciplina Arquitetura e Organização de Computadores. Embora não foi possível finalizá-lo ainda adquirimos grande conhecimento com ela.

O nome surgiu de um personagem da franquia Dark Souls da From Software, especificamente o Spin off Bloodborne, onde uma estudante de byrgenwerth tentava contato com os deuses ocultos através do arcano, em um sucesso ela ganhou o conhecimento capaz de criar um sonho para todos, mas foi amaldiçoada a viver no pesadelo eternamente.