

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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO DO PROJETO: PROCESSADOR ProIA

ALUNOS:

Ibukun chife Didier Adjitche - 2016030686 Pedro Aleph Gomes de Souza Vasconcelos - 2016007150

> Janeiro de 2018 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 ProIA

Janeiro de 2018 Boa Vista/Roraima

Resumo

Esse projeto aborda a implementação de um processador baseado na arquitetura Mips. O processador ProlA apresenta as diversos componentes necessários para que o mesmo funcione e apresenta a característica de ter memórias separadas, ou seja, memória para dados e memória para instruções; sendo estes componentes a ULA, a unidade de controle, memória de dado, memória de instrução, banco de registradores e program counter.

Cada componente tem uma característica especifica, onde a unidade de controle vai enviar os sinais pelas flags para que de acordo com a instrução que deve operar, o PC apresenta a união a um somador, para que o mesmo possa fazer o cálculo da próxima instrução, As memórias servem para armazenar as informações, sendo uma para os dados que serão utilizados e outro para as instruções que devem ser executadas. A ULA apresenta as operações básicas, com exceção da divisão, e não trata os casos de overflow.

Conteúdo

7	Especi	1
olvimento	1.1 P	
s8	1.2	
e9	1.3 C	
9	1.3.1	
10	1.3.2	
11	1.3.3	
11	1.3.4	
os	1.3.5	
ruções13	1.3.6	
Erreur ! Signet non défini.	1.3.7	
Erreur ! Signet non défini.	1.3.8	
Erreur ! Signet non défini.	1.3.9	
14	1.3.10	
Erreur ! Signet non défini.	1.3.11	
	1.4 [
16	Simula	2
22	Consid	3

Lista de Figuras

FIGURA 1 - ESPECIFICAÇÕES NO QUARTUS	7
FIGURA 2 - BLOCO SIMBÓLICO DO COMPONENTE QALU GERADO PELO QUARTUS	
FIGURA 19 - RESULTADO NA WAVEFORM.	ERREUR ! SIGNET NON DEFINI

Lista de Tabelas

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

1 Especificação

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

1.1 Plataforma de desenvolvimento

Para a implementação do processador Prol
A foi utilizado a IDE: Quartus Prime version 17.0.

Flow Status	Successful - Wed Jan 17 16:45:19 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	ProIA
Top-level Entity Name	ProIA
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	225 / 56,480 (< 1 %)
Total registers	141
Total pins	145 / 268 (54 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0/6(0%)
Total HSSI PMA RX Deserializers	0/6(0%)
Total HSSI TX PCSs	0/6(0%)
Total HSSI PMA TX Serializers	0/6(0%)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figura 1 - Especificações no Quartus

1.2 Conjunto de instruções

O processador ProIA possui 4 registradores: \$s0, \$s1, \$s2 e \$s3. Assim como 2 formatos de instruções de 8 bits cada, Instruções do **tipo R (operações aritméticas) Tipo I (load e store)**, 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: instruções baseadas em operações aritméticas.

Formato para escrita de código na linguagem Quantum:

OPCODE	Reg1	Reg2
--------	------	------

Formato para escrita em código binário:

OPCODE	Reg1	Reg2		
4 Bits	2 Bits	2 Bits		
7-4	3-2	1-0		

- Formato do tipo I: instruções de Load e Store.

Formato para escrita de código na linguagem Quantum:

OPCODE	Reg1	Reg2	Endereço
--------	------	------	----------

Formato para escrita em código binário:

OPCODE	Reg1	Endereço		
4 Bits	2 Bits	2 Bits		
7-4	3-2	1-0		

- Instrução do tipo J (Jump)

Formato para escrita de código na linguagem Quantum:

	OPCODE	Endereço
--	--------	----------

Formato para escrita em código binário:

OPCODE		Endereço
4 Bits	4 Bits	
7-4	3-0	

Visão geral das instruções do Processador ProIA:

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

OPCODE	INSTRUÇÃO	NOME	TIPO	EXEMPLO	SIGNIFICADO
0000	ADD	add	r	add \$s0, \$s2,\$s1	\$s0 = \$s2 + \$s1
0001	SUB	sub	r	sub \$s0 , \$s2,\$s1	\$s0 = \$s2 - \$s1
0010	AND	and	r	and \$s0, \$s2,\$s1	\$s0 = \$s2 & \$s1
0011	OR	or	r	or \$s0,\$s2,\$s1	\$s0 = \$s2 \$s1
0100	MULTIPLICAÇÃO	mult	r	mult \$s0,\$s2	(\$s0 * \$s2)
0101	BEQ	beq	i	beq \$s0,\$s2.25	If (\$s0 == \$s2) go to 25
0110	SLT	slt	i	slt \$s0,\$s2,\$s1	if(\$s2 < \$s1) \$s0=1 else \$s0=0
0111	LOAD IM	li	i	li \$s0, 3	\$s0 = 3
1000	LOAD	lw	i	lw \$s0, 4(\$s2)	\$s0 =Memoria[\$s2+4]
1001	STORE	SW	i	sw \$s0 , 4 (\$s2)	Memoria[\$s2+4] = \$s0
1010	JUMP	jump	j	jump 1111	Go to 1111
1111	*EXIT			Ele serve de	
				encerramento ao	
				um programa. e	
				geralmente é de	
				tipo 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 ALU (Q 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 ALU efetua operações de comparação de valor como maior ou igual, menor ou igual, somente maior, menor ou igual. O componente ALU 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. O QALU também possui duas saídas: zero – identificador de resultado (2bit) para comparações (1 se verdade e 0 caso contrário);e result – saída com o resultado das operações aritméticas; O processador não trata os casos de Overflow, os valores que ultrapassam sua capacidade de 8bits.

Commenté [h1]: Um example da descrição

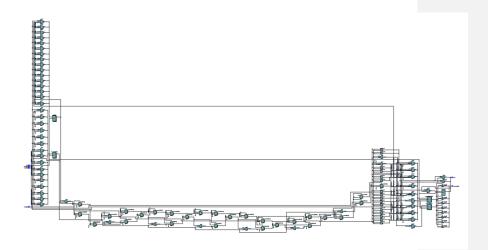


Figura 2 - Bloco simbólico do componente QALU gerado pelo Quartus

1.3.2 Banco de Registradores 8bit

O banco de registradores é um componente digital composto por um conjunto de registradores que podem ser acessados de forma organizada. De uma maneira geral, podem ser executadas operações de leitura dos dados anteriormente gravados e de escrita de dados para modificar as informações internas. As informações que estão sendo processado em um determinado momento devem estar armazenadas no banco

de registradores. Possuindo assim: dado - com 8 bits responsável pelo o armazenamento de dados; saida - possui 8 bits e mostra o resultado.

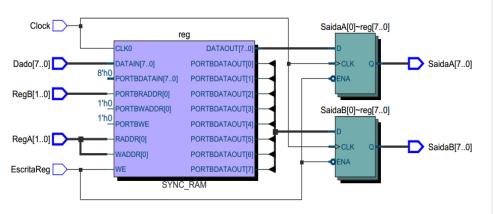


Figura 3 - Bloco simbólico do componente banco de registradores gerado pelo Quartus

1.3.3 Clock

O clock é responsável por coordenar ações de dois ou mais componentes. Um sinal de clock oscila entre os estados alto e baixo (no estado de 1 e 0), normalmente usando um para definir algum tipo de operação.

1.3.4 Unidade de Controle

A unidade de controle é responsável por enviar os comandos para as demais unidades. Estes comandos são necessários para a execução de uma determinada instrução. Para isso este componente tem como entrada o Opcode responsável pela instrução, ou seja, para a instrução de soma o opcode que será entrada da unidade de controle será o 0000. Com a entrada do opcode é ativado as flags de saída e com isso os demais componentes serão "configurados" para que a operação de soma seja executada.

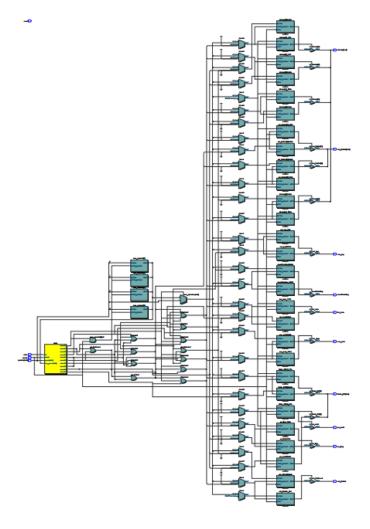


Figura 4 - Bloco simbólico da unidade de controle gerado pelo Quartus

Abaixo segue a tabela, onde é feita a associação entre os opcodes e as flags de controle:

Tabela 2 - Detalhes das flags de controle do processador.

Comando	PcFlag	PcCond	EscReg	LerMem	EscMem	MemparaReg	FonteA	FonteB	PCFonte	ULAOP
Add	0	0	1	0	0	0	1	00	00	0000
Sub	0	0	1	0	0	0	1	00	00	0001
And	0	0	1	0	0	0	1	00	00	0010
Or	0	0	1	0	0	0	1	00	00	0011
Mult	0	0	1	0	0	0	1	00	00	0100
BEQ	0	1	0	0	0	1	1	00	01	1010
SLT	0	1	0	0	0	1	1	00	01	1011
LOADIM	0	0	1	0	0	1	1	11	00	0111
LOAD	0	0	1	0	0	1	1	10	00	1000
STORE	0	0	0	0	1	1	1	10	00	1001

JUMP	1	0	0	0	0	1	0	01	10	0000	l
EXIT	Z	Z	Z	Z	Z	Z	Z	ZZ	ZZ	ZZZZ	ı

1.3.5 Memória de dados

A memória de dados é usada para pelo processador para armazenar os arquivos e programas que estão sendo processada. A principal característica da memória de dados é que ela é volátil, ou seja, os dados se perdem ao reiniciar o computador. Ao ligar é necessário refazer todo o processo de carregamento, em que o sistema operacional e aplicativos usados são transferidos do HD para a memória, onde podem ser executados pelo processador. A memória de dados possui os seguintes componentes: clock - possui 1 bits e atua como um sinal para sincronismo; reset - 1 bits e usado para reiniciar; leitura - 1 bits responsável pela leitura do dado; escrita - 1 bits recebe os dados escritos; endereço – 8 bits ele pode armazenar dados ou buscar um determinado valor, depende da instrução; dado - 8 bits responsável pelo o armazenamento de dados ; saída - 8 bits, onde mostra o valor sinal da operação.

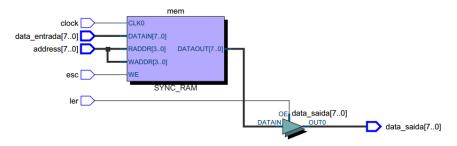


Figura 4 - Bloco simbólico do componente da memória de dados gerado pelo Quartus

1.3.6 Memória de Instruções

A memória de instruções é um termo genérico usado para designar as partes do computador ou dos dispositivos periféricos onde os dados e programas são armazenados. Sem uma memória de onde os processadores podem ler e escrever informações, não haveria nenhum computador digital de programa armazenado. Sendo formada por: 2 registradores - com 2 bits responsável pelo conjunto de dados armazenados; opcode - possui 4 bits para conseguir realizar determinadas tarefas; endereço - com 8 bits para armazenar dados.

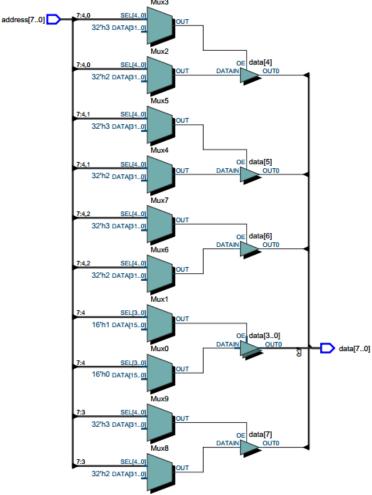


Figura 5 - Bloco simbólico do componente da memória de instruções gerado pelo Quartus.

1.3.7 PC

O pc (Program Counter) é a unidade responsável por armazenar instruções a ser executada, além disso pode ser feita algumas operações para descobrir a próxima instrução, pois tem um somador integrado. Sendo assim: clock - possui 1 bits e atua como um sinal para sincronismo; reset - 1 bits e usado para reiniciar; flag - 1 bits e usado como interruptor (isto é, valores 1/0, ligado/desligado, ativo/inativo) permite otimizar as estruturas de dados; entrada - 8 bits e uma informação que é recebida e processada; saída - 8 bits mostra a saída do dado processado; saida2 - 8 bits mostra a saída do dado processado.

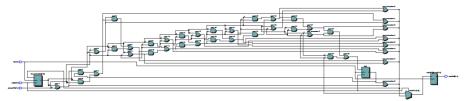


Figura 7 - Bloco simbólico do componente da pc gerado pelo Quartus

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 classes de instruções.

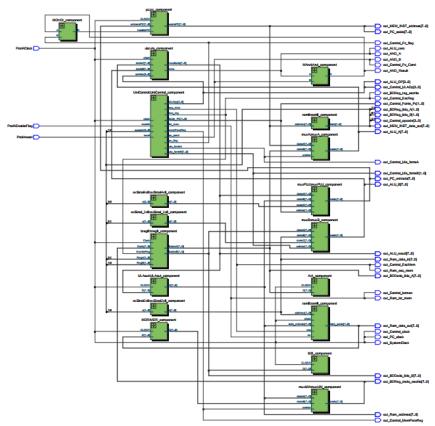
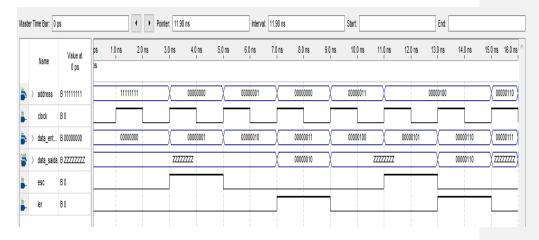


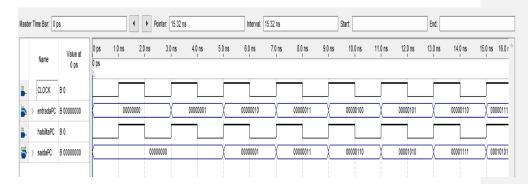
Figura 8 - Bloco simbólico do componente da Datapath gerado pelo Quartus

2 Simulações e Testes

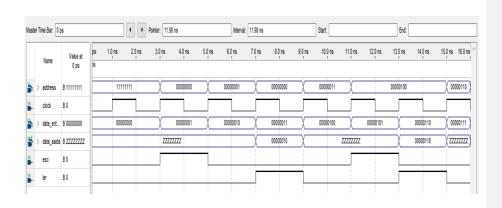
Objetivando analisar e verificar o funcionamento do processador, efetuamos alguns testes analisando cada componente do processador em especifico, em seguida efetuamos testes de cada instrução que o processador implementa. Para demonstrar o funcionamento do processador ProIA, os testes dos seguintes componentes forma realizados.



Componente Banco de registradores 8bit



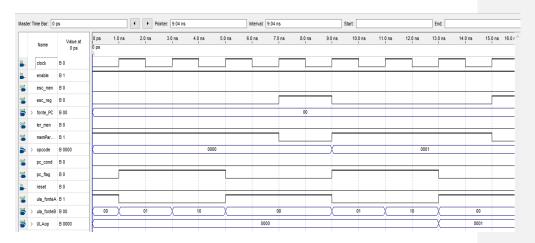
Componente PC



Componente memória Ram 8bit

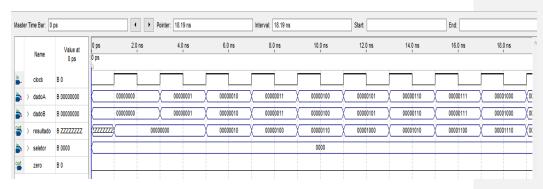


Componente memória Rom 8bit

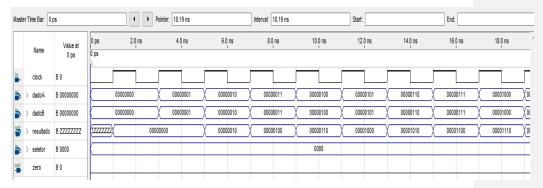


Componente Unidade de Controle

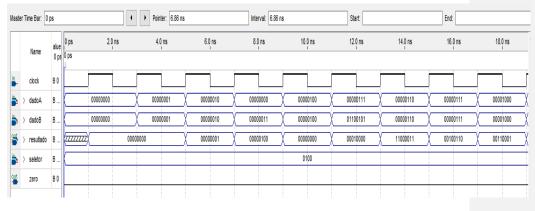
A seguir todos os testes na ULA...



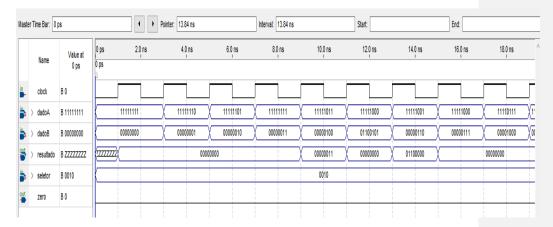
ULA-SOMA



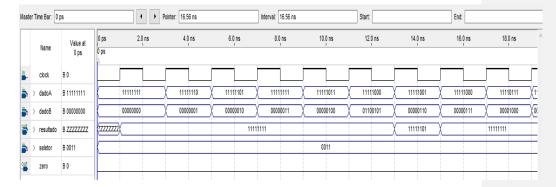
ULA-SUBTRAÇÃO



ULA-MULTIPLICAÇÃO



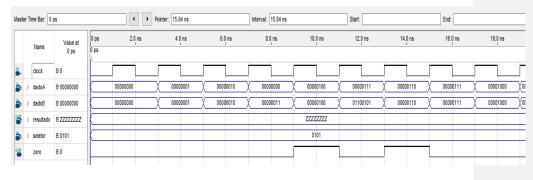
ULA-AND



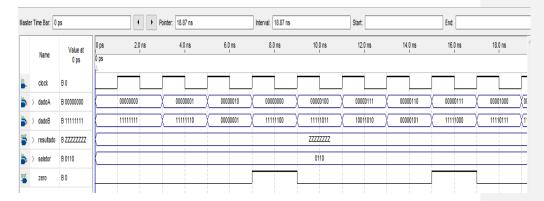
ULA-OR



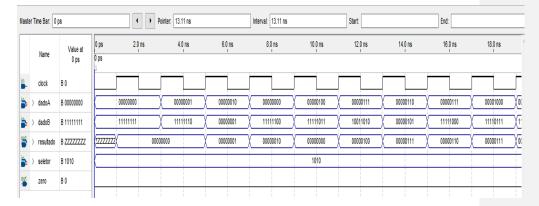
ULA-STORE



ULA-BEQ



ULA-SLT



ULA-JUMP

3 Considerações finais

Este trabalho apresentou o projeto e implementação do processador de 8 bits denominado de ProIA como trabalho final da disciplina Arquitetura e Sistema de Computadores, com o objetivo de demostrar o que aprendemos ao longo da disciplina na prática, o funcionamento como um todo com seus componentes, o que fazem e como funcionam.