

**PODER EXECUTIVO**

**MINISTÉRIO DA EDUCAÇÃO**

**UNIVERSIDADE FEDERAL DE RORAIMA**

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

**ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

**RELATÓRIO DO PROJETO: PROCESSADOR KRAKEN**

**ALUNOS:**

**DÉBORA BIANCA TAVEIRA DE MOURA – 2016011555**

**EWELLY FABIANE CUNHA DE SOUSA – 2016011439**

**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 KRAKEN**

**Janeiro de 2018**

**Boa Vista/Roraima**

**Resumo**

Este relatório aborda o projeto e a implementação do processador de 8 bits **Kraken**, da disciplina de Arquitetura e organização de computadores, desenvolvido pelas alunas Débora Bianca e Ewelly Fabiane. Neste documento estarão descritos os componentes do processador e seu respectivo datapath, o formato de suas instruções, baseado no aprendizado da disciplina.

O nome do processador faz analogia ao **Kraken** da mitologia nórdica, sendo 8 de seus tentáculos referência aos 8 bits do processador. Os seus tentáculos têm a intenção de capturar alimento, assim como nosso processador captura suas oito instruções, mergulhado em um mar de binários.

**Conteúdo**

[1 Especificação 7](#_Toc444681789)

[1.1 Plataforma de desenvolvimento 7](#_Toc444681790)

[1.2 Conjunto de instruções](#_Toc444681791) 7

[1.3 Descrição do Hardware](#_Toc444681792) 8

[1.3.1 ALU ou ULA](#_Toc444681793) 8

[1.3.2 BDRegister 9](#_Toc444681794)

[1.3.3 Clock 9](#_Toc444681795)

[1.3.4 Controle 9](#_Toc444681796)

[1.3.5 Memória de dados 10](#_Toc444681797)

[1.3.6 Memória de Instruções 10](#_Toc444681798)

[1.3.7 Somador 10](#_Toc444681799)

[1.3.8 And 10](#_Toc444681800)

[1.3.9 Mux\_2x1 10](#_Toc444681801)

[1.3.10 PC 10](#_Toc444681802)

[1.3.11 ZERO 11](#_Toc444681803)

[1.4 Datapath 11](#_Toc444681804)

[2 Simulações e Testes 13](#_Toc444681805)

[3 Considerações finais 14](#_Toc444681806)

**Lista de Figuras**

[Figura 1 - Especificações no Quartus 6](#_Toc444681815)

[Figura 2 - Bloco simbólico do componente QALU gerado pelo Quartus 8](#_Toc444681816)

[Figura 19 - Resultado na waveform. 13](#_Toc444681817)

**Lista de Tabelas**

[Tabela 1 – Tabela que mostra a lista de Opcodes utilizadas pelo processador XXXX. 7](#_Toc444681822)

[Tabela 2 - Detalhes das flags de controle do processador. 9](#_Toc444681823)

[Tabela 3 - Código Fibonacci para o processador Quantum/EXEMPLO. 12](#_Toc444681824)

# Especificação

Nesta seção é apresentado o conjunto de itens para o desenvolvimento do processador **Kraken**, bem como a descrição detalhada de cada etapa da construção do processador e a definição de seus componentes.

## Plataforma de desenvolvimento

Para a implementação do processador **Kraken** foi utilizado a IDE Quartus Prime 17.1 Lite, e a linguagem utilizada foi VHDL.

## Conjunto de instruções

O processador Kraken possui 2 registradores: S0, S1. Assim como três formatos de instruções de 8 bits cada, Instruções do **tipo i (transferência), r (Aritmético), j (Lógico)**, seguem algumas considerações sobre as estruturas contidas nas instruções:

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

**TIPO DE INSTRUÇÃO:**

* **Formato do tipo R (Aritmético):** Esse formato aborda instruções de adição, subtração,

|  |  |  |
| --- | --- | --- |
| **Tipo R** | **Reg1** | **Reg2** |
| Opcode | rs | rt |
| 4 bits | 2 bits | 2 bits |

* **Formato do tipo I (Transferência)**: Este formato aborda instruções de Load, Store e instruções baseadas em operações aritméticas.

|  |  |  |
| --- | --- | --- |
| **Tipo I** | **Reg1** | **Reg2** |
| Opcode | rs | rt |
| 4 bits | 2 bits | 2 bits |

* **Formato do tipo J (Jump)**: Este formato aborda instruções de desvio incondicional, como por exemplo o Jump.

|  |  |
| --- | --- |
| **Tipo J** | **Reg2** |
| Opcode | rt |
| 4 bits | 4 bits |

**Visão geral das instruções do Processador Kraken:**

O número de bits do campo **Opcode** das instruções é igual a quatro, sendo assim obtemos um total () de 16 **Opcodes (0-3)**, entretanto, apenas 8 são distribuídos entre as instruções, assim como é apresentado na Tabela a seguir.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opcode** | **Nome** | **Formato** | **Breve Descrição** | **Exemplo** |
| **0000** | add | R | Soma | **add** $s1, $s1,$s2 |
| **0001** | sub | R | Subtração | **sub** $s1, $s1, $s2 |
| **0010** | mult | R | Multiplicação | **mult** $s1,$s2 |
| **0011** | lw | I | Load | **lw** $s1,100 |
| **0100** | sw | I | Store | **sw** $s1,100 |
| **0101** | j | J | Jump | **j** 100 |
| **0110** | beq | I | Beq | **beq** $s1, 100 |
| **0111** | and | R | And | **and** $s1, $s1,$s2 |

*Obs.: A instrução de beq receberá em seu registrador o valor lógico da instrução* ***and****, sendo ele decisivo para a realização do desvio, já que nosso processador possui apenas dois registradores a comparação de dois registradores e o salto na mesma instrução era inviável, logo a instrução foi dividida em duas.*

## Descrição do Hardware

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

### ALU ou ULA

O componente ALU (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 QALU 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 ALU também possui três saídas: **zero** – identificador de resultado (2bit) para comparações (1 se verdade e 0 caso contrário); e **result** – saída com o resultado das operações aritméticas.

### Banco de Registradores

**[Todo] Descrição**

### PC (Program Counter)

**[Todo] Descrição**

### 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:

* **DvC**: XXXX.
* **en\_data**: XXXX.
* **EscMem**: XXXX.
* **MemParaReg**: XXXX.
* **UlaOp**: XXXX.
* **LwSwOp**: XXXX.
* **EscReg:** XXXX.
* **Wrt\_LRT**: XXXX.
* **FlagPC**: XXXX.

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** | **DvC** | **off\_instruction** | **en\_data** | **Esc**  **Mem** | **Mem**  **ParaReg** | **UlaOp** | **LwSw**  **Op** | **EscReg** | **Wrt\_LRT** | **FlagPc** | **enJmp** |
| **add** | 0 | 1 | 1 | 0 | 1 | 0000 | 0 | 1 | 0 | 1 | 0 |
| **sub** | 0 | 1 | 1 | 0 | 1 | 0001 | 0 | 1 | 0 | 1 | 0 |
| **div** | 0 | 1 | 1 | 0 | 1 | 0111 | 0 | 1 | 0 | 1 | 0 |
| **Inicialização** | 0 | 1 | 1 | Z | Z | ZZZZ | Z | Z | Z | 1 | 0 |

### Memória de dados

**[Todo] Descrição**

### Memória de Instruções

**[Todo] Descrição**

### Somador

**[Todo] Descrição**

### Controle da ULA

**[Todo] Descrição**

### Multiplexador 2x1

**[Todo] Descrição**

### Extensor de sinal (2 para 8 bits)

**[Todo] Descrição**

### Extensor de sinal (2 para 4)

**[Todo] Descrição**

## 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...

**1.4.1 Datapath Idealizado**

# Simulações e Testes

As simulações e testes não foram realizados pois seu datapath não foi codificado. Seu teste seria realizado com calculo de um número fatorial e com a verificação de um número par ou impar, assim, utilizando todas as instruções solicitadas na criação do processador.

# Considerações finais

Este trabalho apresentou o projeto do processador de 8 bits denominado de Kraken, que apesar de não ter sido implementado, testado e finalizado, ele contribuiu fortemente para o entendimento da disciplina de Arquitetura e Organização de Computadores e a formação acadêmica da dupla desenvolvedora, nos ensinando a base de um processador de modo geral, seu funcionamento, suas necessidades, além de demonstrar possíveis situações que podem ocorrer durante o desenvolvimento ou estudo de um processador.