



UFRR

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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO DO PROJETO: PROCESSADOR_KBC

**ALUNOS: Luiz Gustavo das Chagas Lima - 1201524436
Lucas Daniel Miranda Freitas - 2018005584**

**28 de Novembro de 2019
Boa Vista/Roraima**



UFRR

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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO DO PROJETO: PROCESSADOR_KBC

**28 de novembro de 2019
Boa Vista/Roraima**

Resumo

Esse trabalho aborda um projeto de processador chamado Procesador_KBC com arquitetura de 8 (oito) bits escrito em na linguagem VHDL na estrutura do programa Quartus da Intel.

Será descrito com detalhes todos os componentes importantes do processado que são necessários para o funcionamento adequado e o seus testes mostrando a eficácia do funcionamento desses mesmos componentes.

Conteúdo

1	Especificação	7
1.1	Plataforma de desenvolvimento	7
1.2	Conjunto de instruções	7
1.3	Tpos de instrucoes	8
1.3.1	Visao geral das instrucoes	8
1.3.2	Descricao do hardware	9
1.3.3	Memória de instrucao	9
1.3.4	Extensor de sinal de 2 para 8	10
1.3.5	Extensor d sinal de 4 para 8	10
1.3.6	registrador do tipo Flip-Flop	10
1.3.7	Banco de registradores	10
1.3.8	Multiplexador	10
1.3.9	Mux_2x1	10
1.3.10	Mux_3x1	10
1.3.11	ULA	11
1.3.12	Memória de dados	11
1.3.13	Unidade de controle	11
1.4	Datapath	12
2	Simulações e Testes	12
3	Considerações finais	13

Lista de Figuras

FIGURA 1 - ESPECIFICAÇÕES NO QUARTUS

7

Lista de Tabelas

TABELA 1 – TABELA QUE MOSTRA A A SEPARACAO DOS BITS PARA A INSTRUCAO DO TIPO R	8
TABELA 2 - TABELA QUE MOSTRA A A SEPARACAO DOS BITS PARA A INSTRUCAO DO TIPO J	8
TABELA 3 – TABELA QIE MOSTRA LISTA DE OPCODE UTILIZADOS PELO PROCSSADOR_KBC	8

1. Especificação

Nesta seção é apresentado o conjunto de itens para o desenvolvimento do Processador_KBC 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_KBC foi utilizado a IDE Quartus Prime, versão 15.1, com o simulador ModelSim-Altera versão 15.1.0.185 e a linguagem usada para a devida programação é VHDL

Flow Status	Successful - Mon Aug 27 17:33:50 2012
Quartus II Version	9.0 Build 184 04/29/2009 SP 1 SJ Web Edition
Revision Name	Quantum
Top-level Entity Name	Quantum_Pro
Family	Cyclone III
Met timing requirements	N/A
Total logic elements	3,123 / 15,408 (20 %)
Total combinational functions	2,148 / 15,408 (14 %)
Dedicated logic registers	2,137 / 15,408 (14 %)
Total registers	2137
Total pins	188 / 347 (54 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	1 / 112 (< 1 %)
Total PLLs	0 / 4 (0 %)
Device	EP3C16F484C6
Timing Models	Final

Figura 1 - Especificações no Quartus

1.2 Conjunto de instruções

- ❑ O processador_KBC possui 4 registradores junto ao banco de registradores: S0, S1, S2, S3. Assim como duas instruções de 8 bits, (do tipo **R** e **J**), seguem algumas considerações sobre as estruturas contidas nas instruções:

- **Opcode:** código de operação, realiza operação básica a ser executada pelo processador;
- **RS:** é o registrador que contém o primeiro operando fonte e adiciona para alguns tipos de instruções que é o registrador de destino;
- **RT:** o registrador contendo o segundo operando fonte;
- **IMM:** endereço de memória, para instruções do tipo jump condicional e incondicional.

1.3 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_KBC:

Tabela 1 - Tabela que mostra a separação dos bits para instruções do tipo R.

OPCODE	RS	RT
7 – 4	3 – 2	1 – 0
4	2	2

Formato para escrita em código binário:

Tabela 2 - Tabela que mostra a separação dos bits do tipo J.

APCODE	IMM
7 - 4	3 – 0
4	4

Visão geral das instruções do Processador_KBC:

O número de bits do campo **Opcode** das instruções é igual a 4 (quatro), sendo assim obtemos um total ($Bit(0 \text{ e } 1)^4 \therefore 2^4 = 16$) de 16 **Opcodes (0-16)** que são distribuídos entre as instruções, assim como é apresentado na Tabela 5.

Tabela 3 - Tabela que mostra a lista de Opcodes utilizadas pelo processador_KBC

Tipo	OP	RS	RT	IMM	Exemplo	Instrução
R	0000	00-11	00-11	-	add \$s1, \$s2	add
R	0001	00-11	00-11	-	addi \$s1, WORD	addi

R	0010	00-11	00-11	-	sub \$s1, \$s2	sub
R	0011	00-11	00-11	-	subi \$s1, WORD	subi
R	0100	00-11	00-11	-	mult \$s1, \$s2	mult
R	0101	00-11	00-11	-	multi \$s1, WORD	multi
R	0110	00-11	00-11	-	eq \$s1, \$s2	eq
R	0111	00-11	00-11	-	eqi \$s1, WORD	eqi
R	1000	00-11	00-11	-	move \$s1, \$s2	move
R	1001	00-11	00-11	-	movi \$s1, WORD	movi
R	1010	00-11	00-11	-	lw \$s1, WORD	lw
R	1011	00-11	00-11	-	sw \$s1, WORD	sw
R	1100	00-11	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

1.4 Descrição do Hardware

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

1.4.1 Memória de Instrução

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

1.4.2 Extensor de Sinal de 2 para 8 bits

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.

1.4.3 Extensor de sinal de 4 para 8

O extensor de sinal de 4 para 8 bits funciona da seguinte maneira: A entrada input com 4 bits será convertida para uma saída output com 8 bits.

1.4.4 Registrador do tipo Flip-Flop

O registrador flip flop funciona da seguinte maneira: Caso o clock esteja em borda alta e enable possuir valor 1, o valor da entrada input é registrado e o output recebe o input.

1.4.5 Banco de Registradores

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 através do demultiplexador.

1.4.6 Multiplexador de duas entradas

O multiplexador de 2 entradas funciona da seguinte maneira: É um componente onde terá 2 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 selector seja "01", a saída do MUX terá o valor de E1. Obs.: todos os casos necessitam do evento de borda alta no ciclo de clock.

1.4.7 Multiplexador de três entradas

O multiplexador de 2 entradas funciona da seguinte maneira: 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

1.4.8 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 duas saídas: zero – identificador de resultado (2bit) para comparações (1 se verdade e 0 caso contrário); e result – saída com o resultado das operações.

1.4.9 Memória de Dados

Podemos ver acima um trecho do código da memória de dados que demonstra todas as entradas e saídas. 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.

1.4.10 Unidade de Controle

Acima podemos ver um trecho do código da Unidade de Controle do processador NEO. A unidade de controle é componente responsável por ativar as flags e manter o fluxo correto das instruções, podendo suportar instruções do tipo R e J, e capaz de executar os estados básicos do mips, são eles: busca, decodificação, execução, cálculo de memória e escrita no registrador. Abaixo podemos ver na **Figura 23** a RTL view e na **Figura 24** a máquina de estados da unidade de controle.

1.5 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. Para o Processador_KBC 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.

2. Simulações e Testes

Objetivando analisar e verificar o funcionamento do processador, efetuamos alguns testes analisando cada componente do processador em específico, em seguida efetuamos testes de cada instrução que o processador implementa. Para demonstrar o funcionamento do processador_KBC, utilizaremos como exemplo o código para calcular o N-ésimo termo de uma P.A, o fatorial de 3 e a soma de números de ponto flutuante. Os testes podem ser verificados logo abaixo.

3. Considerações finais

Este trabalho apresentou o projeto e implementação do processador de 8 bits denominado de Processador_KBC 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, foram suficientes para concluir a construção do processador com êxito.

O nome escolhido para o processador faz referência a nosso grupo do semestre que tem sempre feito os trabalhos da UFRR. O processador_KBC pode nos proporcionar uma experiência incrível n nossa vida acadêmica e aos 49 do segundo tempo, o nome KBC foi escolhido por deixar essa marca.

