



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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES
RELATÓRIO DO PROJETO: PROCESSADOR NBZA_2521 8 BITS

ALUNOS:
BÁRBARA ZAMPERETE OLIVEIRA – 20172486
NATALIA RIBEIRO DE ALMADA – 2017009364

Novembro de 2019
Boa Vista/Roraima



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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES
RELATÓRIO DO PROJETO: PROCESSADOR NBZA_2521 8 BITS

Novembro de 2019
Boa Vista/Roraima

Resumo

Este trabalho aborda o projeto e implementação dos componentes um processador de 8 bits uniciclo/monocycle, que nomeamos NBZA_2521

Conteúdo

- 1 Especificação6
 - 1.1 Plataforma de desenvolvimento
 - 1.2 Conjunto de instruções
 - 1.3 Descrição do Hardware
 - 1.3.1 ALU ou ULA
 - 1.3.2 BDRegister
 - 1.3.3 Clock9
 - 1.3.4 Memória de dados
 - 1.3.5 Memória de Instruções
 - 1.3.6 Somador
 - 1.3.7 And
 - 1.3.9 Mux_2x1
 - 1.3.10 PC11
 - 1.3.11 ZERO12
 - 1.4 DATAPATH
- 2 Considerações finais

Lista de Figuras

Figura 1- ULA

Figura 2- Registrador

Figura 3- RAM

Figura 4- Somador

Figura 5- AND

Figura 6- Multiplexador

Figura 7- PC

1 Especificação

Nesta seção serão apresentados os itens para o desenvolvimento do processador e a descrição de cada etapa da construção do processador.

1.1 Plataforma de desenvolvimento

Para a implementação do processador NBZA_2521 foi utilizada a IDE: Quartus Prime 18.1 Lite Edition

1.2 Conjunto de instruções

O processador NBZA_2521 possui os registradores S0 e S1, que combinam-se a 3 formatos de instruções do tipo R, I e J de 8 bits:

Opcode: operação básica executada pelo processador.

Reg1: o registrador contendo o primeiro operando, é o registrador de destino;

Reg2: o registrador contendo o segundo operando, é a fonte;

Funct: formato para escrita em código binário:

Tipo de Instruções:

Formato do tipo R: Instruções de operações aritméticas.

OPCODE	R1	R2	FUNCT
3 BITS	1 BIT	1 BIT	3 BITS
7-5	4	3	2-0

Formato do tipo I: Instruções de Load, Store, Load Immediately e Branch.

OPCODE	R1	FUNCT
3 BITS	1 BIT	4 BITS
7-5	4	3-0

Formato do tipo J: Instrução do tipo Jump.

OPCODE	ENDEREÇO
3 BITS	5 BITS
7-5	4-0

Visão geral das instruções do Processador NBZA_2521:

São 4 bits do campo das instruções Opcod, então:
(*Bit(0e1)NumeroTotaldeBitsdoOpcode: 2X = X*)) totalizam 8 Opcodes que preenchem de 0-7

OPCODE	Nome	Formato	Breve Descrição	Exemplo
000000	ADD	R	Soma	add \$S0,\$S1
000001	SUB	R	Subtração	sub \$S0,\$S1
000100	MULT	R	Multiplicação	mult \$S0,\$S1
001	LW	I	Load	lw \$S0,5
110	LI	I	Load immediately	li \$S0,20
010	SW	I	Store	sw \$S0,3
011	BNE	I	Branch Not Equal	bne \$S0,\$S1,LABEL
100	BEQ	I	Branch Equal	beq \$S0,\$S1,LABEL
111	JUMP	J	Jump	jump Endereço

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 de números inteiros, e também efetua operações de comparação de valor como igual ou diferente. A ULA recebe três valores:

a - dado de 8bits para operação,

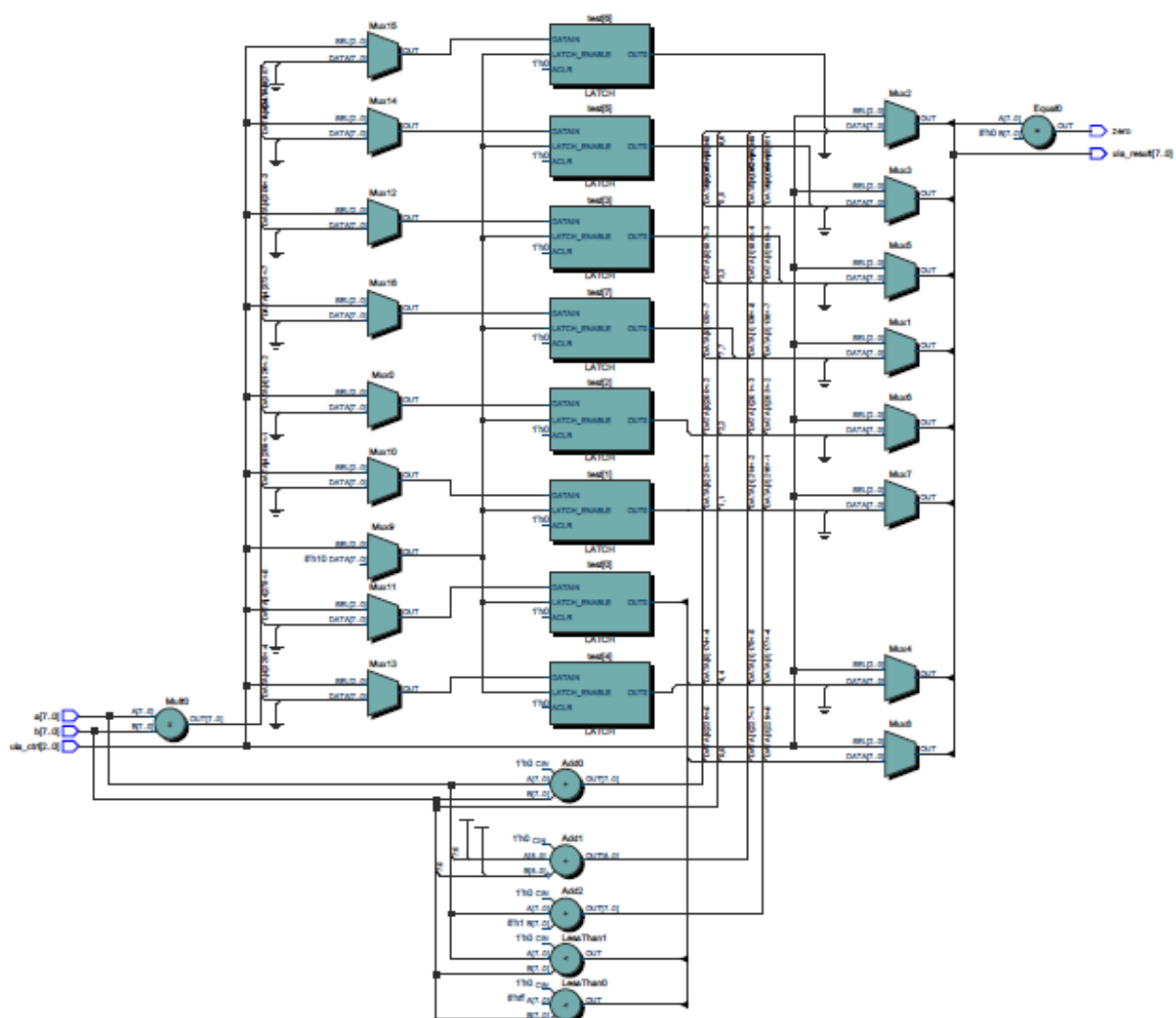
b - dado de 8bits para operação;

ula_controle - identificador da operação que será realizada de 3bits.

A ULA também possui duas saídas:

zero - identificador de resultado (1bits) para comparações (1 se verdade e 0 caso contrário);

Figura1- ULA



1.3.2 Banco de Registradores

O BRegister guarda valores e/ou resultados das operações realizadas pela ULA.

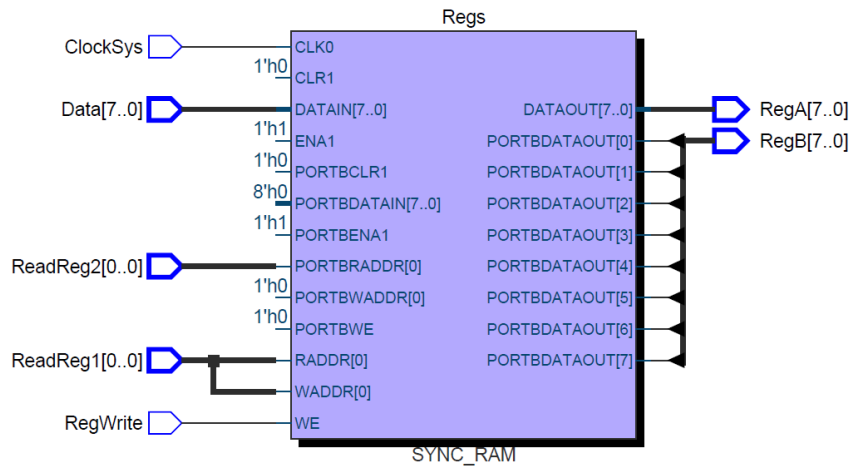


Figura 2 – Banco de Registradores

1.3.3 Clock

Serve para manter os componentes funcionando apenas quando estiver ligado, portanto se for True/1

1.3.5 Memória de dados ou RAM

Armazena os valores na memória e os disponibiliza para serem salvos nos registradores. Tendo como entrada dado_entrada (que define o que vai ser salvo na memória); endereco - o endereço (onde o valor será salvo na memória); EscMem - flag que define se algo vai ser escrito na memória ou não; LeMem - flag que define se algo vai ser lido da memória e passada para o registrador; clk - clock. E ele retorna com o que foi lido na memória - dado_saida.

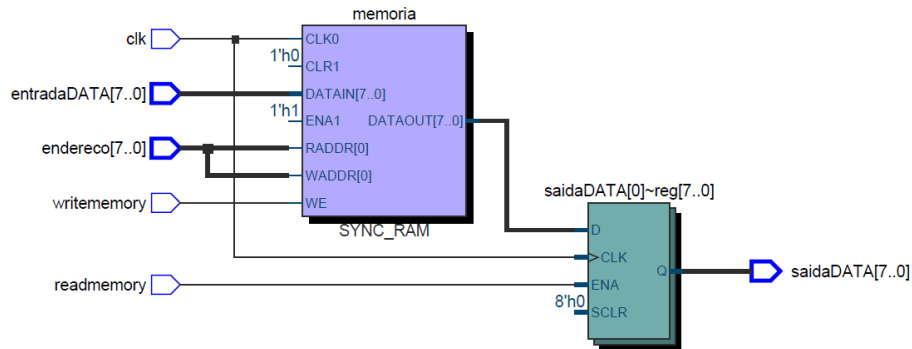


Figura 3 - RAM

1.3.6 Memória de Instruções ou ROM

Onde todas as instruções serão executadas e é a partir dela que o opcode vai para a unidade de controle e que sabemos quais registradores usar. A ROM recebe um endereço que vem do PC e executa o que está nesta instrução.

1.3.7 Somador

Soma cada PC para que ele execute as instruções

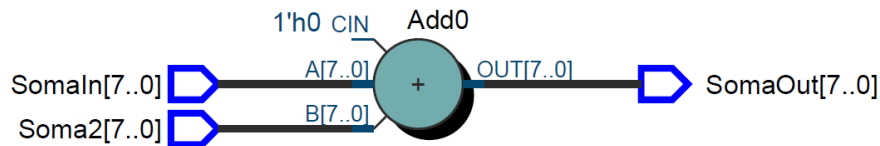


Figura 4 - Somador

1.3.8 And

Verifica se vai ou não ocorrer um branch

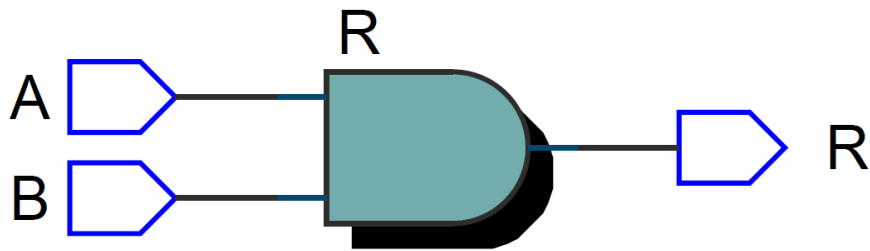


Figura 5 - AND

1.3.9 Mux_2x1

O multiplexador auxilia a decidir que trilha será usada dependendo do valor da flag que está recebendo.

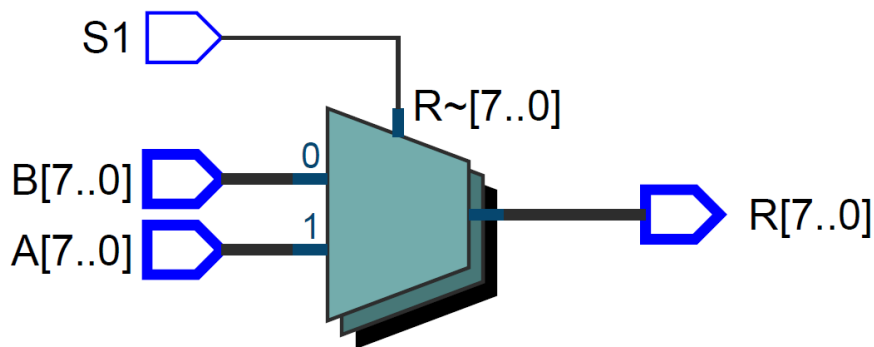


Figura 6 – Multiplexador

1.3.10 PC

Responsável em mandar o endereço para a próxima instrução para a memória de instruções.

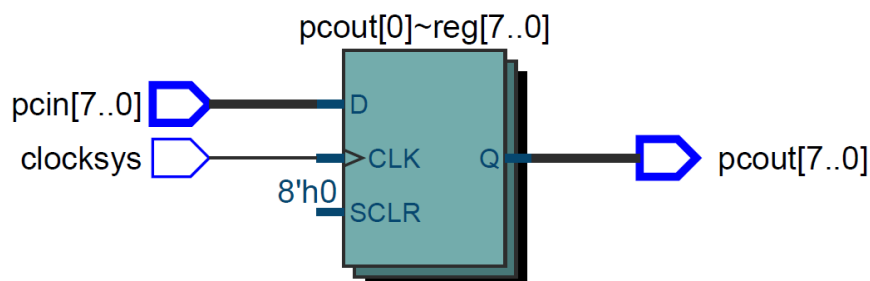


Figura 7 - PC

1.3.11 ZERO

Identificador de resultado (1bit) para comparações (1 se verdade e 0 se falso).

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.

2 Considerações Finais

Este trabalho apresentou o projeto e implementação do processador unicycle de 8 bits referente ao projeto final de disciplina de Arquitetura e Organização de computadores. O processador contém todos os requisitos para que tenha essa alcunha pois contém operações aritméticas, memória de dados, condicionais, controle de dados e afins.