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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

ALUNOS:
Nataly Almeida dos Santos

Novembro / 2022
Boa Vista/Roraima

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

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

RELATÓRIO

Novembro/2022
Boa Vista/Roraima

Resumo

Este trabalho aborda as soluções e metodos feitos para o desenvolvimento da atividade avaliativa, utilizando do software Quartus Prime, da linguagem Assembly MIPS e da linguagem VHDL (Very High-Speed Integration Circuit HDL).

Os componentes em questão servirá de base para o nosso estudo e aplicação no processador 8bits e para avaliação da disciplina de Arquitetura e Organização de Computadores (AOC), ministrada pelo professor Herbert Oliveira Rocha.

Sumário

1. Registrador Flip Flop tipo D

Esse registrador é descrito por armazenar o bit de entrada, possuindo-a diretamente ligada a saída quando o clock se altera, independetemente se D assumir 1 ou 0.

D	Q	**Q***
0	0	0
0	1	0
1	0	1
1	1	1

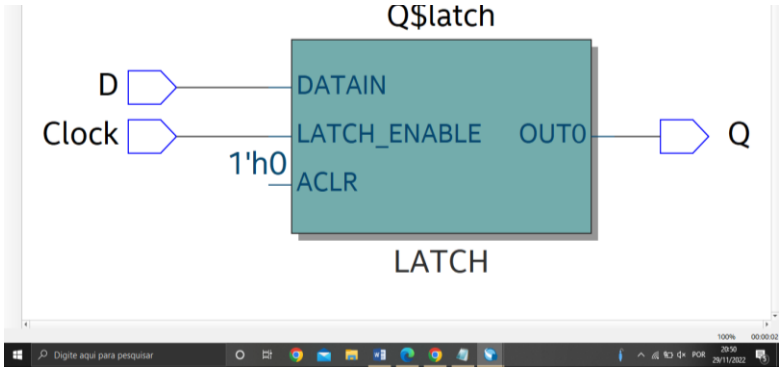

figura 1 - flipd

1.2 Registrador Flip Flop tipo JK

Quando é aplicado nivel lógico em ambas as entradas (J ou K), é resultante em inversão desses níveis antes da aplicação do clock. Então, resumidamente falando, o valor será aquele inverso antes do clock. Na imagem a seguir, o reset sendo representado por “resetar”, o signal como “nr” e por seguida as entradas J e K; “Q” como seu output.

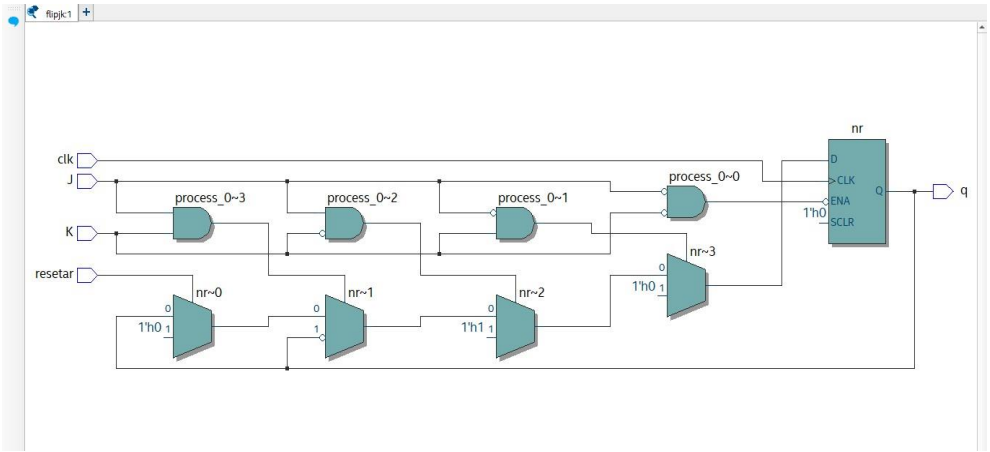

figura 2– jk

2. Multiplexador 4x1

O multiplexador (ou mux) possui várias entradas com apenas uma saída; na demonstração a seguir o mux apresenta 4 entradas, percorrendo até a saída através de dados recebidos. A entrada denominada “entvetor” é um vetor, devido ao multiplexador apresentar 4 bits (precisa de 2 bits de seleção, declarar “1 downto 0” ocuparia 2 posições).

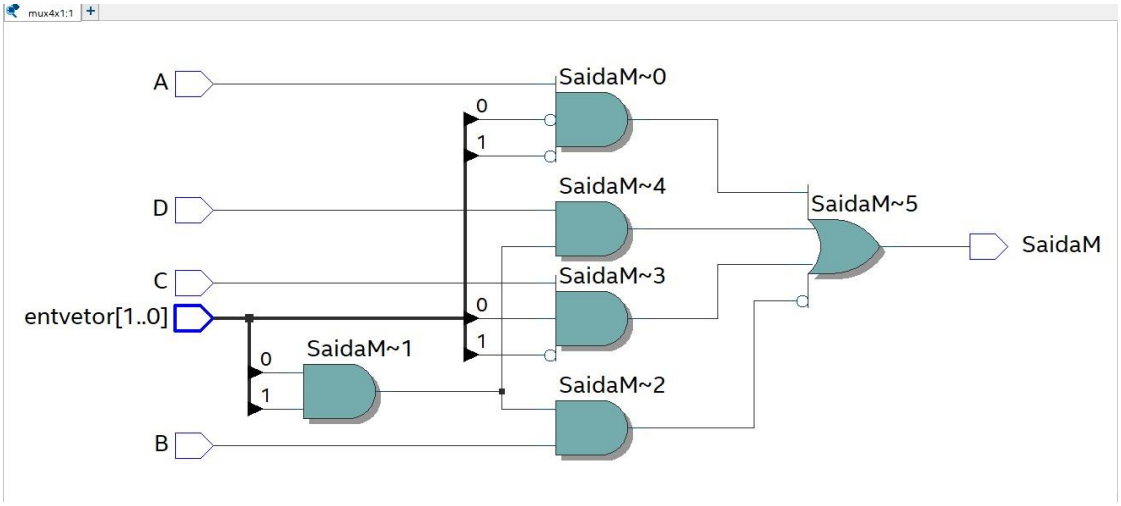

figura 3 -multiplexador(mux)

3. Porta lógica XOR

A porta lógica XOR é uma porta com duas entradas, que a partir delas determina os níveis lógicos que a saída vai produzir. Por exemplo, se as entradas estiverem com valores diferentes o nível lógico vai ser o 1, caso contrário, se as entradas estiverem com valores iguais será 0.

Nas imagens a seguir, será observado os circuitos AND, OR e NOT produzidas no *Quartus*; são portas básicas que no fim formará a porta lógica XOR.

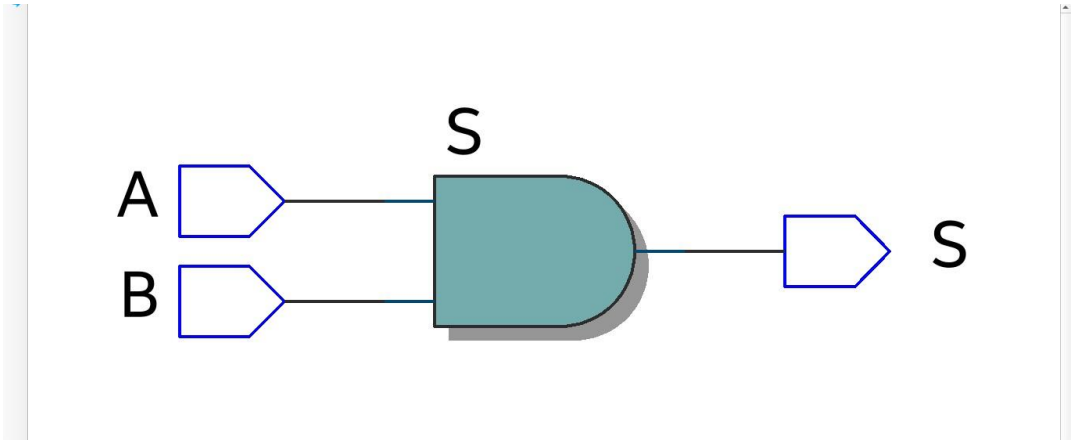

figura 4 – porta básica AND

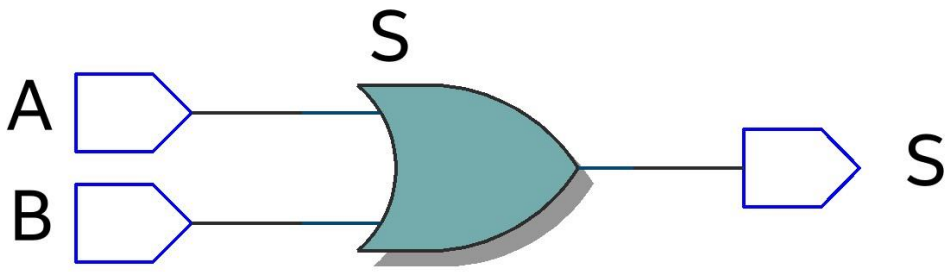

figura 5 - porta básica OR

Mais abaixo temos o circuito XOR com todas as portas básicas citadas acima.

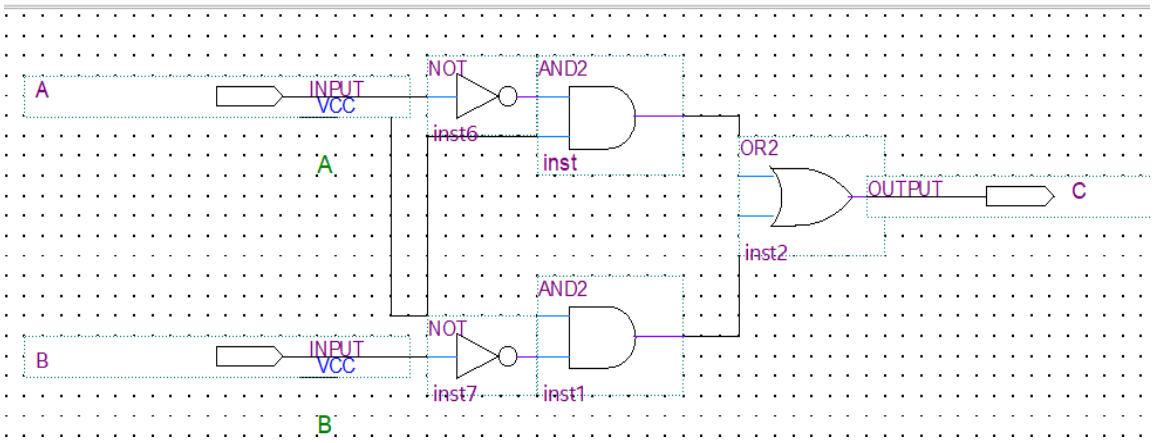

figura 6 - xor completo

4. Somador de 8 bits e somatório 8 bits +4

Como o próprio nome diz, a função do somador é efetuar a adição de 2 números; primeiramente, o circuito recebe esses dois números pra no final ser feito a soma. Na próxima figura vemos a representação desse somador de 8 bits somando com +4 e no outro um somador comum. A chave principal pra formação desse somador foi o uso do std logic vector com “downto”. Dependendo dos valores da soma é possível usar essas declarações.

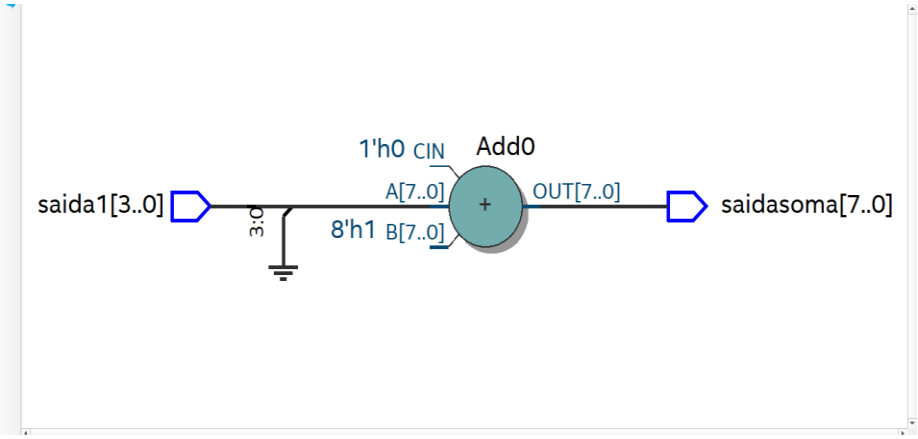

figura 7 - somatório +4

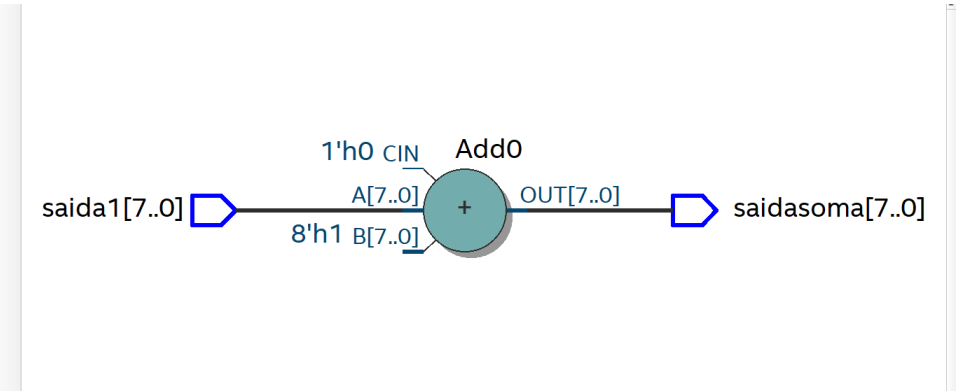

figura 8 - somatorio

5. Memória ROM 8 bits

A memória ROM é responsável pela leitura dos dados armazenados, geralmente são usados para sistemas operacionais;

6. Banco de Registradores 8 bits

O componente banco de registradores tem como principal objetivo escrever, ler e armazenar valores nos registradores.

7. Unidade de Controle uniciclo Mips 16 bits

O componente unidade de controle tem como principal objetivo administrar as flags necessárias para cada tipo de instrução;ela que deve garantir a correta execução dos programas e a utilização dos dados corretos nas operações que as manipulam. Abaixo, temos a implementação de uma UC. Esse mesmo componente tem por suas entradas o clock “uc_clk” (1 bit) e o opcode (3 bits de operacao na alu). Saídas: o jump (1 bit de verificacao de flag e operacao do tipo jump), branch (1 bit de verificação para instrução tipo j), mread (1bit leitura de memória), reg (1 bit memoria registrador), ulop (3 bits operacao na alu), mwrite (1 bit flag memória), src (1 bit se caso tiver operação) e regwrit (1 bit se tem dados escritos no banco de registradores).

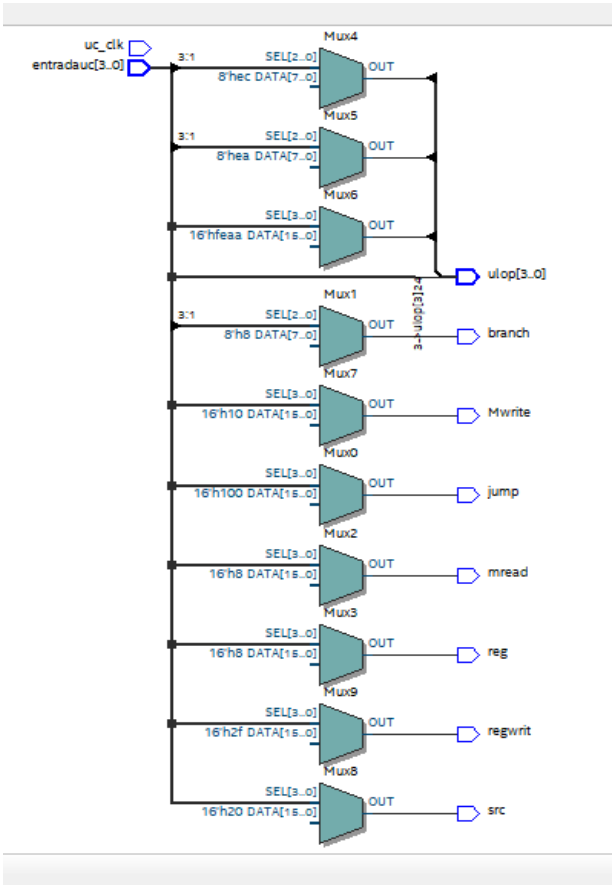

figura 9 —uc

OPCODE	JUMP	BRANCH	MREAD	MREG	ULOP	MEMWRIT	SRC	REGWRITE
0000	0	0	0	0	0	0	0	1
0001	0	0	0	0	0	0	1	1
0010	0	0	0	0	0	0	0	1
0011	0	0	0	0	0	0	1	1
0100	0	0	1	1	0	0	0	1
0101	0	0	0	0	0	1	0	0
0110	0	0	0	0	0	0	1	1
0111	0	1	0	0	0	0	0	0
1000	0	0	0	0	0	0	0	0

REPOSITÓRIO: NatalyAlmeida/AOC_Nataly_UFRR_LabCircuitos_2022: Atividade avaliativa de laboratório de circuitos referente a disciplina de Arquitetura e Organização de Computadores. (github.com)