

UNIVERSIDADE FEDERAL DE RORAIMA

CENTRO DE CIENCIAS E TECNOLOGIA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

GABRIEL ARCANJO CAMPELO FADOUL

**LABORÁTORIO VHDL**

Boa Vista, RR

2018

GABRIEL ARCANJO CAMPELO FADOUL

**LABORATÓRIO DE VHDL**

Trabalho apresentado como requisito parcial para obtenção de nota na disciplina de Arquitetura e Organização de Computadores, ofertada pelo curso de Ciência da Computação da Universidade Federal de Roraima.

Prof. Herbert Oliveira Rocha

Boa Vista, RR

2018

Sumário

[Sumário 3](#_Toc528422389)

[1. Questão [Flip-Flop D e JK] 6](#_Toc528422390)

[1.1 Código 6](#_Toc528422391)

[1.2 RTL 7](#_Toc528422392)

[1.3 Wave Form 8](#_Toc528422393)

[1.4 Descrição 8](#_Toc528422394)

[2. Questão [Multiplexador de 4 entradas] 9](#_Toc528422395)

[2.1 Código 9](#_Toc528422396)

[2.2 RTL 9](#_Toc528422397)

[2.3 Wave Form 9](#_Toc528422398)

[2.4 Descrição 10](#_Toc528422399)

[3. Questão [Porta XOR Utilizando Port Map] 10](#_Toc528422400)

[3.1 Código 10](#_Toc528422401)

[3.2 RTL 11](#_Toc528422402)

[3.3 Wave Form 11](#_Toc528422403)

[3.4 Descrição 11](#_Toc528422404)

[4. Questão [Somador de valor com 4] 12](#_Toc528422405)

[4.1 Código 12](#_Toc528422406)

[4.2 RTL 12](#_Toc528422407)

[4.3 Wave Form 12](#_Toc528422408)

[4.4 Descrição 12](#_Toc528422409)

[5. Questão [Memória ROM de 16 bits] 13](#_Toc528422410)

[5.1 Código 13](#_Toc528422411)

[5.2 RTL 13](#_Toc528422412)

[5.3 Wave Form 14](#_Toc528422413)

[5.4 Descrição 14](#_Toc528422414)

[6. Questão [Memória RAM de 16 bits] 15](#_Toc528422415)

[6.1 Códigos 15](#_Toc528422416)

[6.2 RTL 15](#_Toc528422417)

[6.3 Wave Form 16](#_Toc528422418)

[6.4 Descrição 16](#_Toc528422419)

[7. Questão [Banco de Registradores de 16 bits] 16](#_Toc528422420)

[7.1 Código 16](#_Toc528422421)

[7.2 RTL 17](#_Toc528422422)

[7.3 Wave Form 17](#_Toc528422423)

[7.4 Descrição 17](#_Toc528422424)

[8. Questão [Somador 16 bits] 18](#_Toc528422425)

[8.1 Códigos 18](#_Toc528422426)

[8.2 RTL 18](#_Toc528422427)

[8.3 Wave Form 18](#_Toc528422428)

[8.4 Descrição 18](#_Toc528422429)

[9. Questão [Unidade de Controle] 19](#_Toc528422430)

[9.1 Código 19](#_Toc528422431)

[9.2 RTL 19](#_Toc528422432)

[9.3 Wave Form 19](#_Toc528422433)

[9.4 Descrição 19](#_Toc528422434)

[10. Questão [ULA] 19](#_Toc528422435)

[10.1 Códigos 19](#_Toc528422436)

[10.2 RTL 20](#_Toc528422437)

[10.3 Wave Form 20](#_Toc528422438)

[10.4 Descrição 20](#_Toc528422439)

[11. Questão [Extensor de Sinal] 20](#_Toc528422440)

[11.1 Códigos 20](#_Toc528422441)

[11.2 RTL 21](#_Toc528422442)

[11.3 Wave Form 21](#_Toc528422443)

[11.4 Descrição 21](#_Toc528422444)

[12. Questão [Máquina de Estados] 22](#_Toc528422445)

[12.1 Códigos 22](#_Toc528422446)

[12.2 RTL 22](#_Toc528422447)

[12.3 Wave Form 22](#_Toc528422448)

[12.4 Descrição 22](#_Toc528422449)

[13. Questão [Contador Síncrono] 22](#_Toc528422450)

[13.1 Códigos 22](#_Toc528422451)

[13.2 RTL 22](#_Toc528422452)

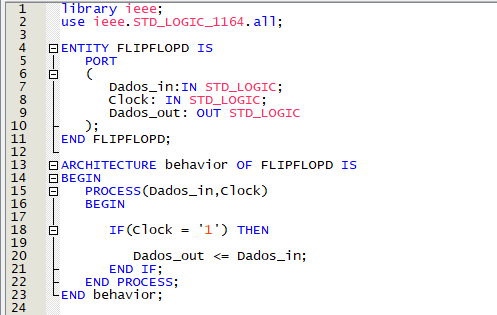
[13.3 Wave Form 22](#_Toc528422453)

[13.4 Descrição 22](#_Toc528422454)

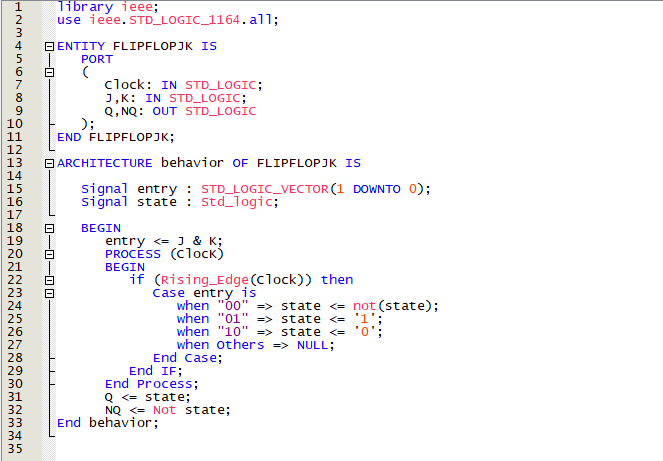
# Questão [Flip-Flop D e JK]

## Código

* Tipo – D:

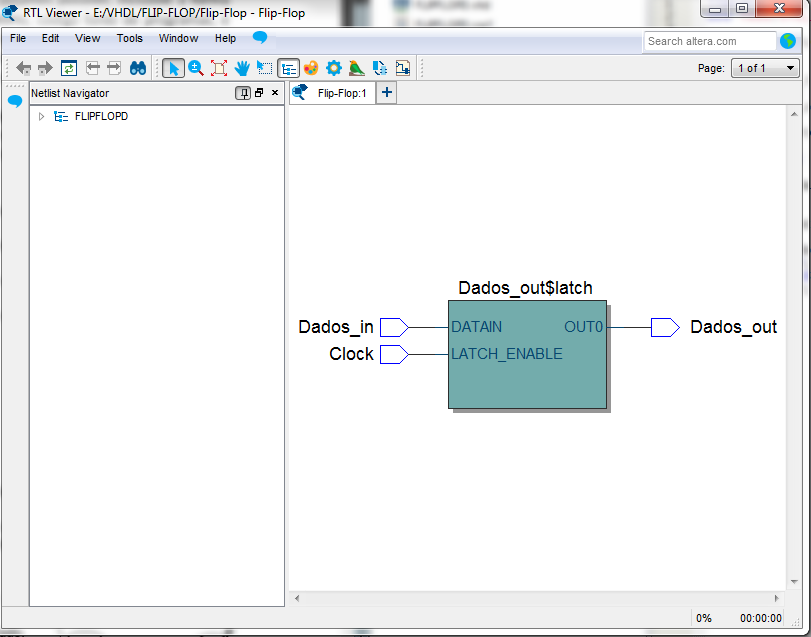


* Tipo JK

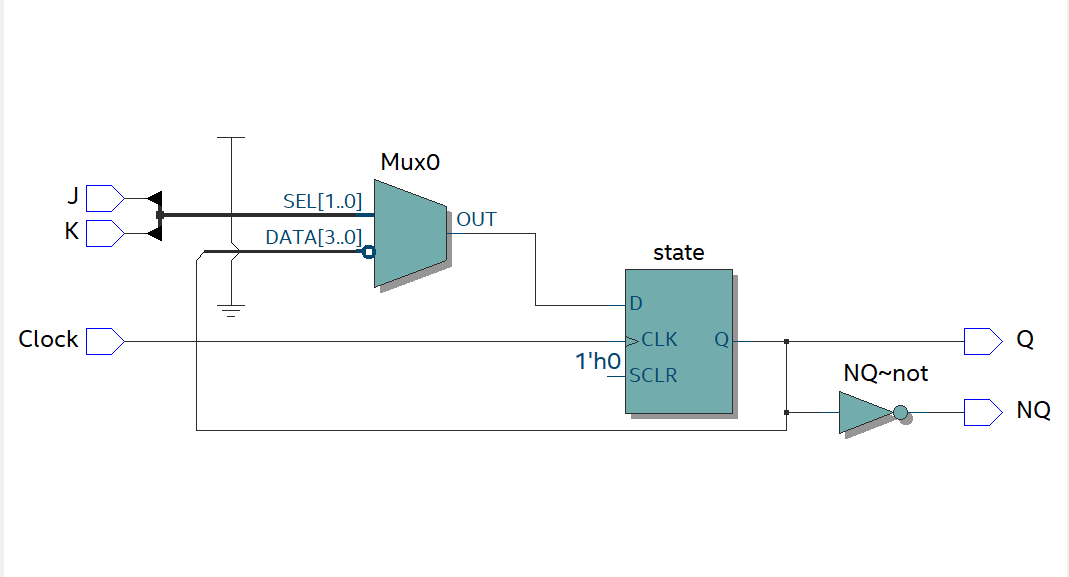


## RTL

* Tipo – D:

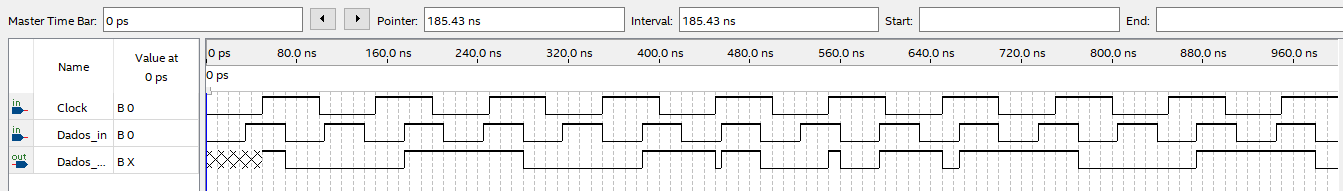


* Tipo JK

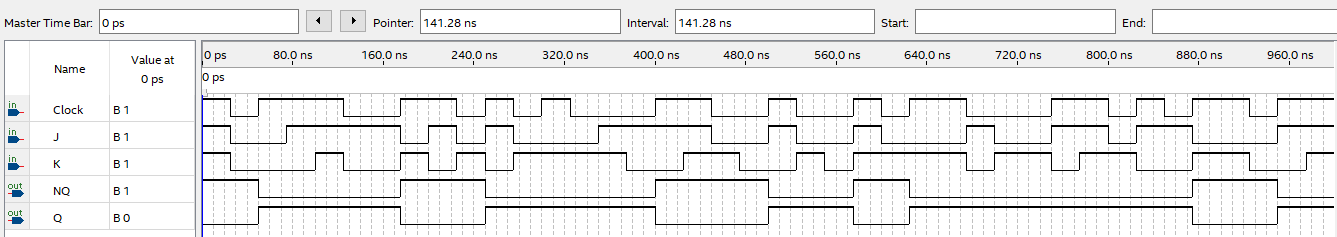


## Wave Form

* Tipo – D:



* Tipo JK:



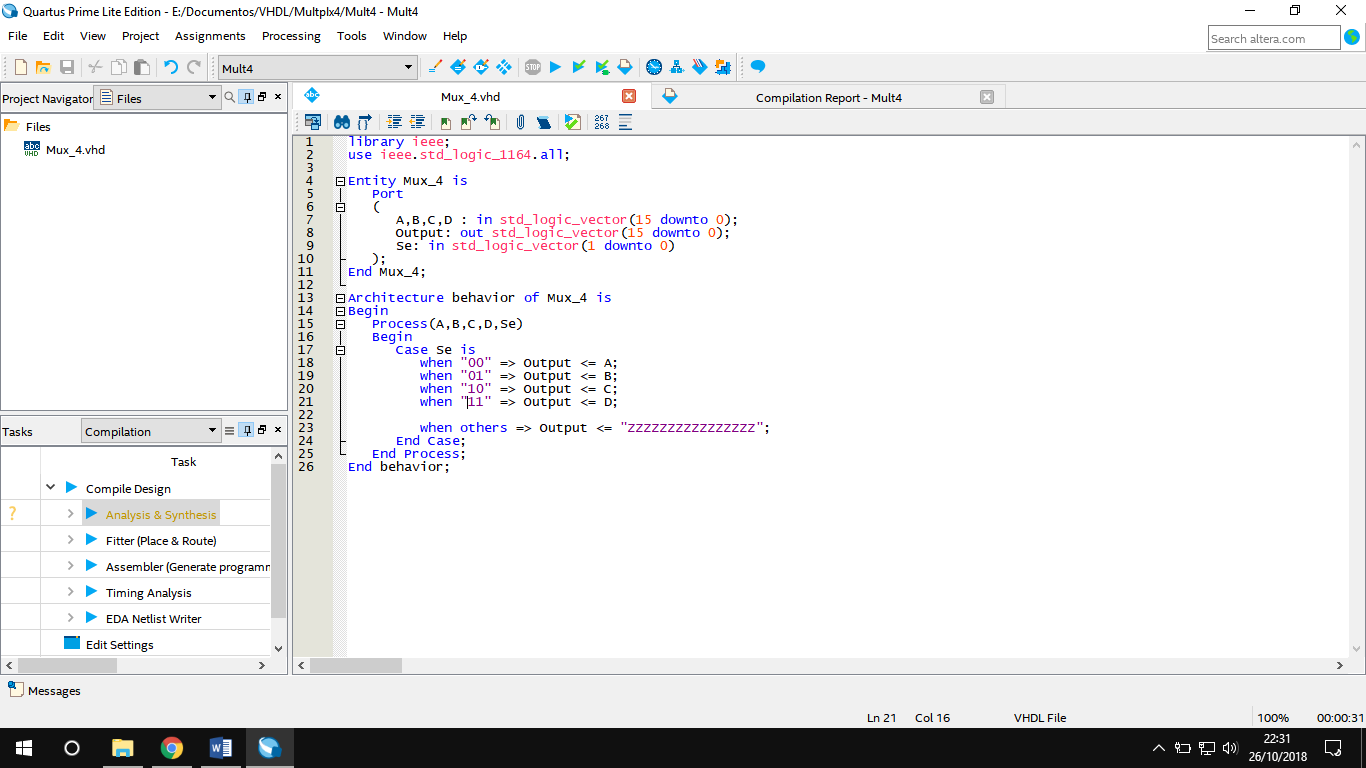
## Descrição

O Flip Flop D armazena o valor que estiver na porta de entrada toda vez que o Clock é ativado. Já o Flip Flop JK é um pouco mais complexo, ele tem duas 3 entradas J, K e o Clock, Dependendo do J e do K ele terá um retorno de acordo com a tabela abaixo:

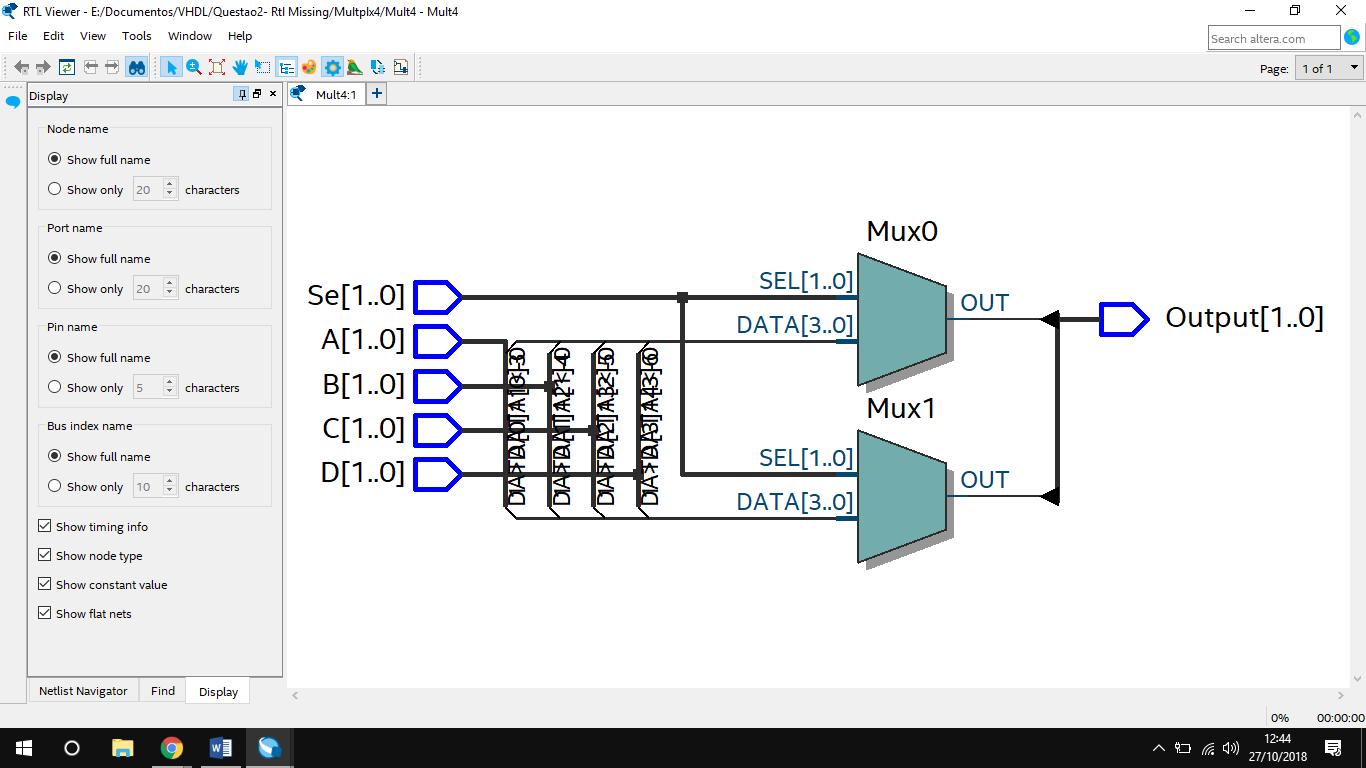
|  |  |  |
| --- | --- | --- |
| Tabela Verdade | | |
| J | K | Saída |
| 1 | 1 | Qant |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | ¬Qant |

# Questão [Multiplexador de 4 entradas]

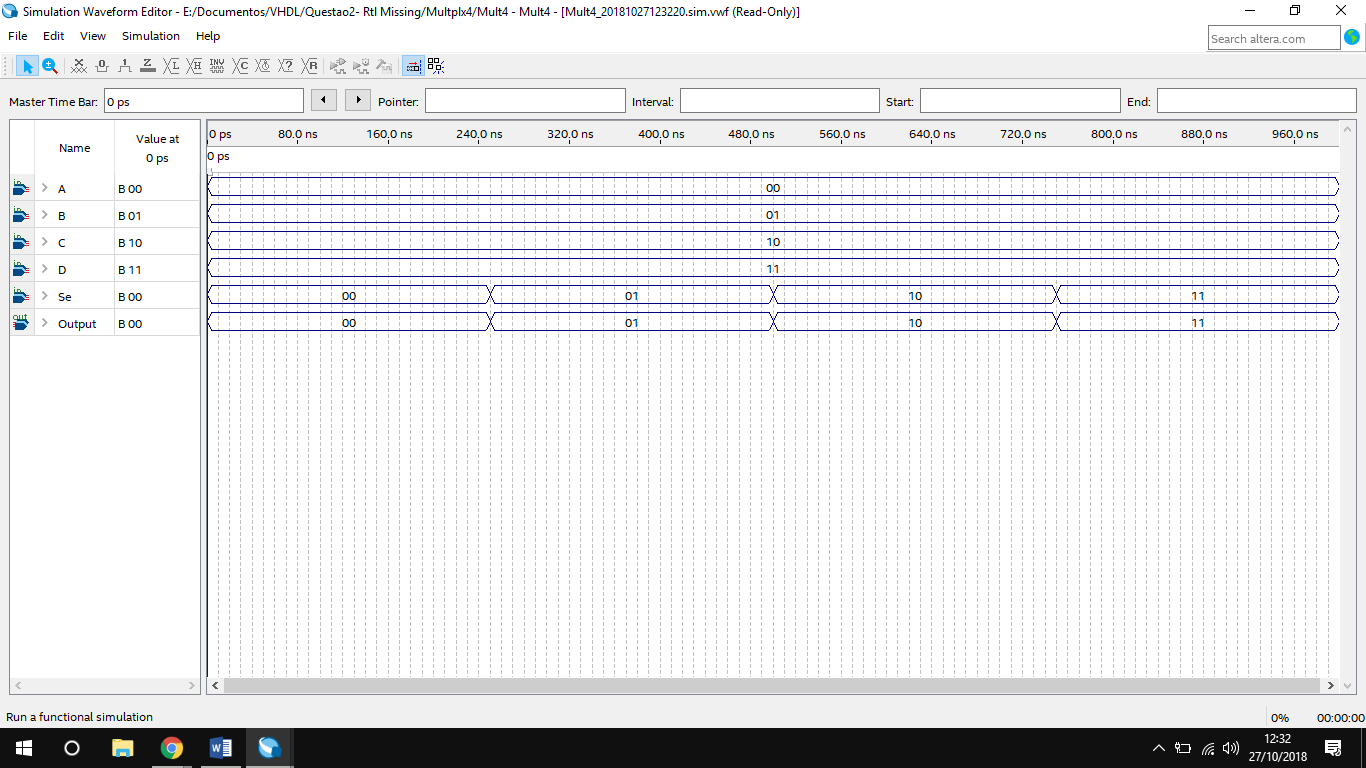
## Código



## RTL



## Wave Form

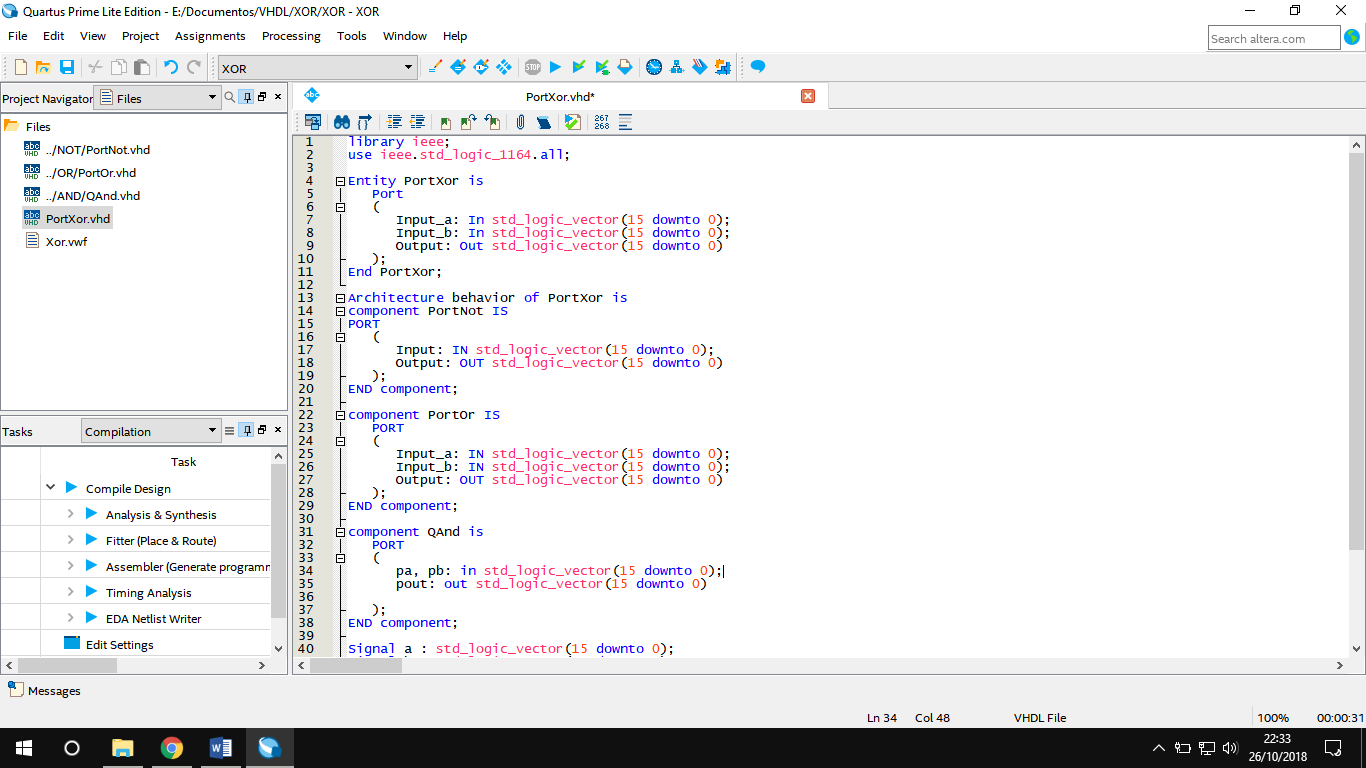


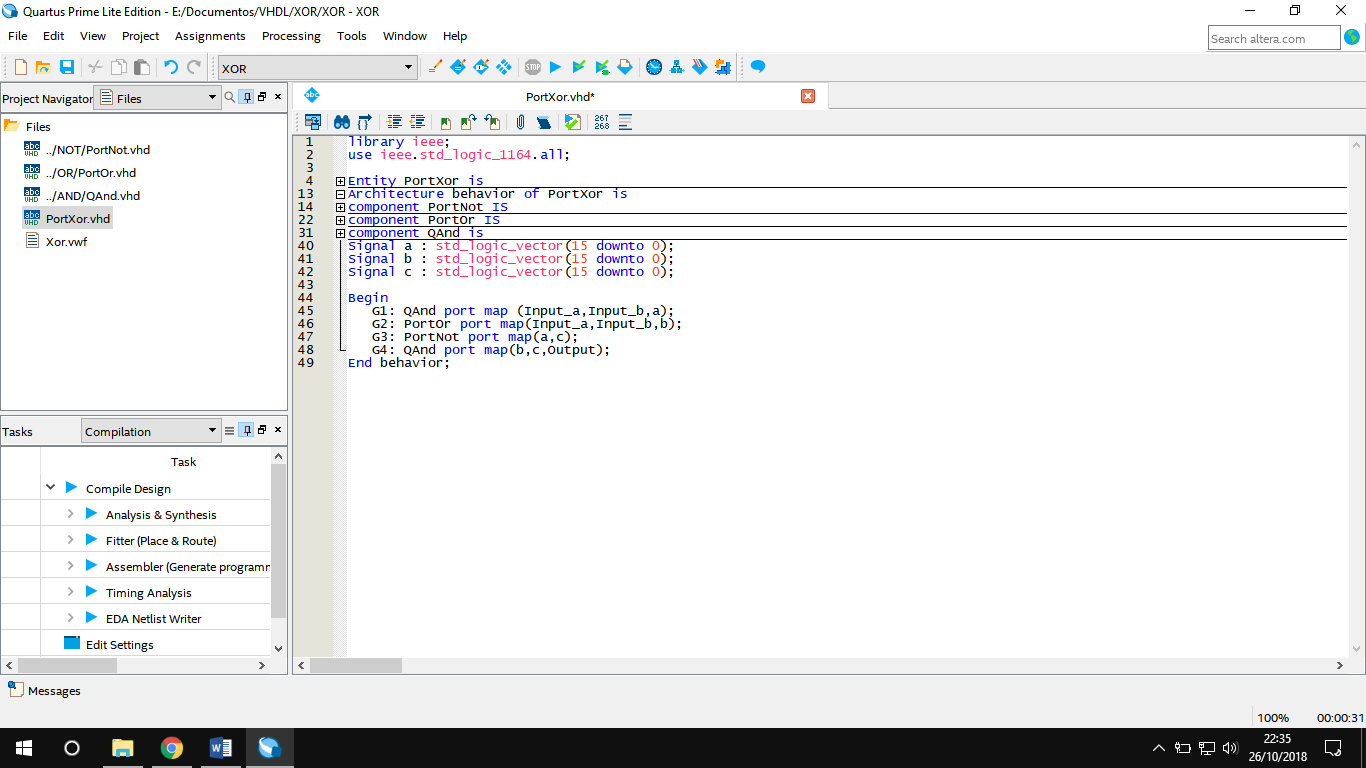
## Descrição

O componente recebe 4 valores, e de acordo com a flag recebida no seletor, um dado específico será direcionado a saída.

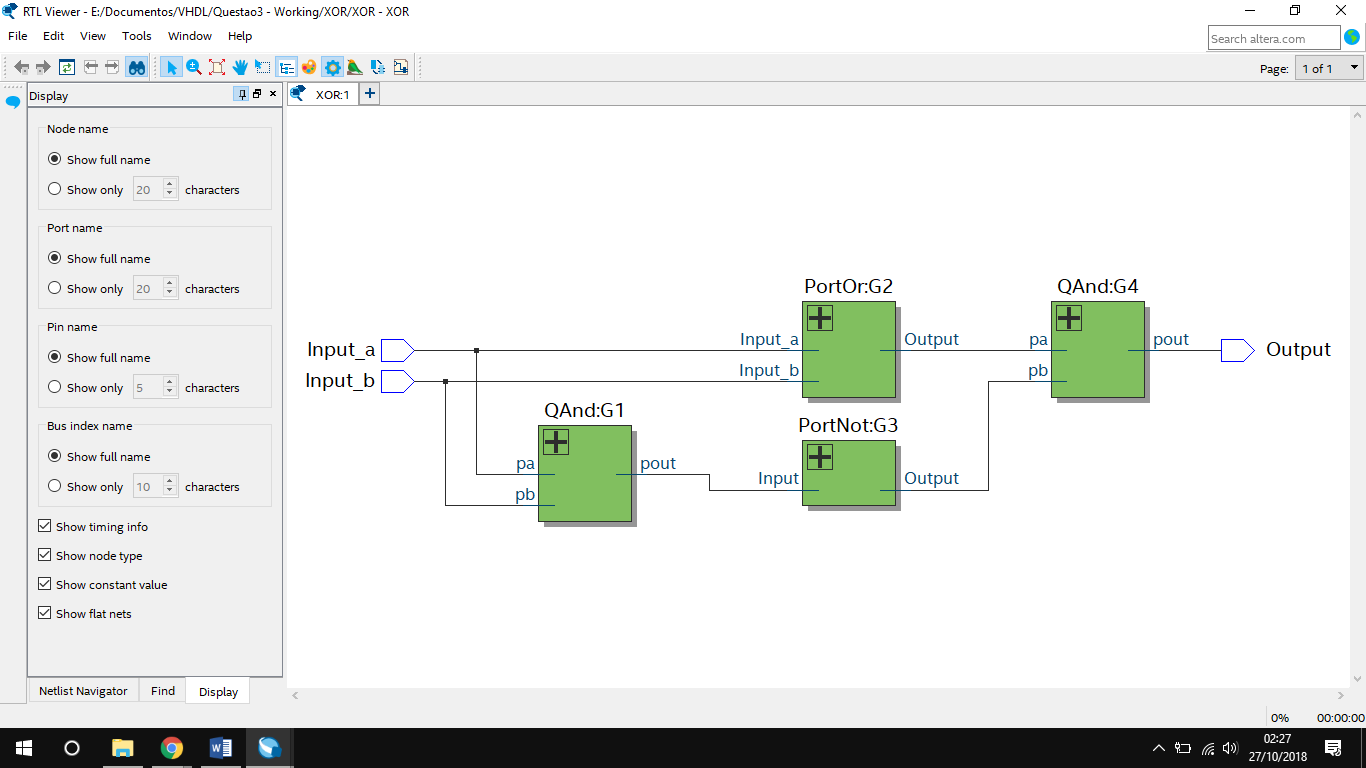
# Questão [Porta XOR Utilizando Port Map]

## Código

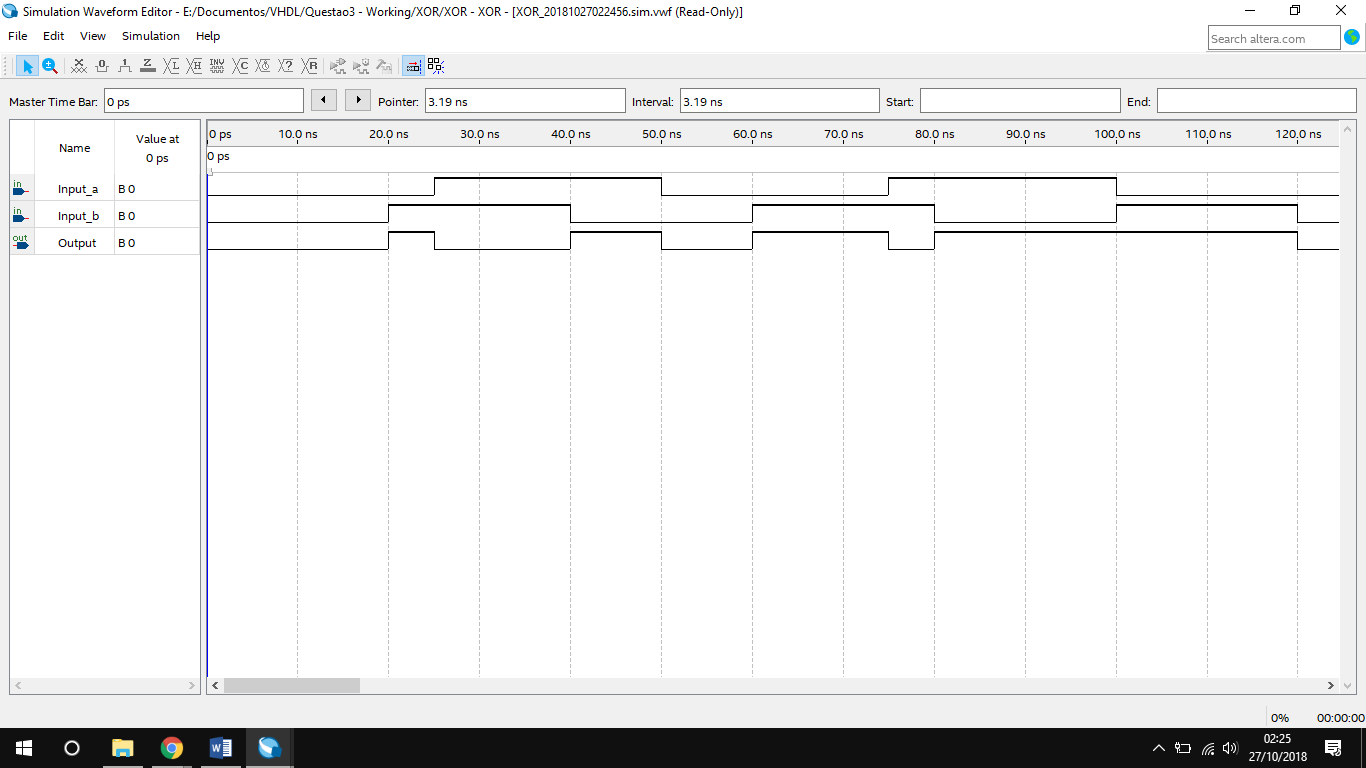




## RTL



## Wave Form

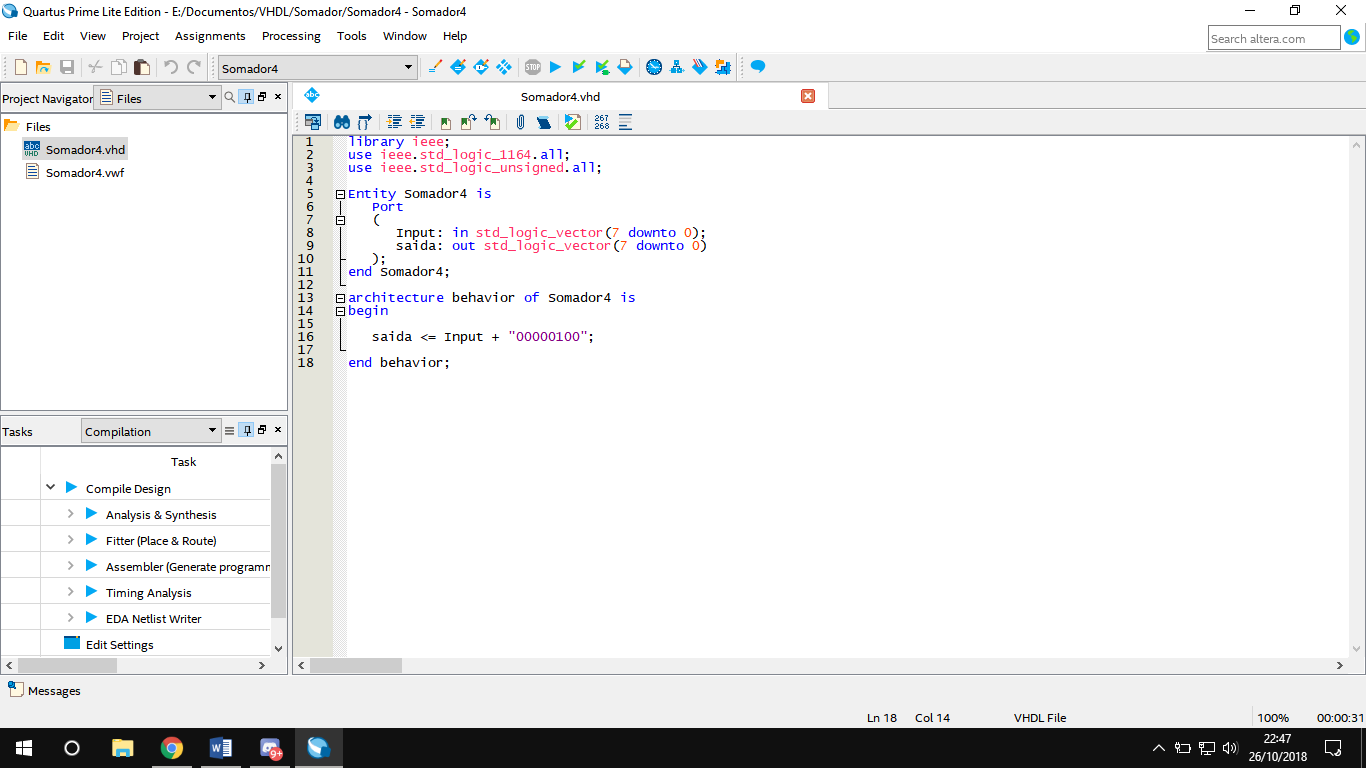


## Descrição

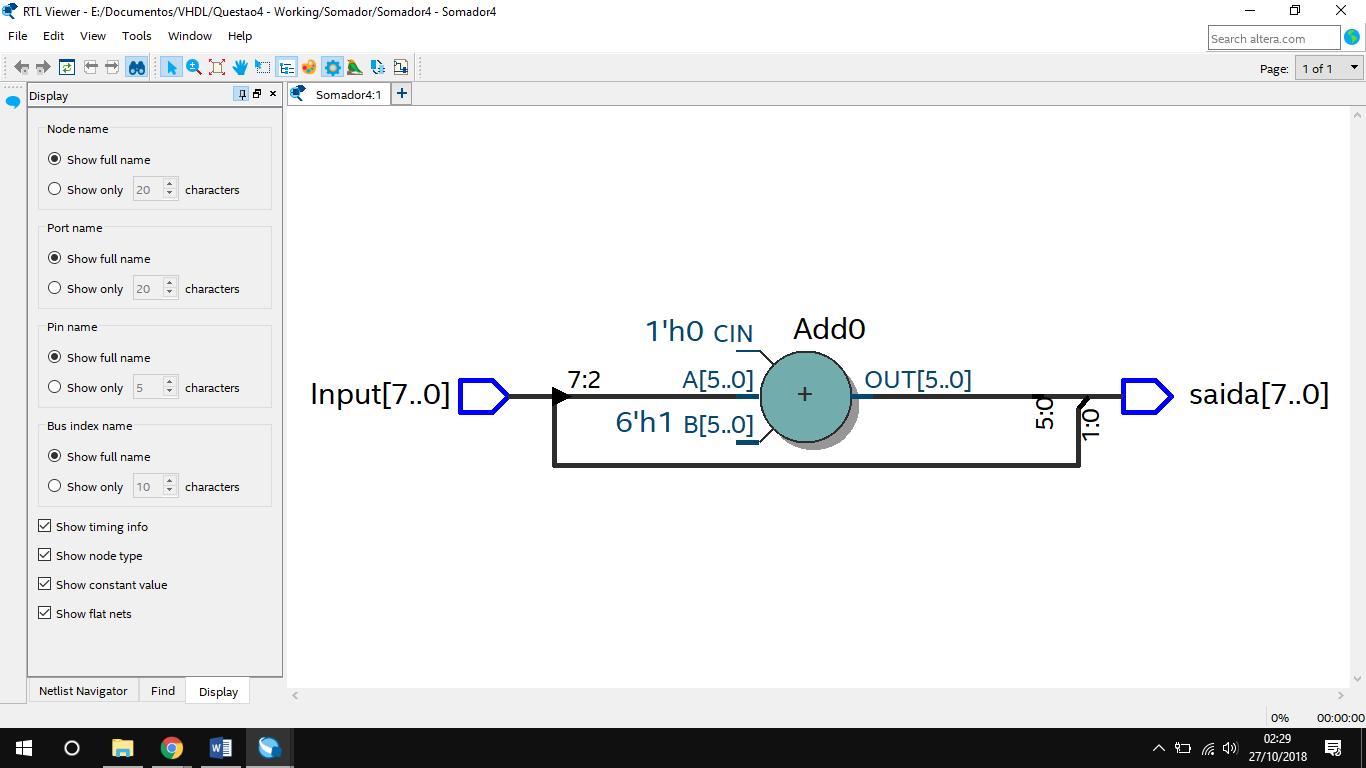
A idéia da questão era a utilização de Port Map, e foi o que fiz, utilizei componentes que eu já havia construído durante as aulas e os conectei, nesse caso usei os componentes AND, OR e NOT, já que utilizei a expressão ¬ (P ^ Q) ^ (P v Q) para o desenvolvimento da porta, utilizei também de barramentos, que são nomeados como ‘a’, ‘b’, ‘c’.

# Questão [Somador de valor com 4]

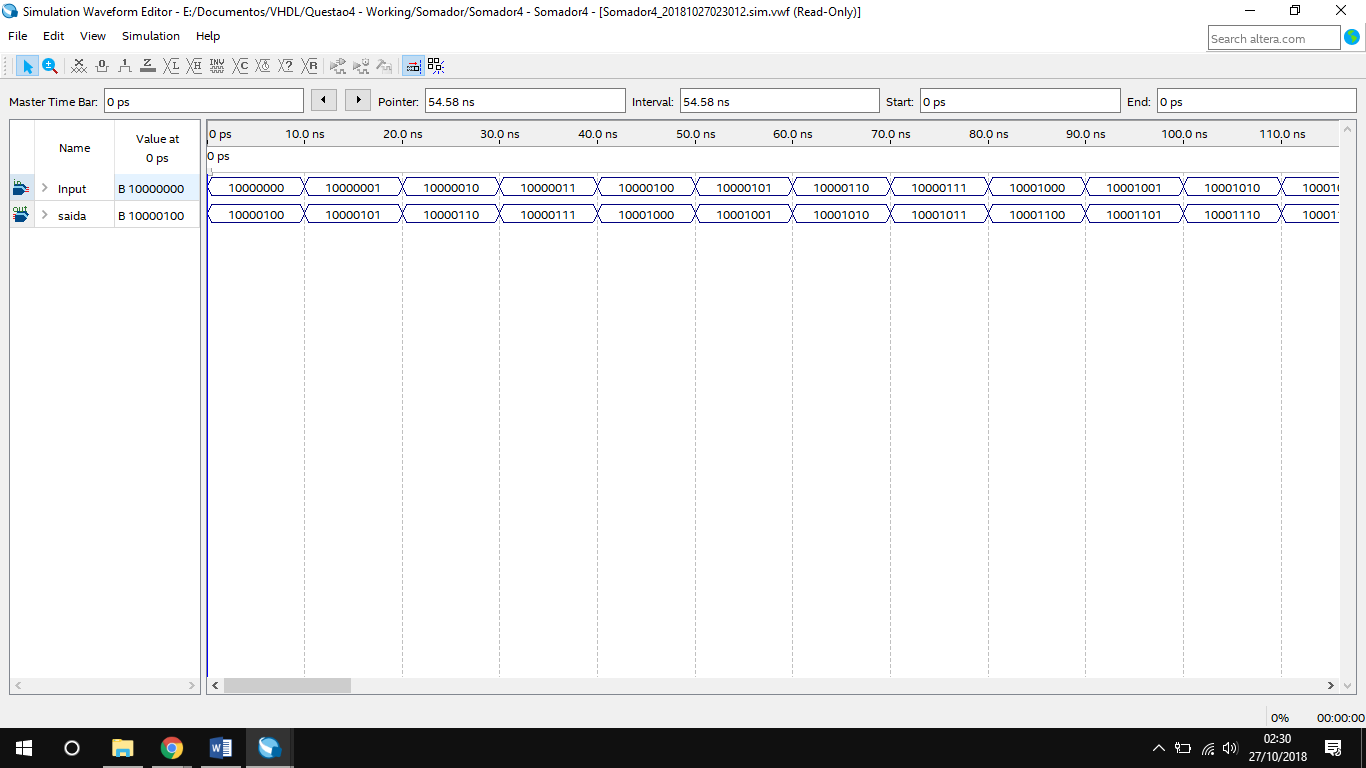
## Código



## RTL



## Wave Form

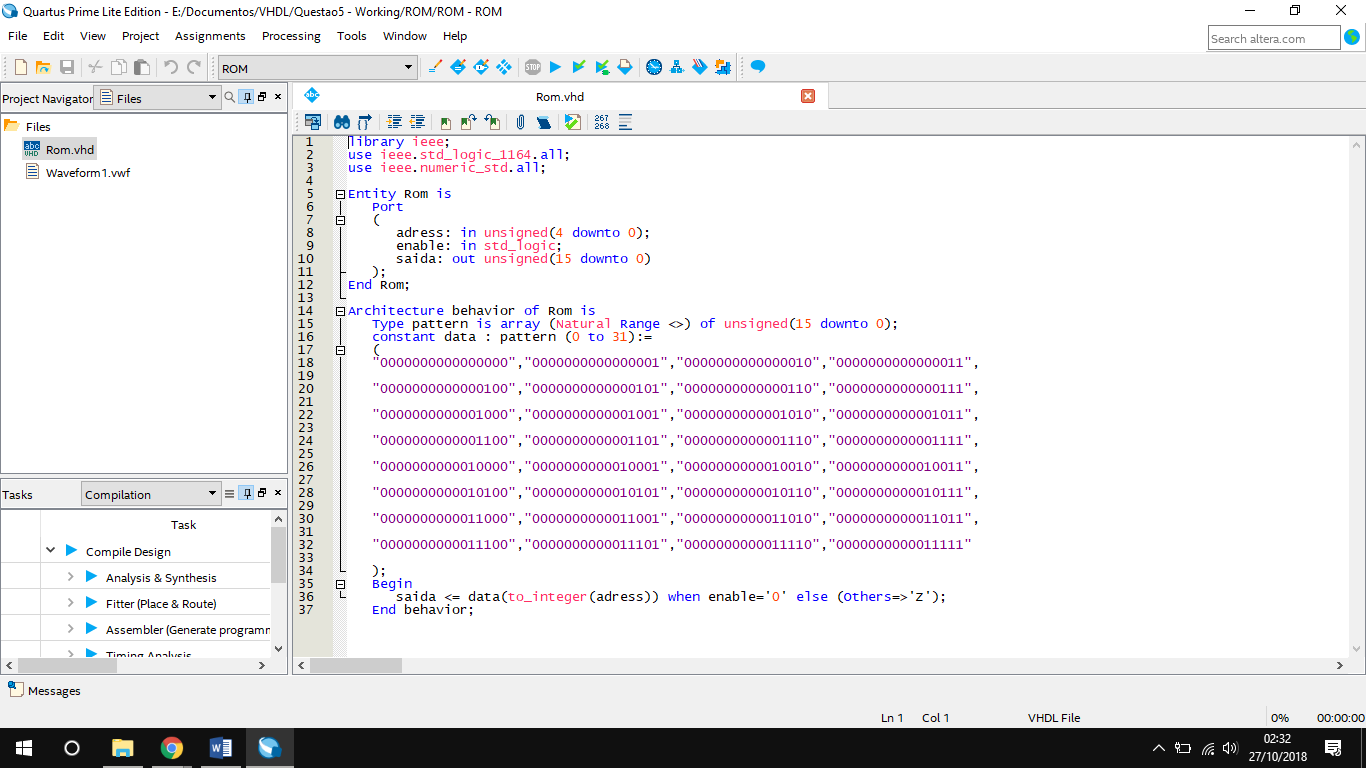


## Descrição

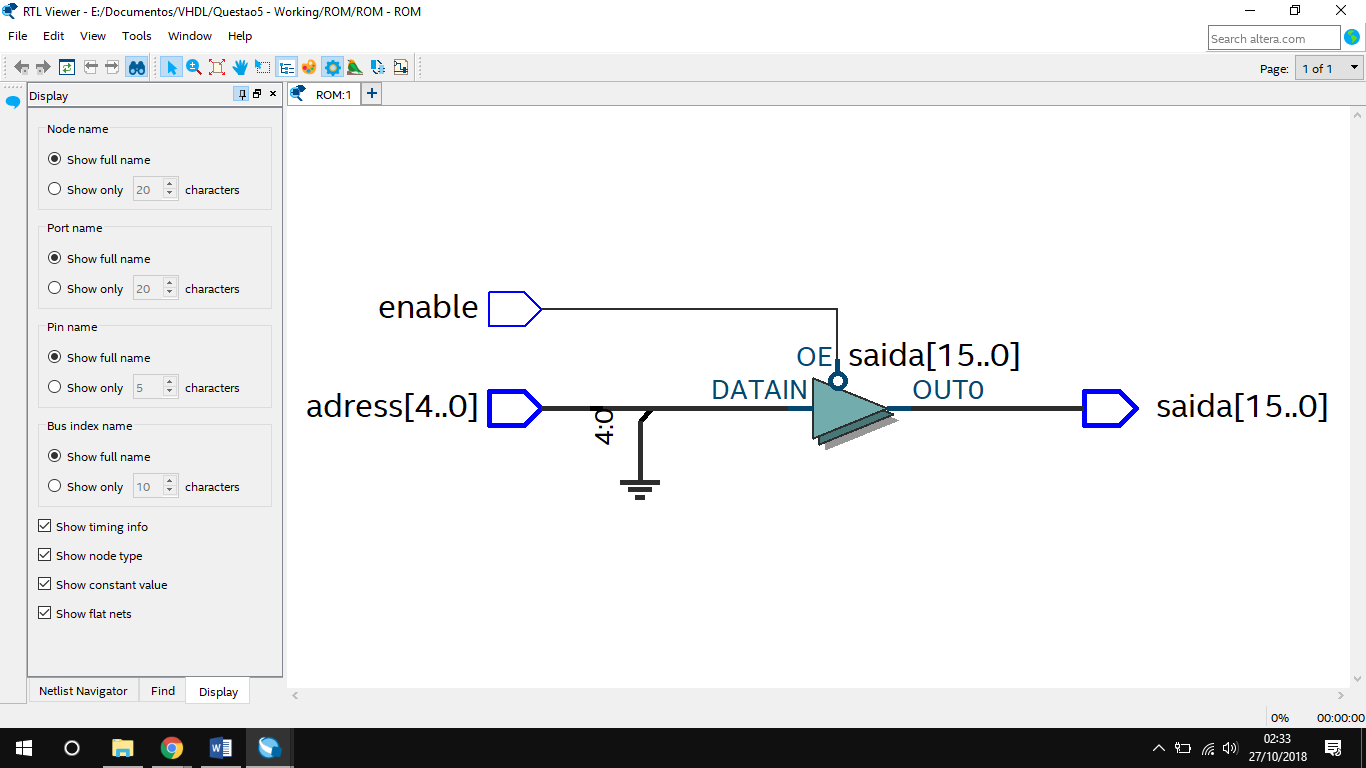
Um somador que ao receber uma de entrada de 8 bits, soma mais 4 a este valor, sem muitos segredos, simplesmente somei um vetor de 8 bits representando 4 ao valor de entrada.

# Questão [Memória ROM de 16 bits]

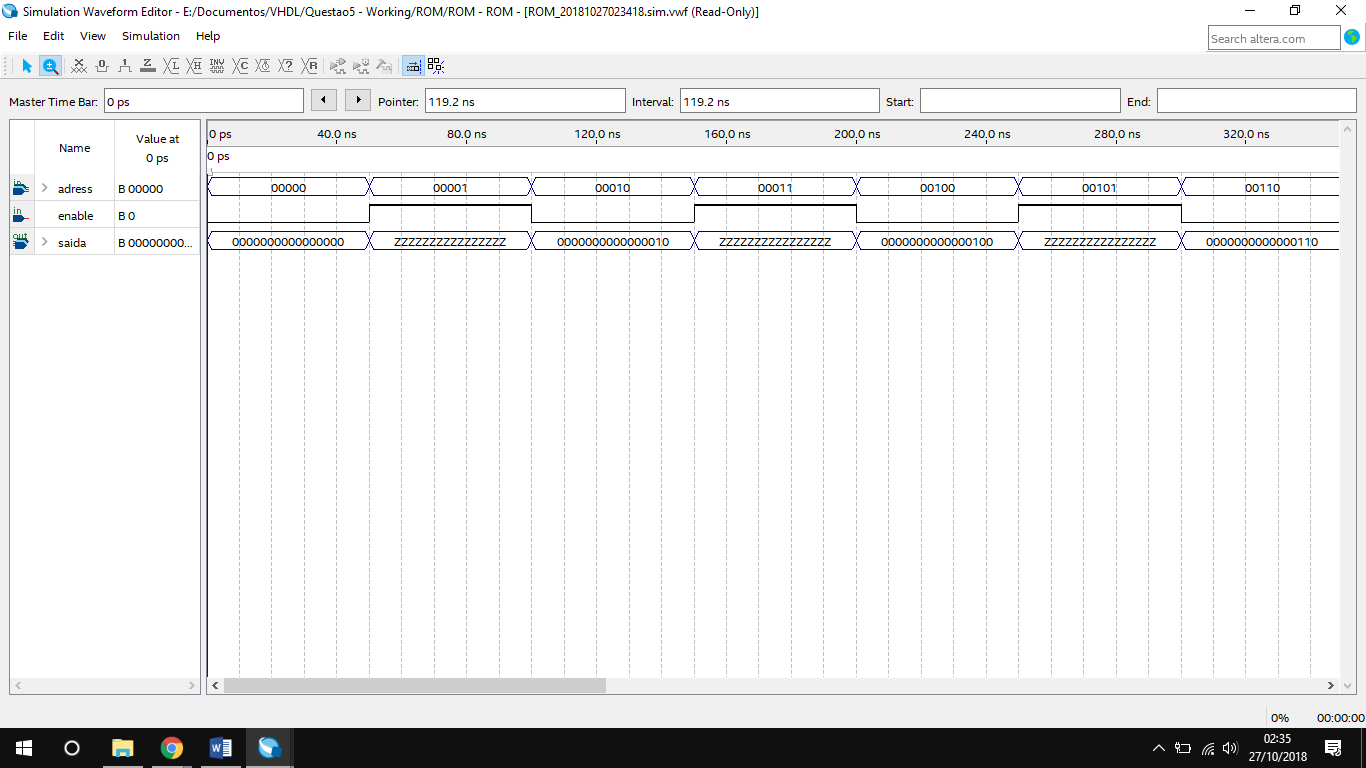
## Código



## RTL



## Wave Form

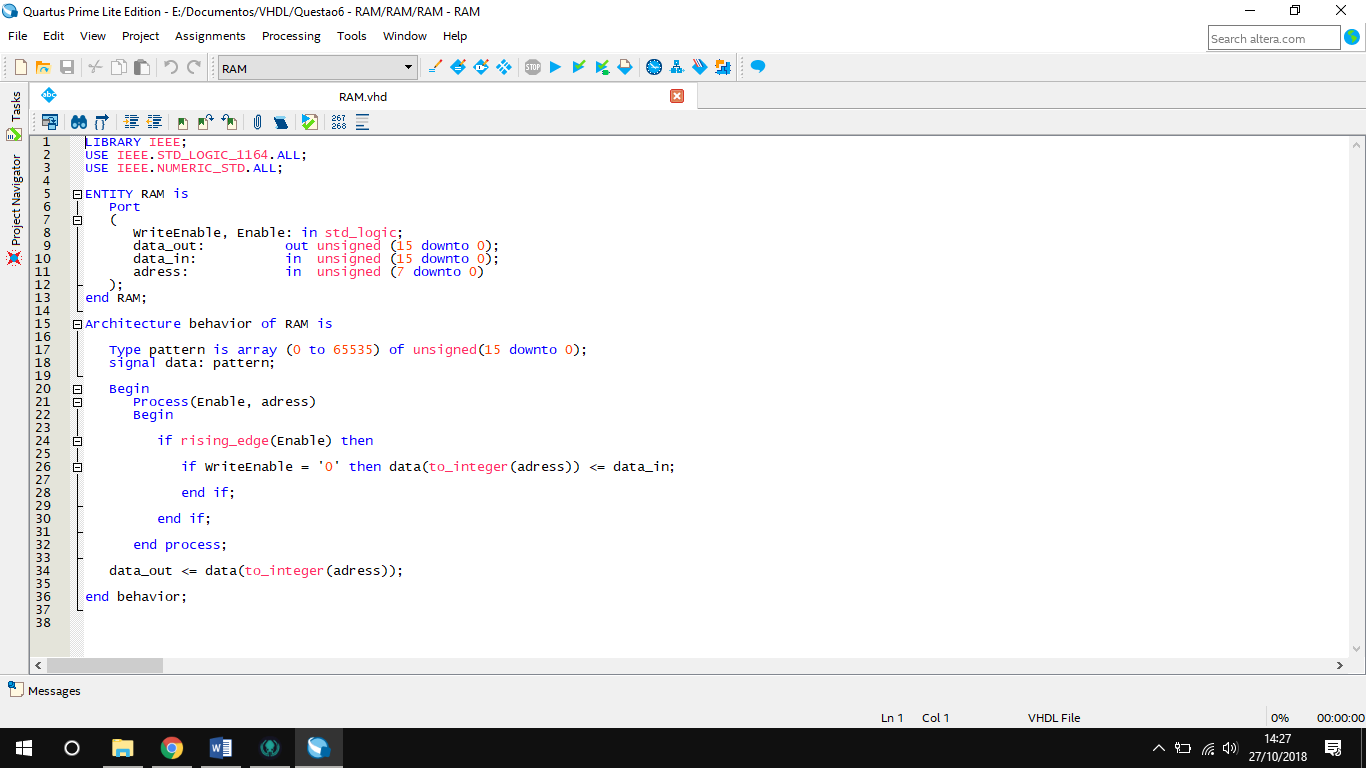


## Descrição

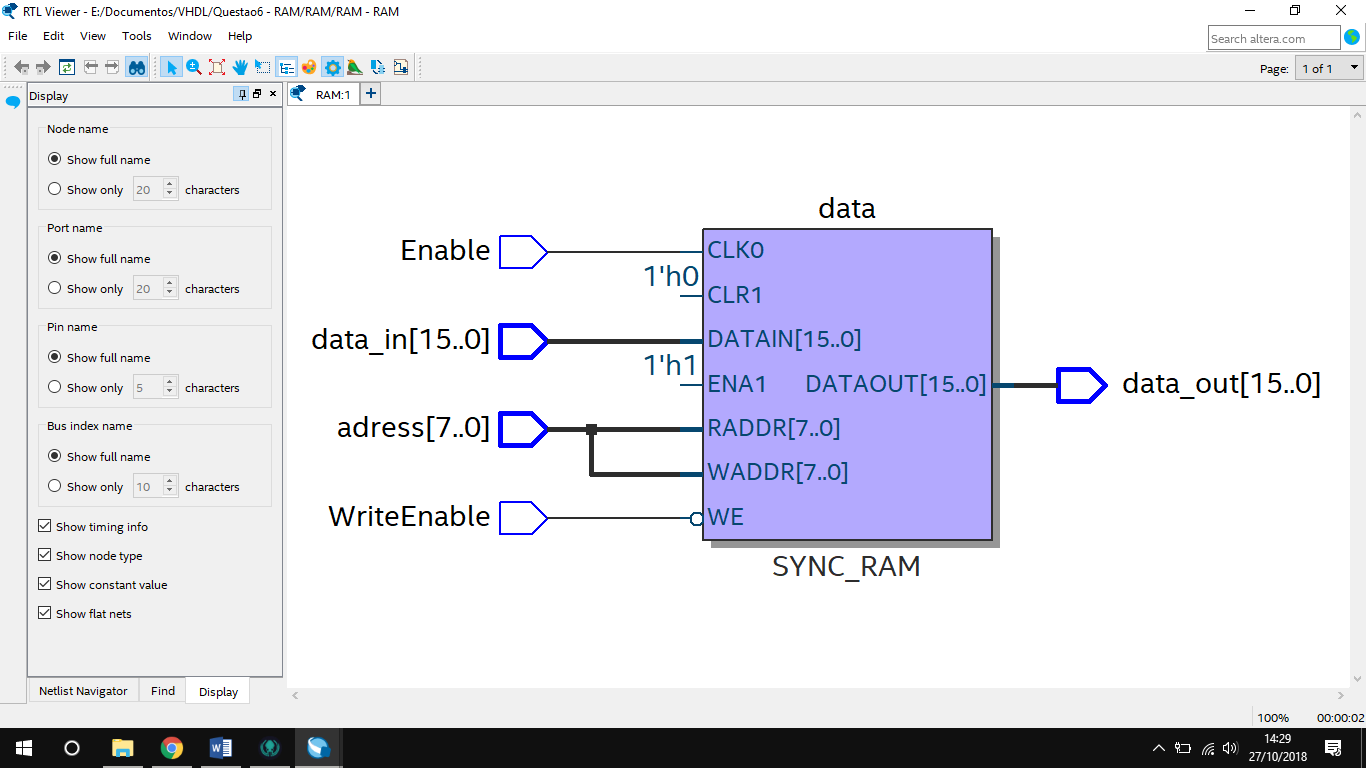
A memória ROM é uma memória não volá til, ou seja, não sensível a oscilações de energia, logo, ela tem a capacidade de armazenar dados mesmo com o componente desligado. Assim como diz o nome, a memória ROM (Read Only Memory) não tem escrita de dados, somente saída de dados. Nesse dispositivo, eu criei um vetor de 32 espaços que armazena valores de 16 bits, usei duas entradas de valores e uma saída. As duas entradas são: entrada indicando o endereço a ser lido, e outra indicando o ligamento e o desligamento do componente.

# Questão [Memória RAM de 16 bits]

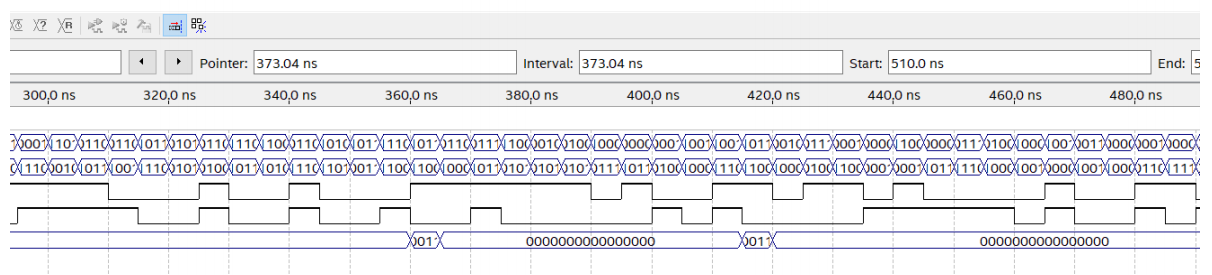
## Códigos



## RTL



## Wave Form

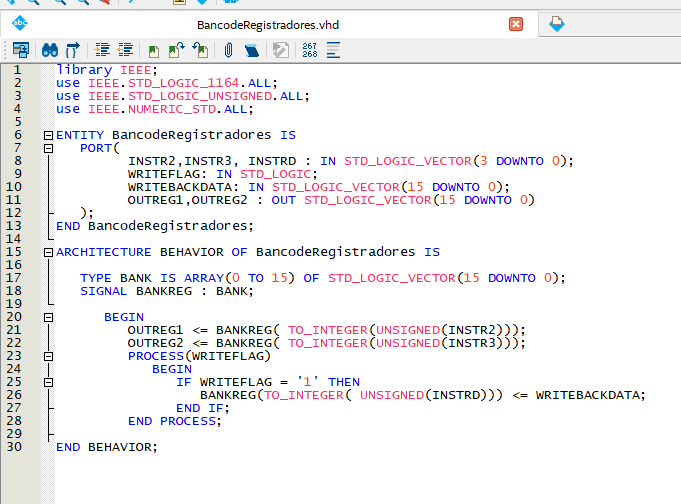


## Descrição

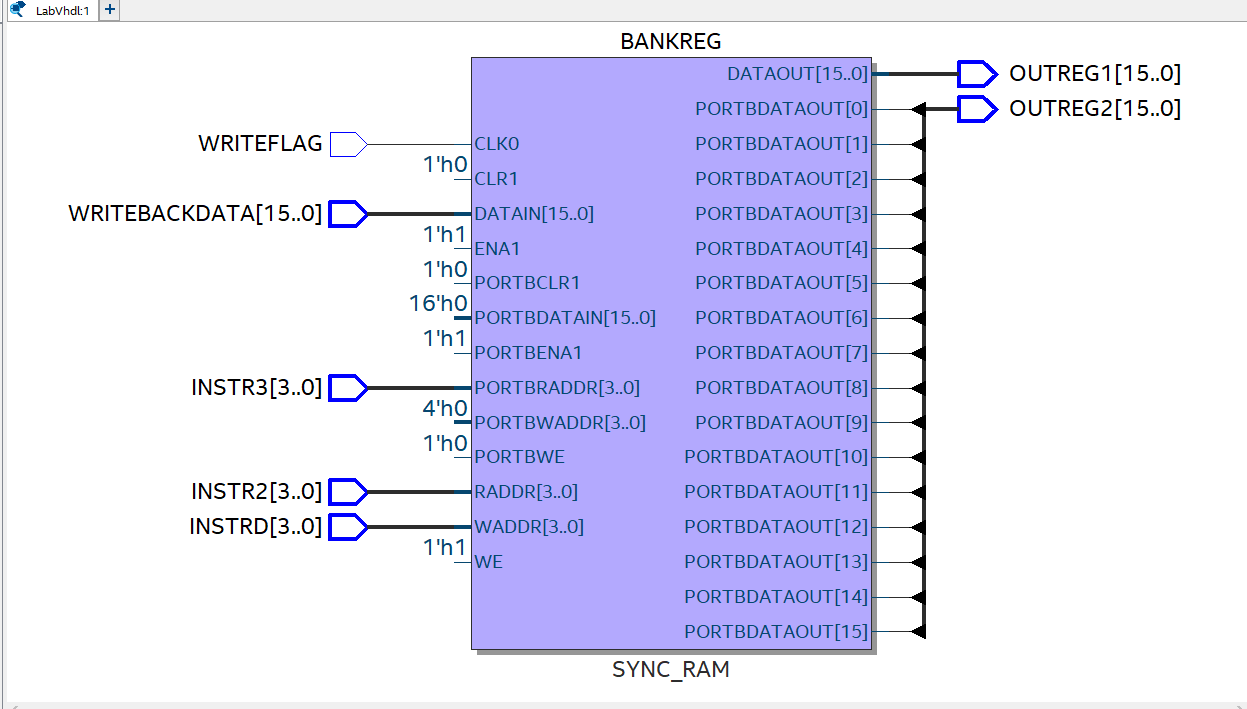
Diferente da memória ROM, a memória RAM (Random Acess Memory) é volátil e tem escrita de dados. Com um funcionamento de recuperação de dados parecido com a da ROM, o que a diferencia da mesma, é que agora temos uma porta de entrada de dados e outra chamada Write Enable que me indica se componente vai ler ou escrever na memória. No caso, se ele estiver em 0, ele irá ler o adress e irá endereçar o valor da porta de entrada de dados na respectiva posição. Caso esteja com valor 1, ele irá jogar na saída de dados o valor que está armazenado no endereço adress.

# Questão [Banco de Registradores de 16 bits]

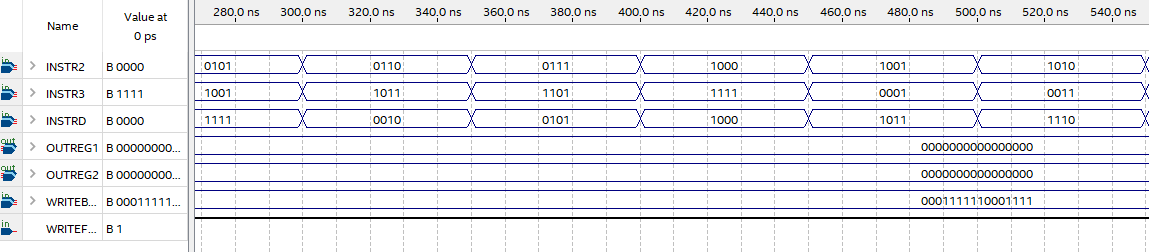
## Código



## RTL



## Wave Form

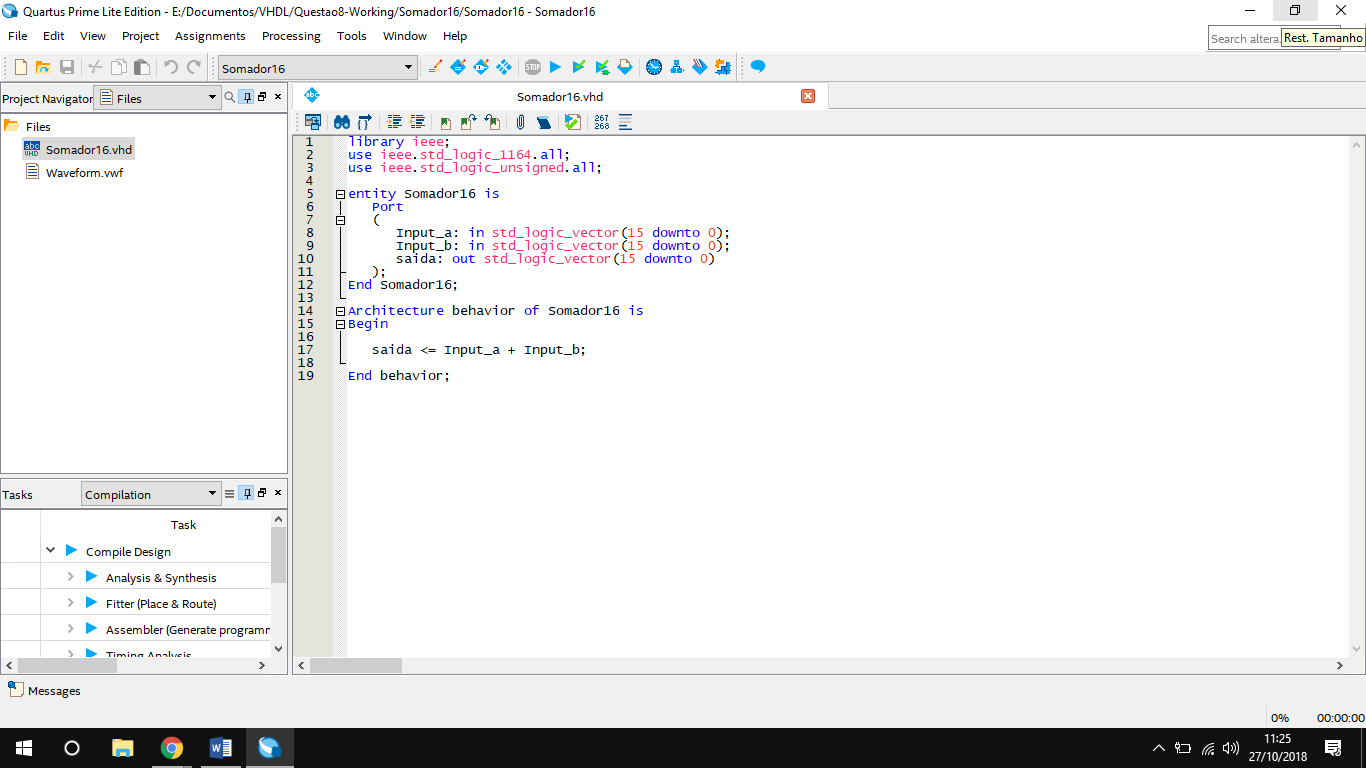


## Descrição

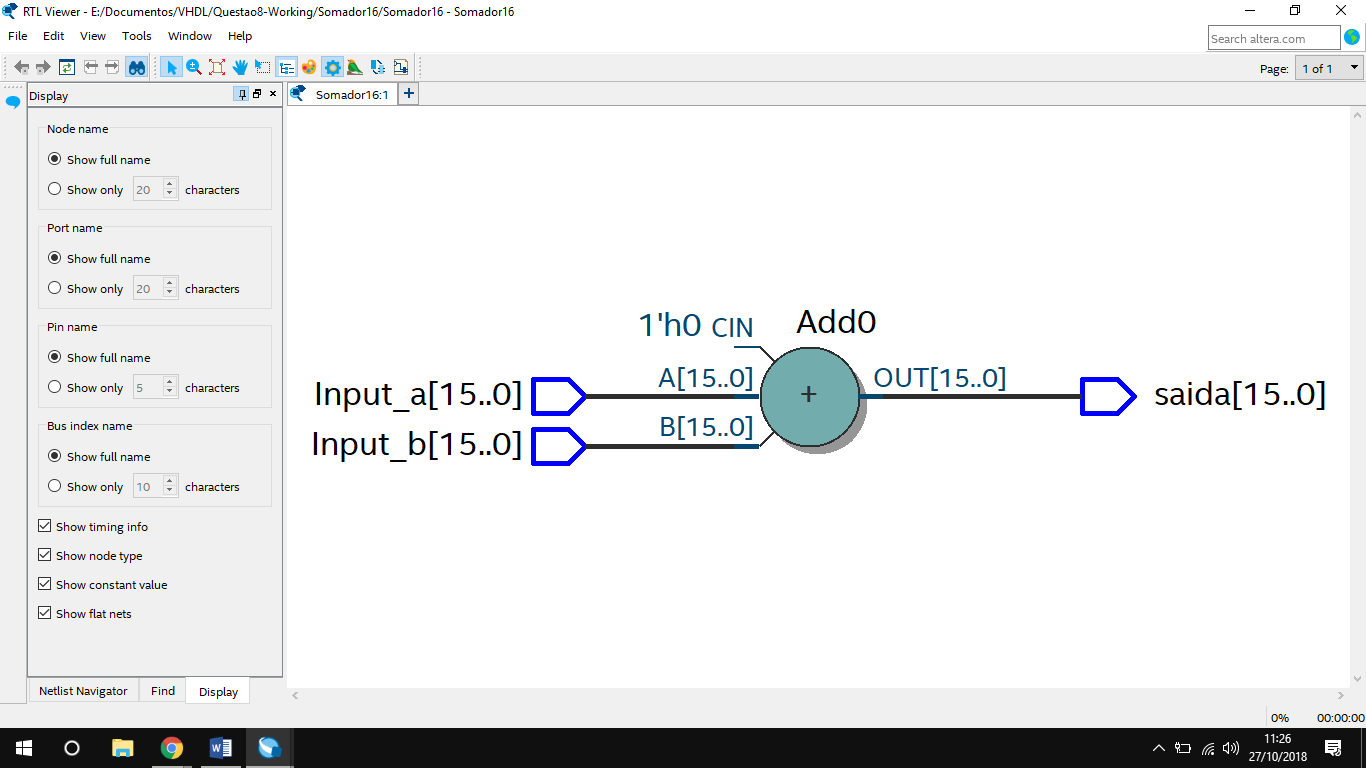
Um Banco de Registradores e um dos componentes mais importantes dentro de um sistema computacional, já que ele vai agrupar os registradores que são os principais responsáveis pela persistência e operação de dados. A implementação que fiz é bem parecida com a da memória RAM, onde teremos a porta WRITEFLAG indicando se irá ocorrer a escrita de dados, e três portas para endereçamento de memória.

# Questão [Somador 16 bits]

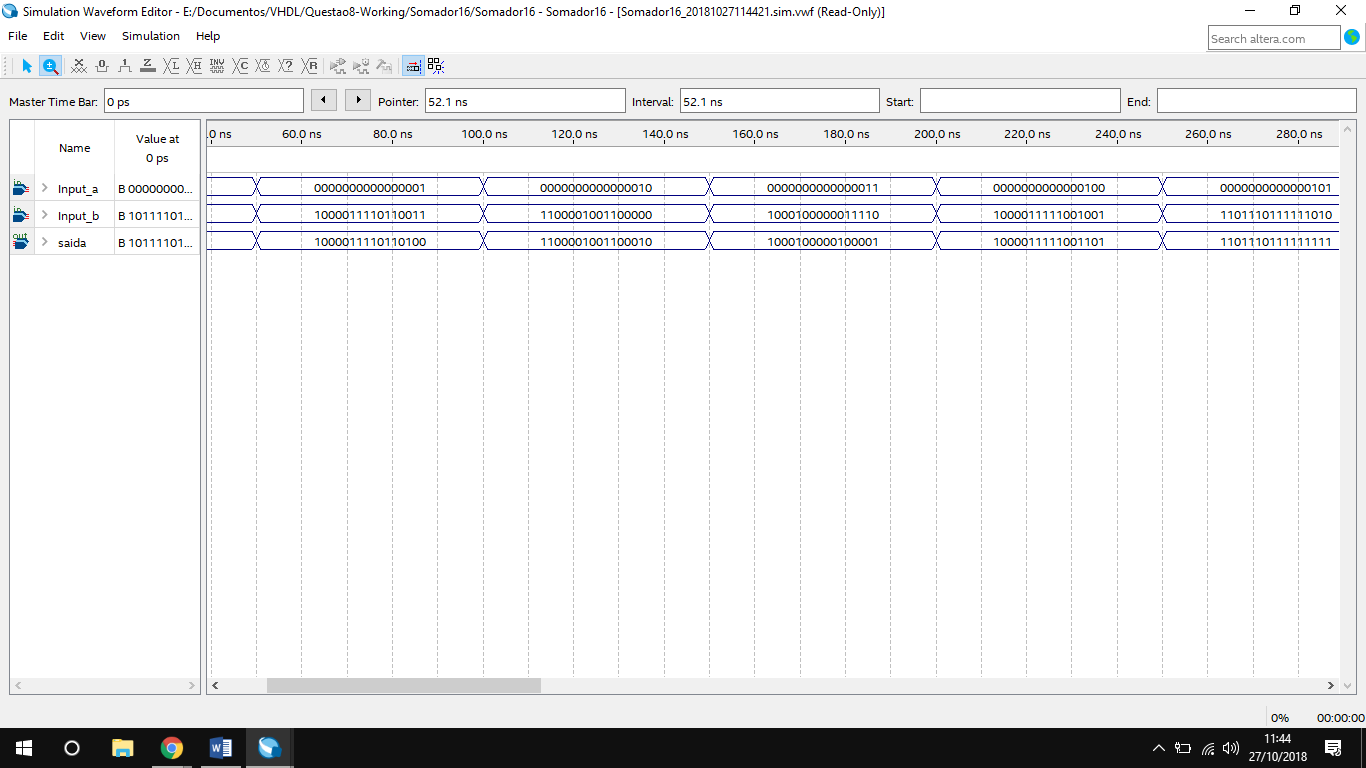
## Códigos



## RTL



## Wave Form



## Descrição

O Somador de 16 bits não tem muitos segredos é simplesmente atribuição de soma às duas entradas de dados.

# Questão [Unidade de Controle]

## Código

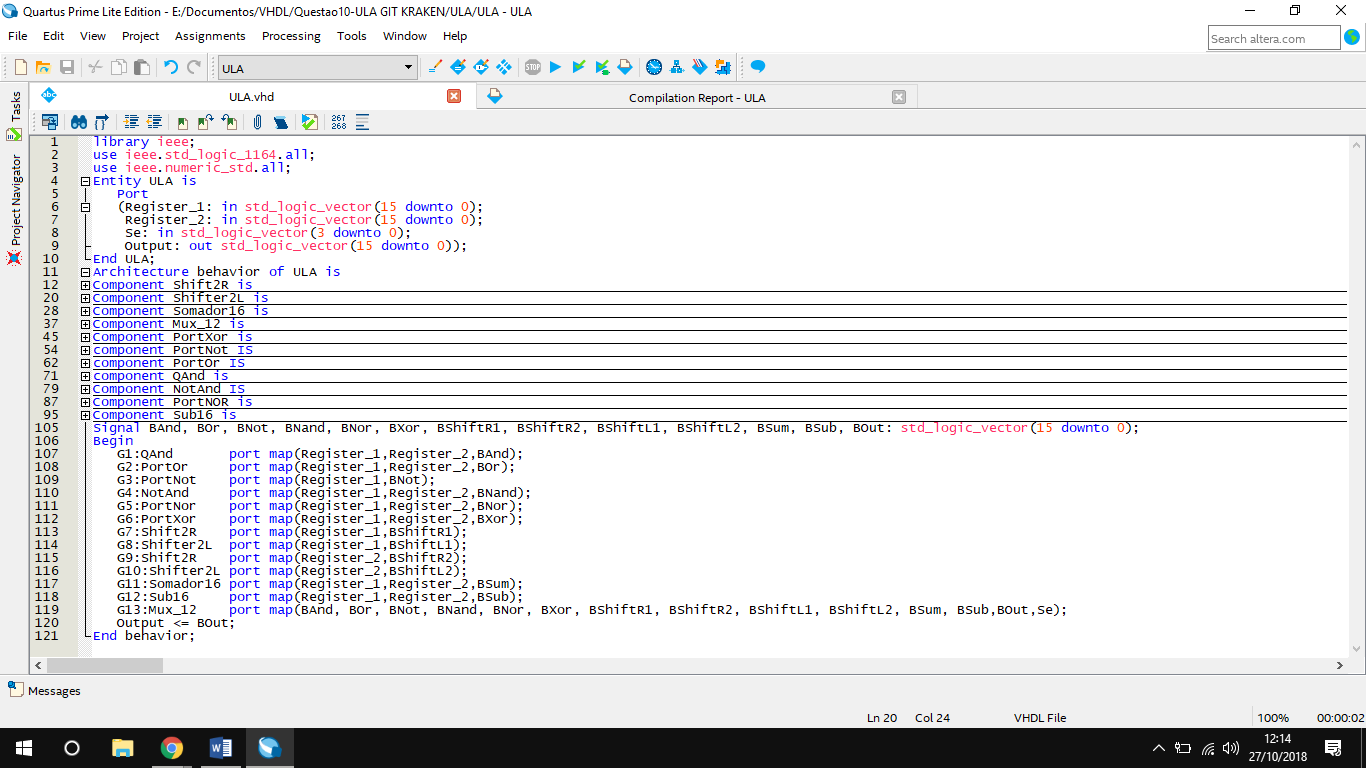
## RTL

## Wave Form

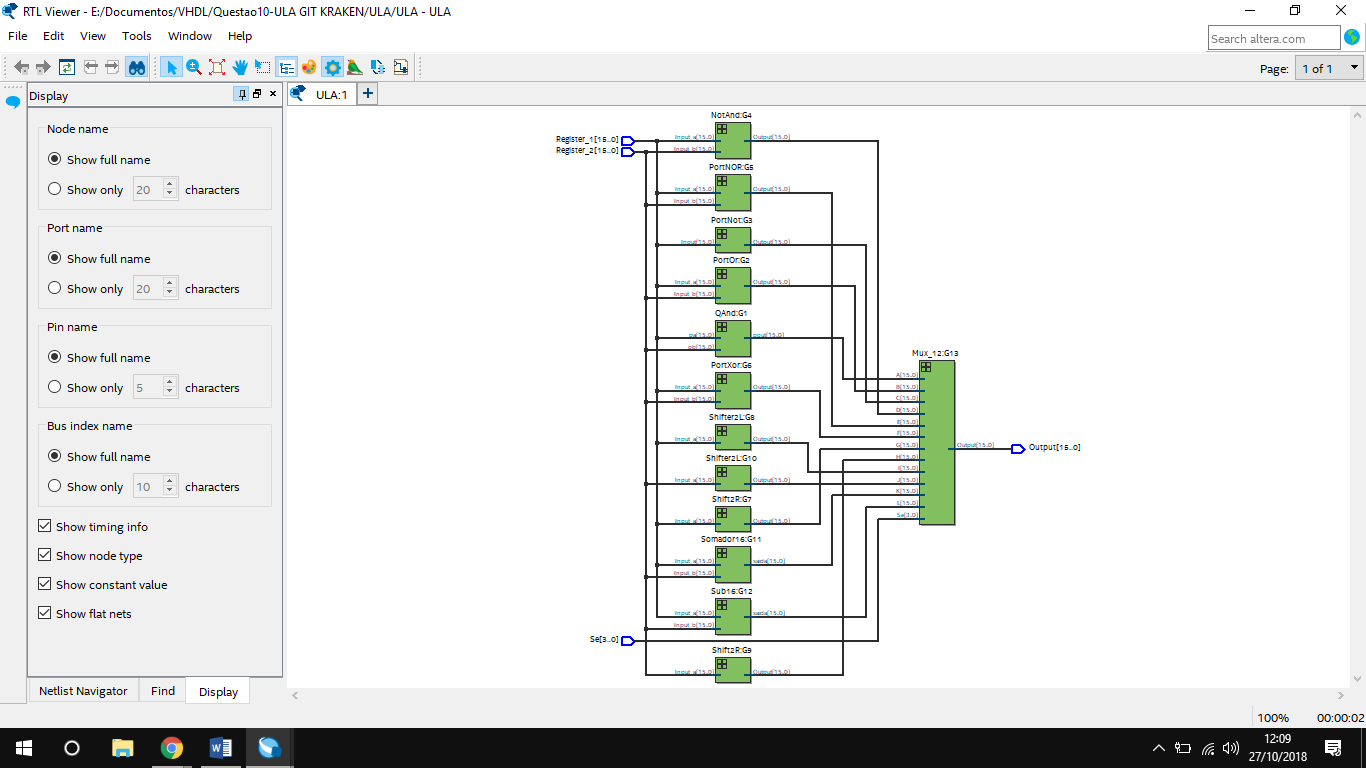
## Descrição

# Questão [ULA]

## Códigos



## RTL



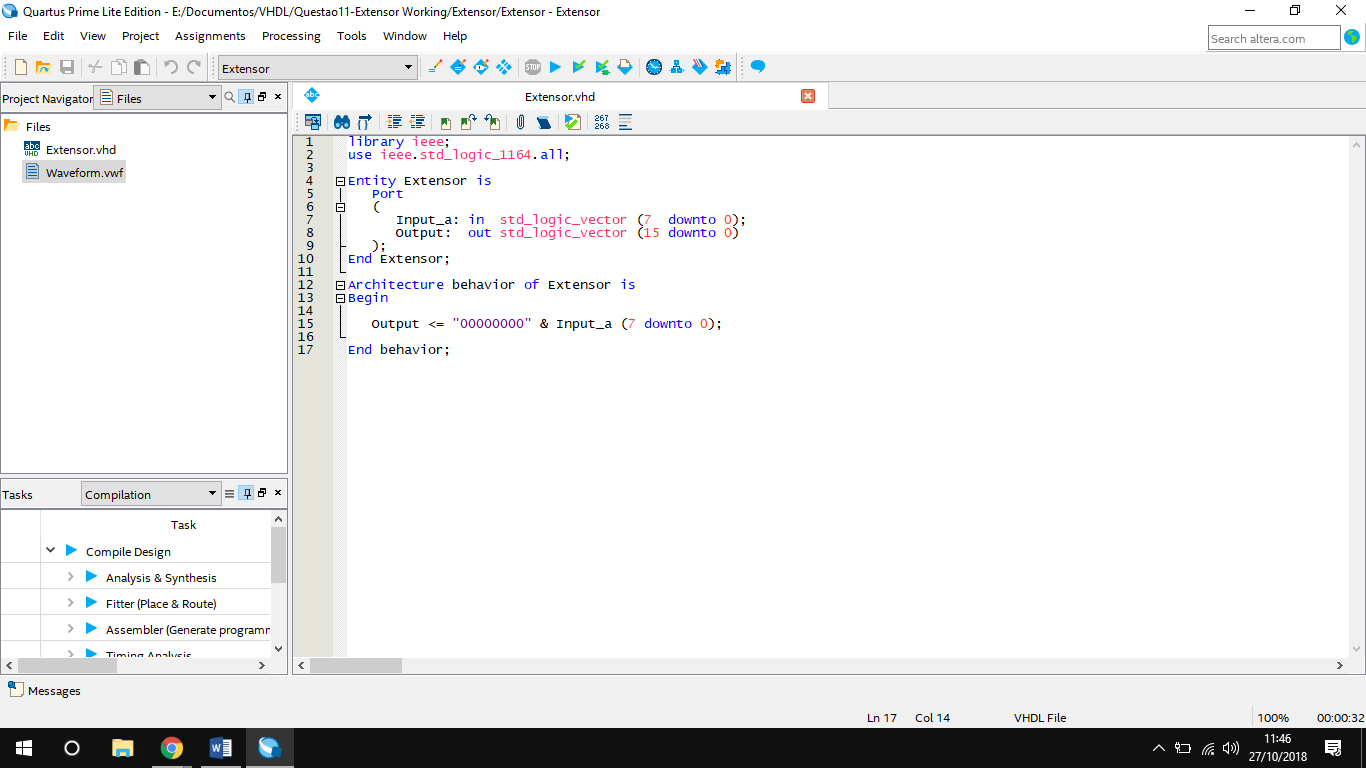
## Wave Form

## Descrição

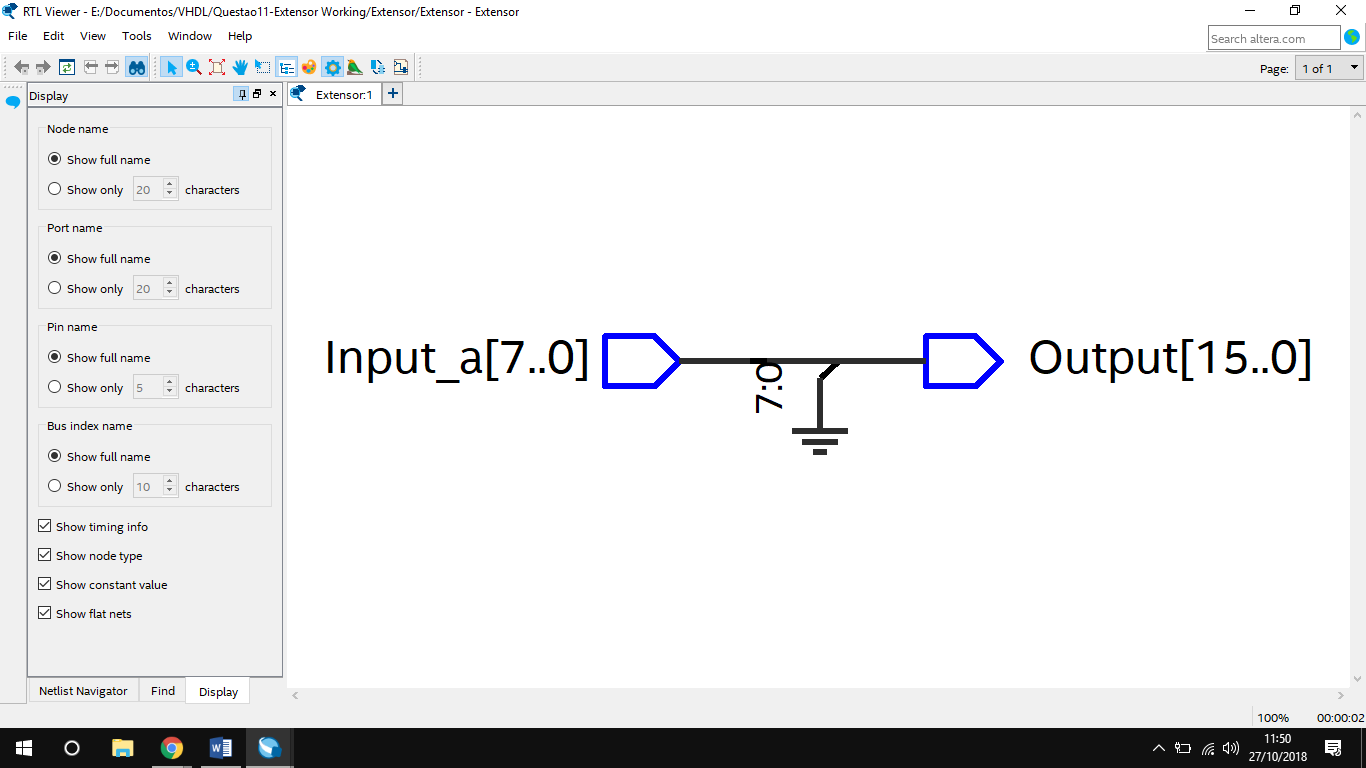
A ULA foi simplesmente a implementação Port Maps de componentes já feitos anteriormente e que foi colocado um multiplexador que irá receber todas as operações, e a partir da porta Se ele irá fazer a seleção da porta desejada.

# Questão [Extensor de Sinal]

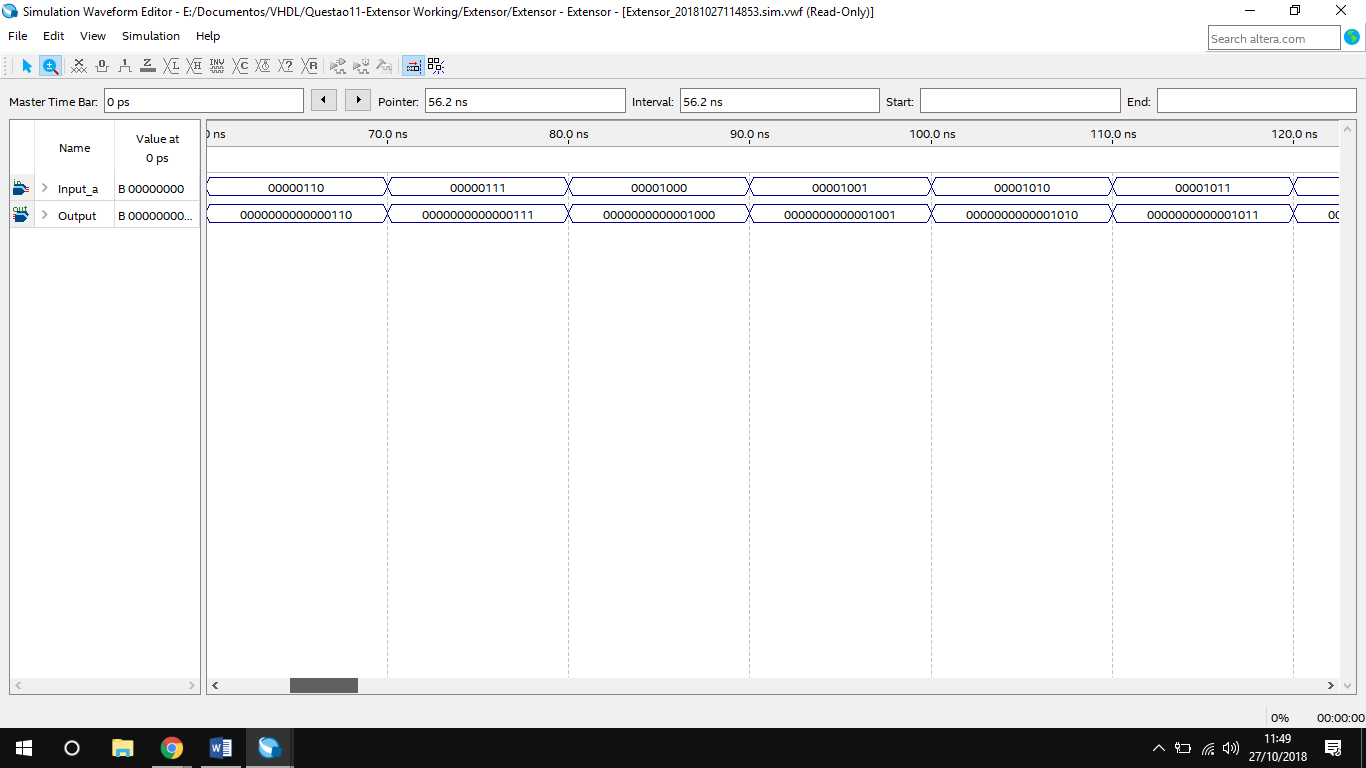
## Códigos



## RTL



## Wave Form



## Descrição

O extensor de sinal foi outro componente de simples implementação, onde ocorre apenas a concatenação de um vetor vazio de 8 bits ao vetor de 8 bits de entrada.

# Questão [Máquina de Estados]

## Códigos

## RTL

## Wave Form

## Descrição

# Questão [Contador Síncrono]

## Códigos

## RTL

## Wave Form

## Descrição