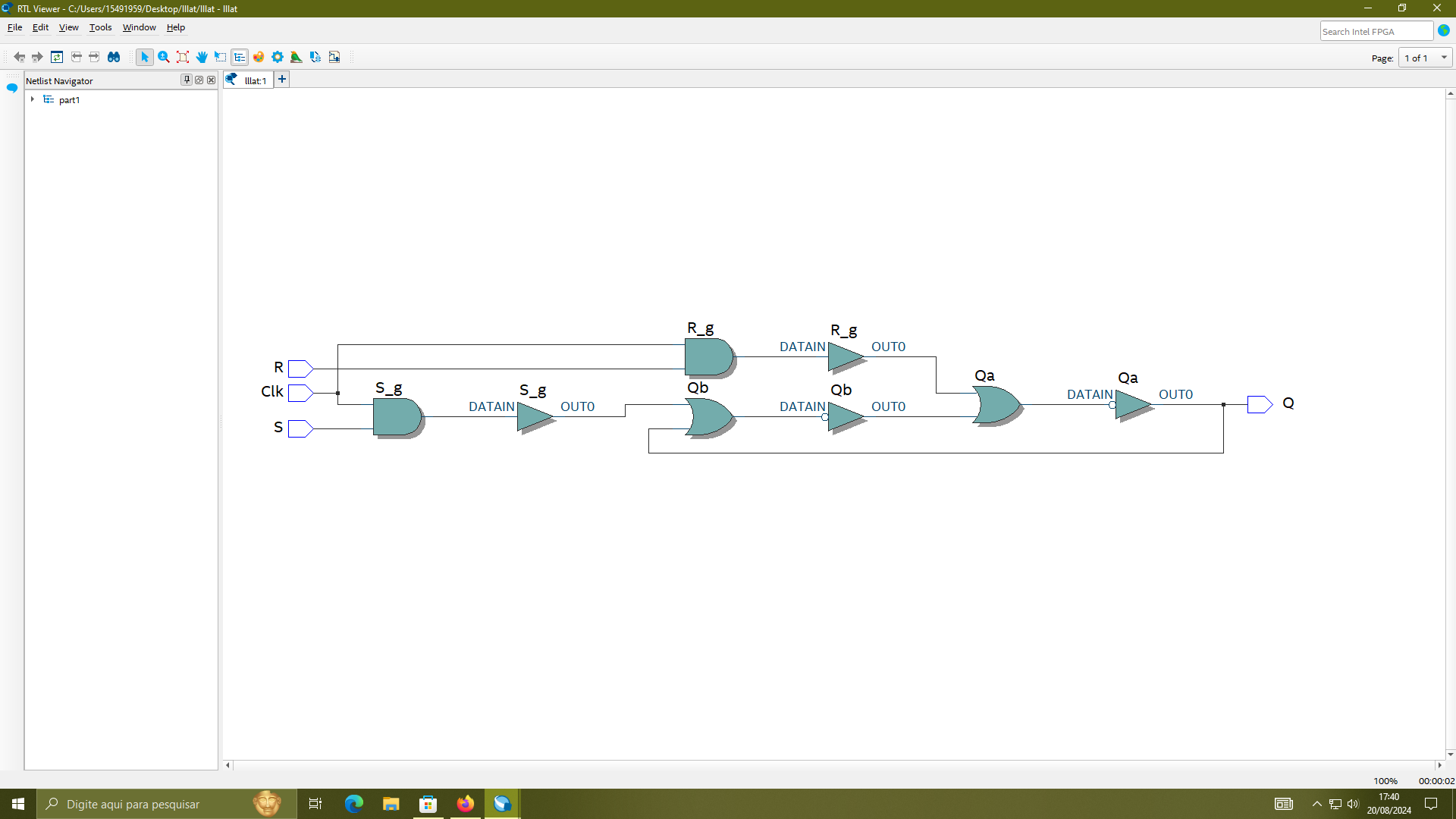
**Parte 1:**

Primeiramente é necessário configurar o Quartus normalmente.

Depois disso é necessário criar um vhd file, para fazer você vai em file e criar VHD, é importante lembrar que tem que colocar como top level para o Quartus saber que é para compilar aquele vhd.

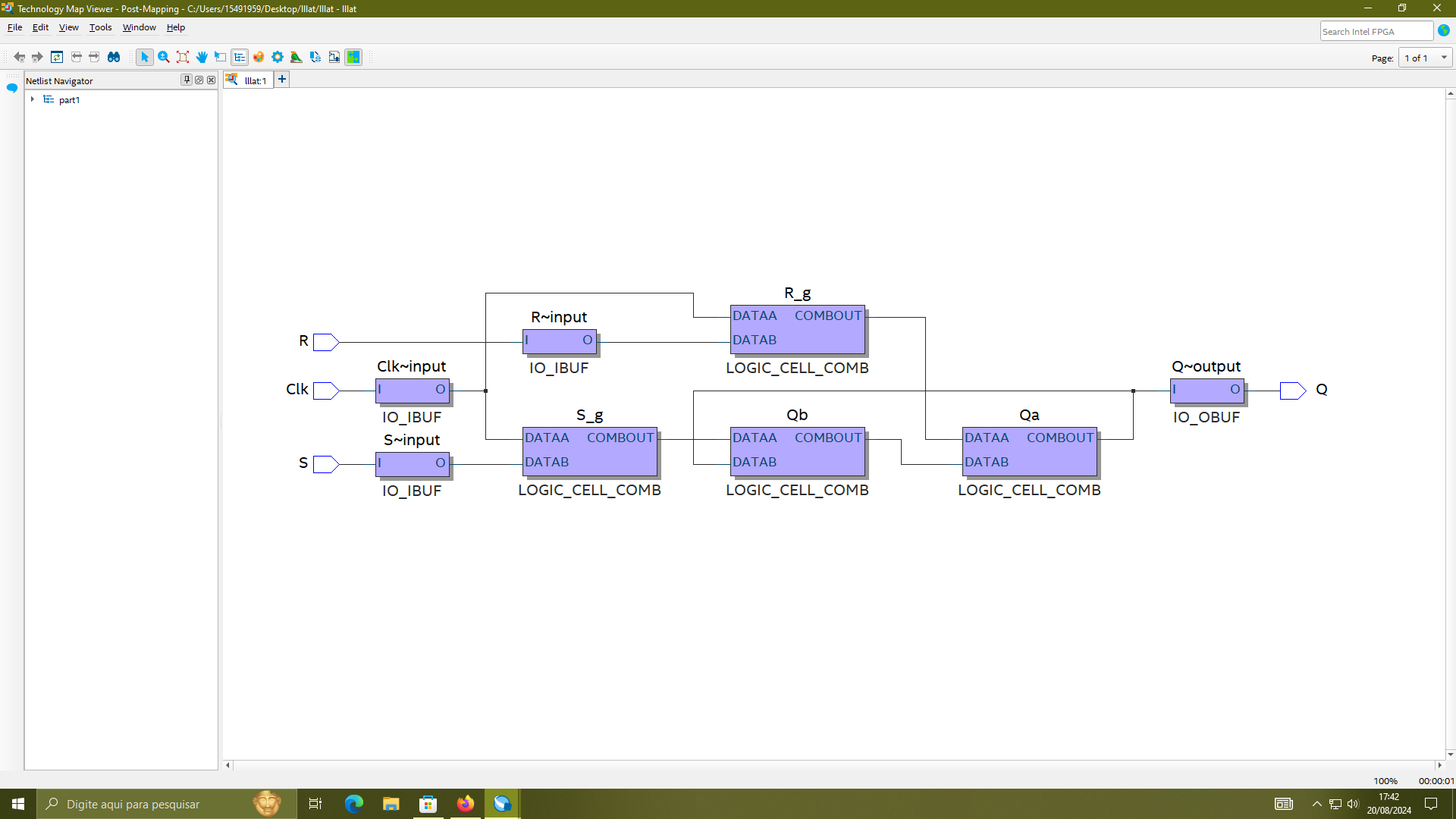
Para colocar top level tem que ir em “project”->”set as top level entity”.

Para visualizar ele como circuito acesse tools e “netlist-viewer”

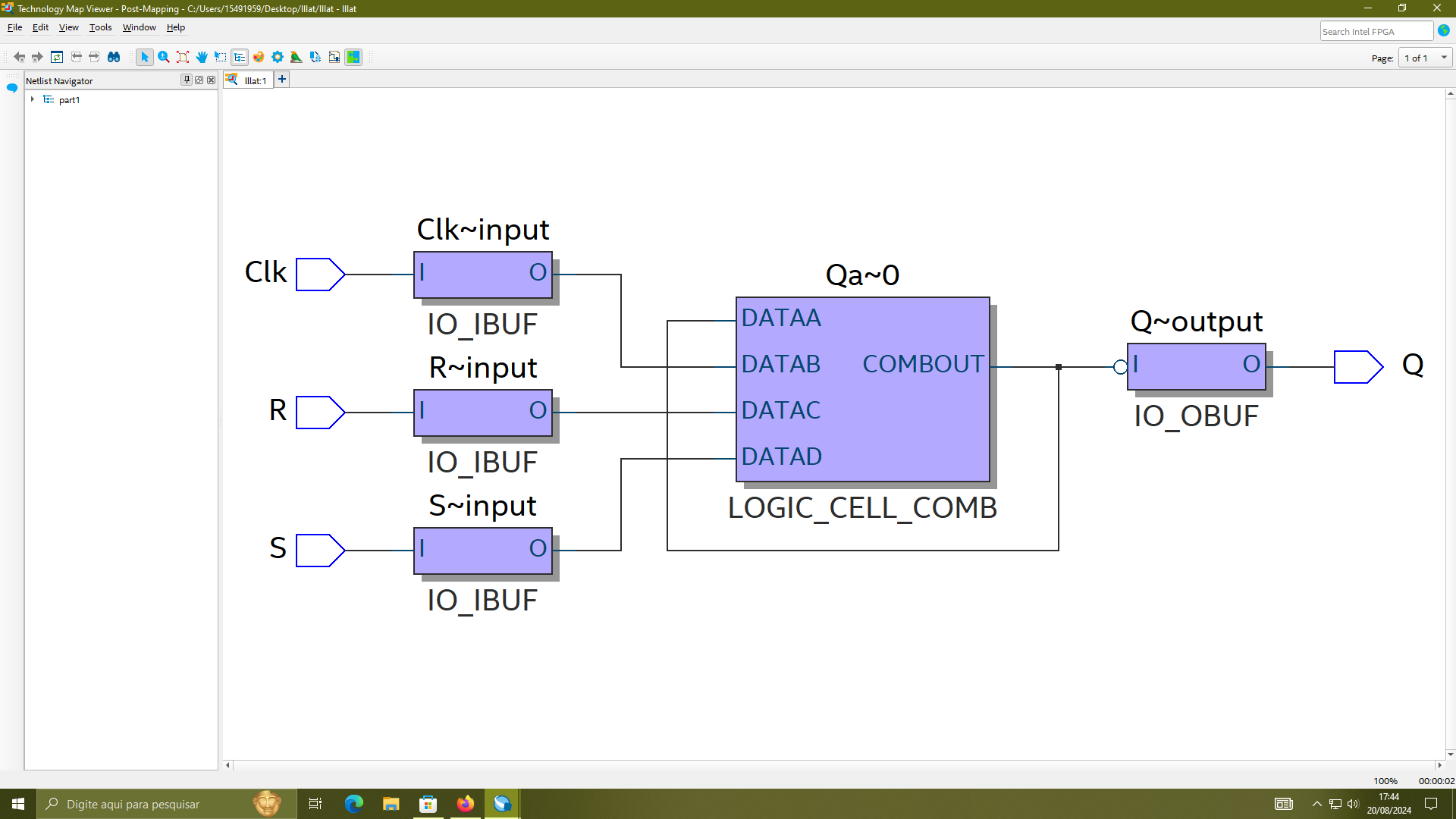


Esse circuito é uma implementação da gated RS latch usando NORS e ands, é a implementação padrão só que feita primeiramente em VHD.

A gated RS latch funciona como um latch normal, que armazena um bit, porém agora ela tem que é necessário que o clock esteja alto para que qualquer mudança seja efetuada sendo ela reset ou set. É importante lembrar que quando R e S são 0 o valor é indefinido (geralmente não permitido) e quando R e S são 1 o valor é o que era previamente, portanto, indefinido.

Esse é o circuito como forma de [LUTS](https://en.wikipedia.org/wiki/Lookup_table)As [LUTS](https://en.wikipedia.org/wiki/Lookup_table) (look up tables) são como as nossas tabelas verdades, basicamente a fpga assinala uma tabela verdade na qual cada uma das funções lógicas são implementadas em uma dessas [luts](https://en.wikipedia.org/wiki/Lookup_table), isso ocorre pois seria relativamente impossível implementar todas as funções booleanas.

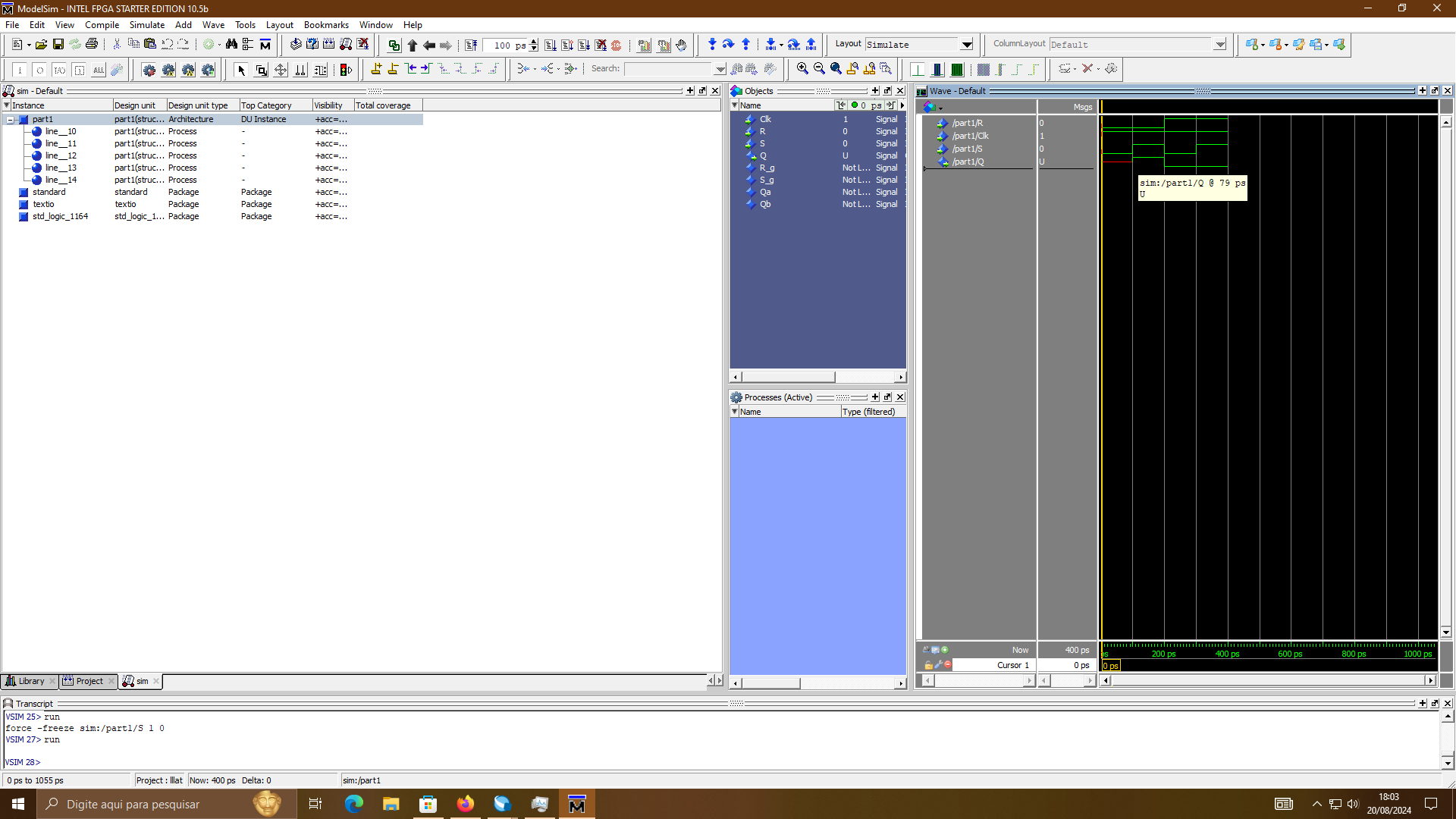
Essa imagem é a implementação em uma única Lut, só que agora não podemos ver o valor do and



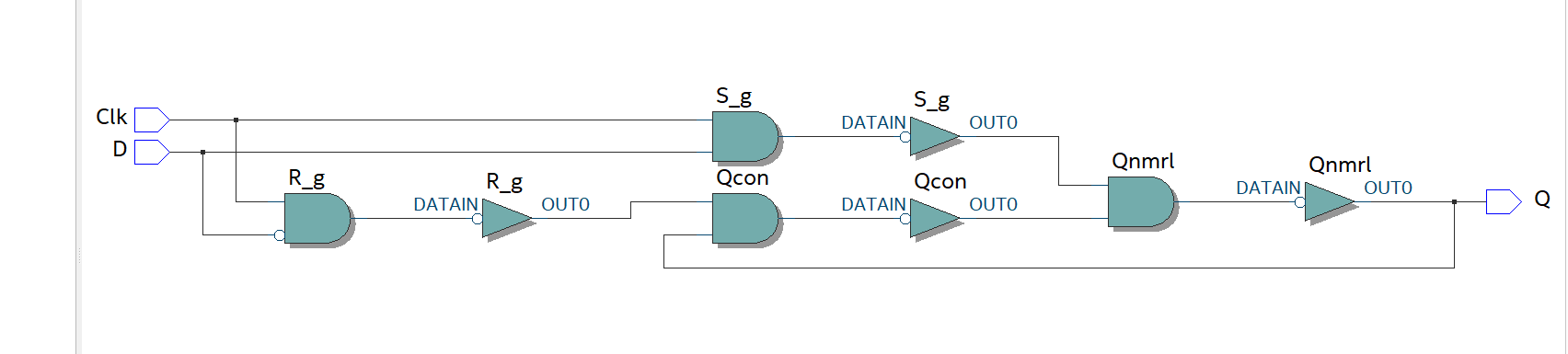
Agora é a parte do modelSim para iniciar você vai em file e cria um novo projeto, usa o vhd como componente do projeto

Depois, para simular é necessário compilar e depois ir em “simular”

Simulação da latch no ModelSim, é importante notar que quando o R é 0 e o S é 0 é indefinido.



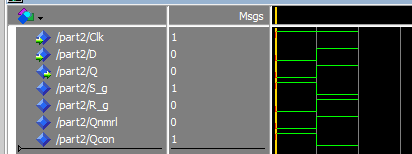
**Parte 2:**

****

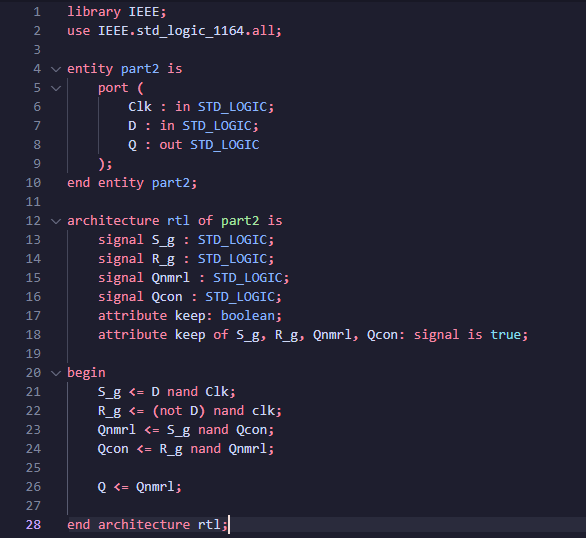
Este circuito é a implementação da D gated latch, que assim como a SR latch tem seu principal objetivo o armazenamento de memória. O seu funcionamento muda principalmente quando relacionado ao fato de agora ser controlado com apenas o “D”, quando o D for alto Q será alto também e seu conjugado será baixo, agora, quando D for baixo Q será baixo e seu conjugado alto.

Isso implica que se não formos mudar o resultado de algo não é mais necessário ficar “resetando” os valores das latches antes da operação, tudo depende do clock e do D.

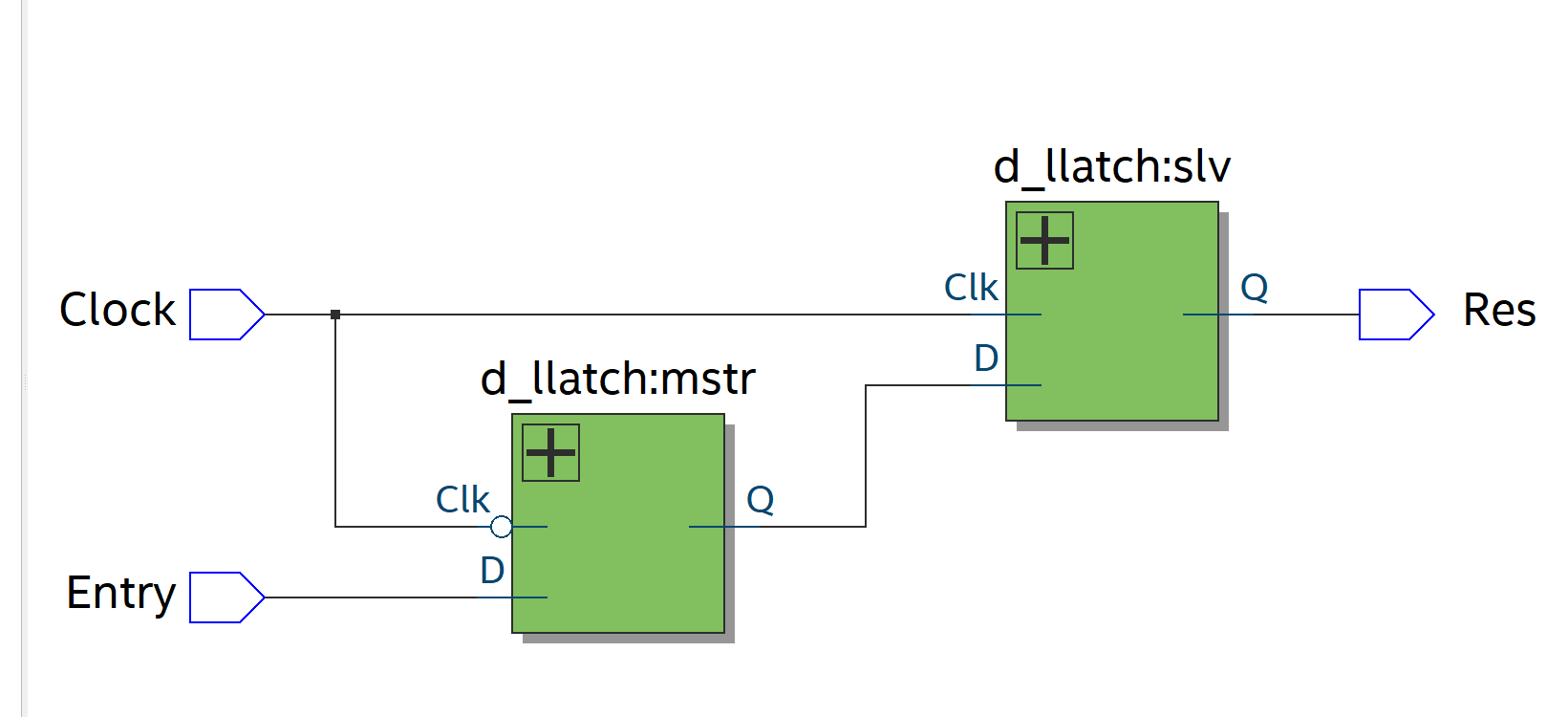
Simulação:



Código:

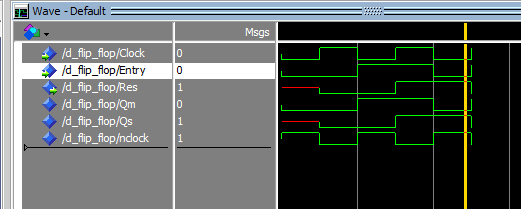


**Parte 3:**

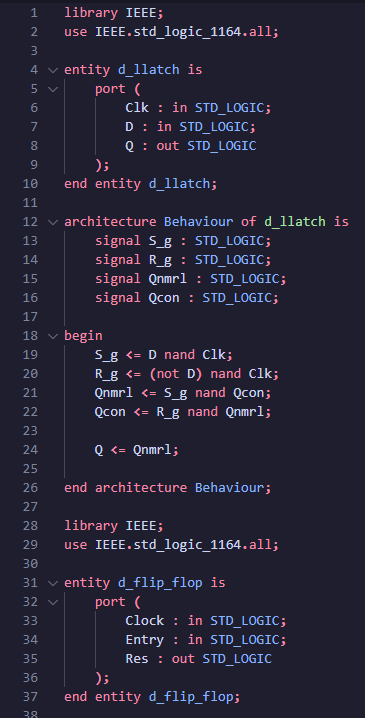


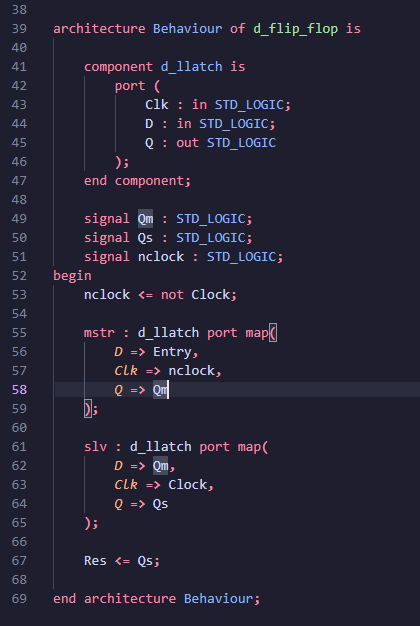
O circuito representa um master slave D flip flop, que é outro componente de armazenamento composto de duas D latches. O funcionamento deste flip flop é o seguinte: é possível modificar o estado do flip-flop quando o clock está alto, entretanto a última mudança feita só tomará efeito quando o clock estiver negativo e em função disso é chamado de positive edge.

Simulação:



Código:





**Part 4:**

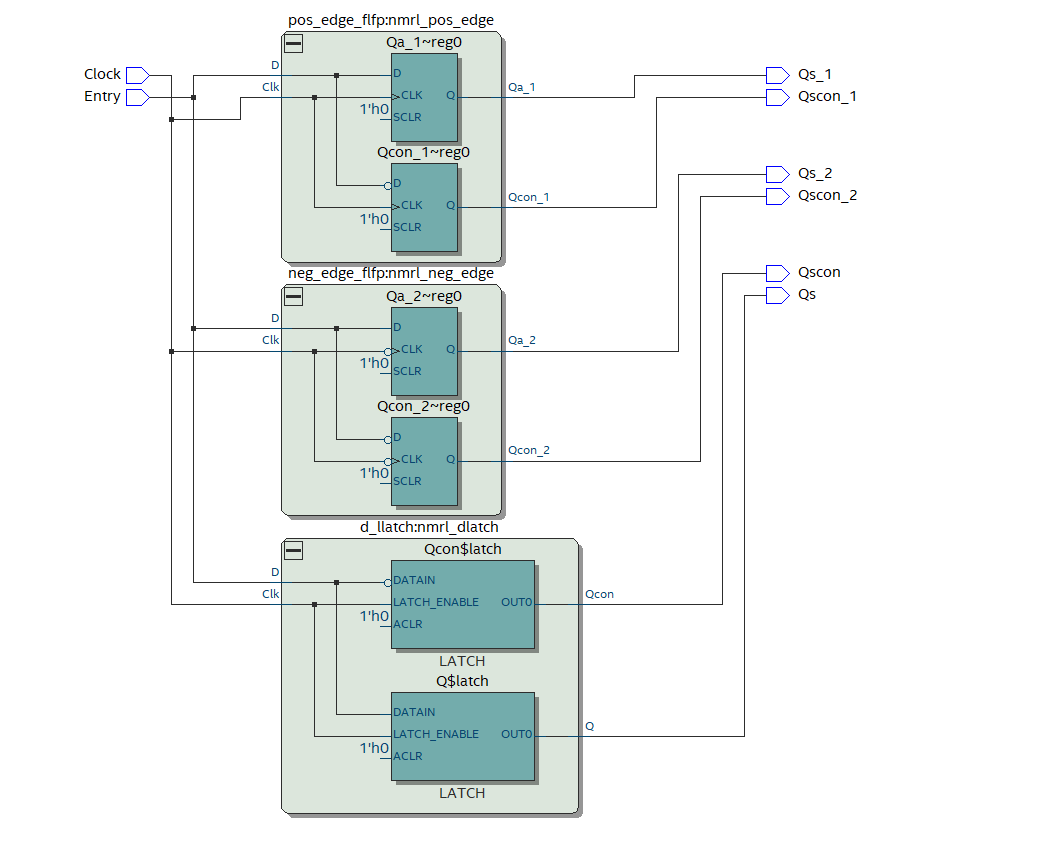
A parte 4 apresenta três circuitos para nós, os quais são variações em armazéns de memórias.

O primeiro circuito é o gated D latch normal o qual só muda seus valores quando o clock está alto.

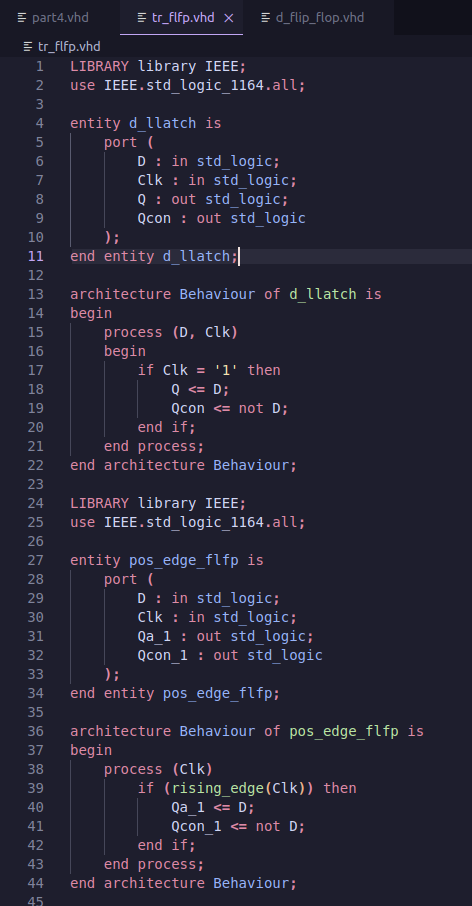
O segundo é o rising edge D flip flop o qual tem seu valor modificado apenas quando a onda está subindo. E mudando seu valor.

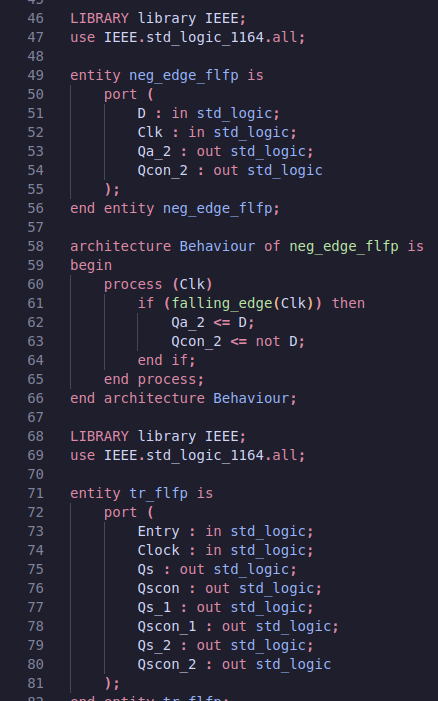
O terceiro é o gated falling edge D flip flop que tem seu valor modificado apenas quando o clock vai de um valor alto para um baixo. Assim armazenando seu valor.

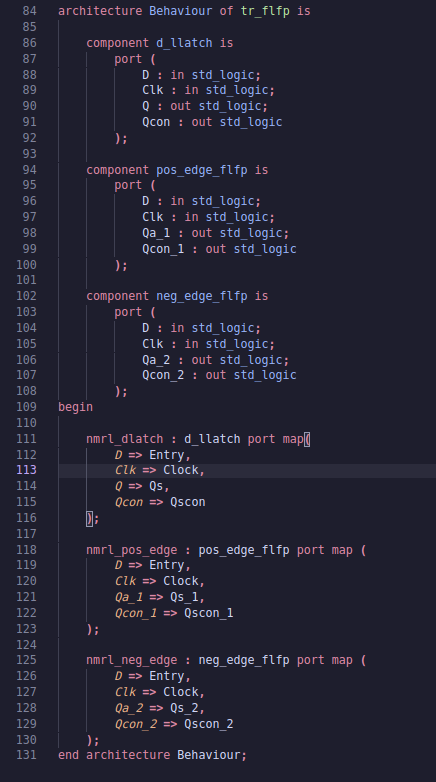
Aqui está a implementação na FPGA utilizando os registros dela:



Código deste 3 circuitos







Simulação do circuito no modelsim:

