

***LOGISIM: FERRAMENTA PARA SIMULAÇÃO DE CIRCUITOS
COMBINACIONAIS e SEQUENCIAIS DIGITAIS***

***LOGISIM: TOOL FOR SIMULATION OF COMBINATIONAL
AND DIGITAL SEQUENTIAL CIRCUITS***

***RAFAEL AUGUSTO ALBUQUERQUE MIQUELINI, HÉLIO
OLIVEIRA FERRARI***

RESUMO

Aprendizagem de lógica de circuitos tem como desafio a compreensão de uma proposição matemática e vê-la ser implantada. A prática da lógica por trás de um circuito eletrônico, digital ou não, pode ser complicada, ainda mais por lidar com abstrações, o que gera vários problemas, tanto para o aluno quanto para o professor. Uma das formas de contornar esse desafio, além do uso do laboratório de circuitos lógicos, é o uso de simuladores, sendo uma delas, o Logisim. Esse trabalho tem como objetivo demonstrar o uso dessa ferramenta de uma maneira acessível e didática, como um instrumento para a prática de projetos de ensino. Para a construção desse trabalho utilizou-se alguns exercícios do referencial teórico para verificar o funcionamento e testar os limites do programa. Chegando na conclusão que apesar dos limites do programa, principalmente na ferramenta de análise usada para flip-flops, suas vantagens, como ser gratuito, leve, modificável, compensam esse pesar, e pode sim entrar na lista de simuladores para ensino.

Palavras-chave: Material didático, simulador lógico, Logisim.

ABSTRACT

Learning circuit logic has the challenge of understanding a mathematical proposition and seeing it implemented. The practice of logic behind an electronic circuit, digital or not, can be complicated, even more so because it deals with abstractions, which creates several problems for both the student and the teacher. One of the ways to get around this challenge, in addition to using the circuit laboratory logical, is the use of simulators, one of which is Logisim. This work aims to demonstrate the use of this tool in an accessible and didactic way, as an instrument for the practice of teaching

projects. For the construction of this work, some exercises from the theoretical framework were used to verify the functioning and test the limits of the program. Coming to the conclusion that despite the program's limitations, especially in the analysis tool used for flip-flops, advantages, such as being free, lightweight, modifiable, make up for this burden, and can indeed be included in the list of simulators for teaching.

Keywords: *Courseware, logic simulator, Logisim.*

INTRODUÇÃO

O crescimento na aplicação do uso de simuladores no ensino, em diversas áreas do conhecimento, é uma realidade que já era notada desde o começo do século (Pedrosa, 2010). O que naquela época era uma percepção hoje é uma realidade, principalmente tratando-se de cursos à distância, onde o laboratório físico fora substituído por uma tela de computador, ou até mesmo de um smartphone.

Principalmente após a instauração do “caos” em que vivemos, devido à pandemia que assola essa nova década, analisando que, no Brasil, esse “novo normal”, onde estamos de quarentena, não temos acesso a laboratórios, e as aulas são em sua maioria à distância, o uso de ferramentas remotas para a educação se torna cada vez mais importante e necessário.

Outro motivo bastante importante para o uso de simuladores, é o custo, que querendo ou não, é melhor você testar em um simulador, onde não tem custo, até funcionar, do que ir direto para a bancada de testes, e poder queimar componentes.

Simulador ou simulação tem várias definições, que variam de autor para autor, mas que no final seguem o mesmo caminho. Um simulador é um aparelho ou um software capaz de reproduzir e simular o comportamento de algum sistema (Mendez, 2004). Simulação implica na modelagem de um processo ou sistema de tal forma que o modelo imite as respostas do sistema real numa sucessão de eventos que ocorrem ao longo do tempo (Schriber, 1974). Simulação é o processo de projetar um modelo de um sistema real e conduzir experimentos com este modelo com o propósito de entender seu comportamento e/ou avaliar estratégias para sua

evolução (Pegden, 1991). Tendo isso em vista, um simulador de um circuito lógico nada mais é do que um programa, onde os itens e ferramentas internas tentam imitar as interações desses equipamentos. Para a construção de um circuito lógico são necessários vários equipamentos e componentes, e alguns cursos de graduação podem não ter sempre esses itens em perfeito estado. Com esta necessidade surgiram-se softwares de simulação, os quais possibilita, não só o curso, mas como qualquer usuário a não obrigatoriedade da compra desses elementos.

Com o avanço da tecnologia, dá-se a criação de diversos simuladores, que vai desde gráficos no celular como o Geogebra, a simuladores mais complexos e completos, onde se consegue criar com fidelidade, circuitos elétricos, e afins, como no caso do Proteus. Os parâmetros de escolha para o software usado no teste deste artigo, foram acima de todos, acessibilidade, ou seja, qualquer computador, por menor que seja seu desempenho e independente do sistema operacional dele, consiga executar as ações do programa, e o escolhido foi o Logisim.

O Logisim é uma ferramenta usada para o estudo e simulação de circuitos lógicos. É um software gratuito, liberado sob os termos da **GNU – General Public License, version 2**, disponível nas plataformas, Linux, MAC OS X e Windows, foi programado inteiramente em Java, e utiliza a biblioteca de interface gráfica do Swing.

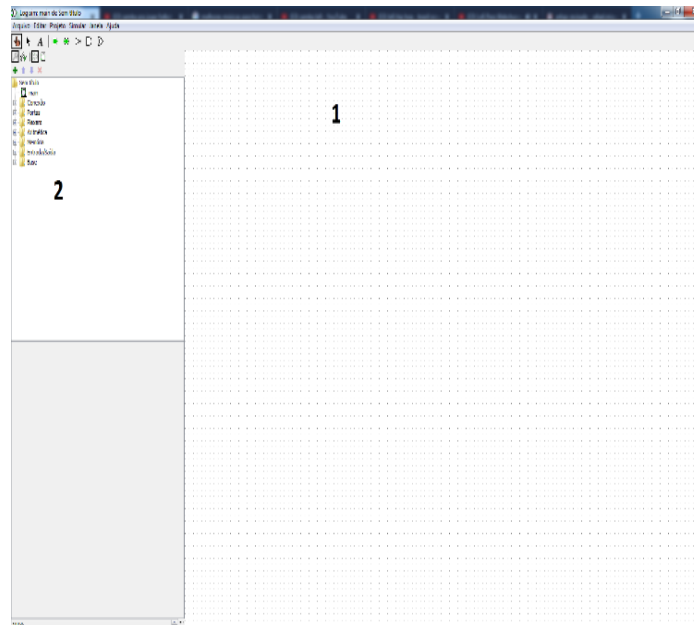
A análise deste simulador será feita a partir das propostas educacionais contidas no livro, Sistemas Digitais: Princípios de Aplicações, bibliografia padrão em cursos de sistemas digitais. Será feita uma análise de como funciona o flip-flop, que nada mais é do que uma das maneiras mais simples de demonstração de memória dentro de um circuito digital. Ele é de suma importância nesse trabalho, pois mostrará um dos limites do programa.

MATERIAIS E MÉTODOS

Logisim

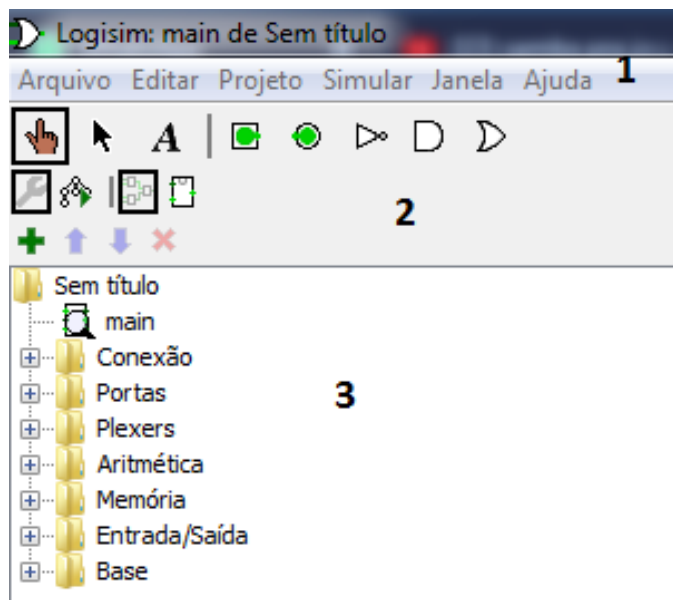
Como dito anteriormente, o Logisim é uma ferramenta usada para o estudo e simulação de circuitos lógicos.

Figura 1 – Tela inicial do Logisim



Fonte: Screenshot do programa Logisim, 2021

Figura 2 – Seção de ferramentas



Fonte: Screenshot do programa Logisim, 2020

Na **Figura 1**, se encontra a tela inicial do programa Logisim, sendo o espaço retangular (1) o espaço onde iremos criar os circuitos, e o espaço a esquerda (2) será o local onde se encontram todas as ferramentas para que consigamos criar esses circuitos.

A **Figura 2**, representa com mais detalhes as ferramentas para a criação e edição do circuito. Aqui encontramos as ferramentas de edição do próprio programa (1), como a opção de “salvar” ou “abrir”, as funções de seleção e acesso rápido (2), como ligar ou desligar um botão e as pastas com todos os componentes do Logisim (3) desde componentes simples, como botões até os mais complexos como teclado.

A versão utilizada neste artigo é a, logisim-win-2.7.1, encontrada no site oficial para fazer o download, <https://sourceforge.net/projects/circuit/>. Apesar de ter outras versões como a Italian Fork, ou o Evolution, que são versões com certos aprimoramentos, modificações e correções de certos erros, mas utilizaremos a original para mostrar que mesmo com ela, se é possível cumprir o objetivo do artigo.

RESULTADOS E DISCUSSÃO

Como dito anteriormente, o modo que usaremos para exemplificar o uso do Logisim, é a utilização de exercícios de lógica combinacional do livro (Tocci, 2008). O primeiro exercício é o desafio 3.48 e está localizado no capítulo 3. É um problema relacionado com a tabela verdade “Mostre como $x = AB\sim C$ pode ser implementado com uma porta NOR de duas entradas e uma porta NAND de duas entradas.”

Para esse desafio vamos precisar saber do funcionamento da porta NOR e da porta NAND, que nada mais são do que as portas OR e AND, com as saídas invertidas, tendo a seguinte tabela verdade:

Tabela 1

OR			NOR		
A	B	S	A	B	S
0	0	0	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	1	1	1	0

Nota: Tabela verdade demonstrando OR/NOR

Tabela 2

AND			NAND		
A	B	S	A	B	S
0	0	0	0	0	1
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	0

Nota: Tabela verdade demonstrando AND/NAND

A, B e S são respectivamente, duas entradas e uma saída, o valor do sinal é representado por 0 e 1, onde zero é sinal baixo, e 1 sinal alto, em outras palavras “0”, entende-se por desligado, e “1”, por ligado, e esses são os únicos estados que o sistema permite, não existindo um meio termo entre eles. Um jeito para se visualizar isso, é, “A” e “B” são interruptores, ou estão ligados (1), ou desligados (0), e “S” é a lâmpada, que está acesa (1), ou apagada (0).

OR e AND e seus respectivos NOR e NAND, são portas lógicas, que por programação, tem exatamente a função apresentada nas tabelas acima

Tendo essas tabelas em mãos e entendendo o funcionamento delas, prosseguimos com o exercício. $x = AB\sim C$ então sua tabela será

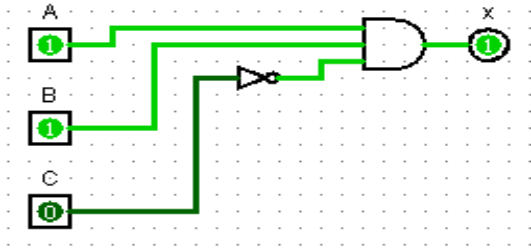
Tabela 3

A	B	$\sim C$	x
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Nota: Tabela verdade demonstrando $x = AB\sim C$

Figura 3

Seção de ferramentas

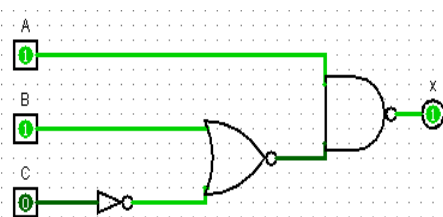


Nota: Screenshot do programa **Logisim, 2019**

Depois de várias tentativas e vários erros, após entender a teoria, não há porque continuar fazendo tabelas, e cálculos. Esforço repetitivo leva ao cansaço, que por sua vez, leva ao erro. Seres humanos, tendem mais facilmente ao erro. Máquinas, criadas como ferramentas auxiliam no dia-dia com tarefas repetitivas minimizando erro por esforço de repetição. Um chef de cozinha, para fazer claras em neve pode usar apenas um fouet, e com a força do seu braço e seu jogo de pulso, bate as claras até elas pegarem o ponto certo, para economizar tempo, para economizar o próprio chef, ele usa a batedeira orbital, que tem o mesmo resultado, as vezes até melhor, pois não há irregularidade em seu movimento. Temos o mesmo exemplo com circuitos lógicos, o estudante, que tiver entendido como uma tabela verdade, ou um mapa de Karnaugh funcionam, não precisa ficar repetindo o mesmo processo várias vezes, existem ferramentas para agilizar processos, e mitigar os erros.

No simulador o erro torna-se fonte de aprendizado através de inúmeras tentativas e erros. Para resolver o exercício proposto pode-se chegar no resultado final apenas “desenhando” o circuito, mas o não é isso que o enunciado quer, pelo menos não com essa resolução (Figura 3) e faremos como Freinet (2011) diz, através de tentativa e erro, mas sem tabelas, apenas criando um circuito e ir testando até chegar no resultado desejado (Figura 4).

Figura 4 - Circuito lógico 2



Nota: Screenshot do programa Logisim, 2019.

Nota: Imagem extraída do livro Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss – Sistemas Digitais_ Princípios e Aplicações-Pearson (2011)

Figura 5

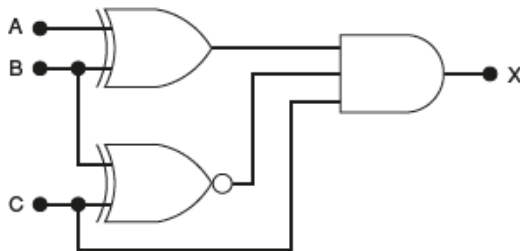


FIGURA 4.71 Problema 4.21.

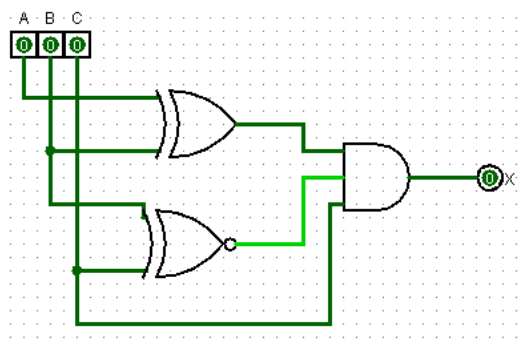
Circuito representando o problema 4.

O Problema 4.21 (Figura 5) que se encontra no capítulo 4 (Tocci, 2008) permite a investigação de outras potencialidades do simulador. Dado o enunciado do exercício: “Determine as condições de entrada necessárias para gerar uma saída $x = 1$ no circuito mostrado na Figura 4.71.”

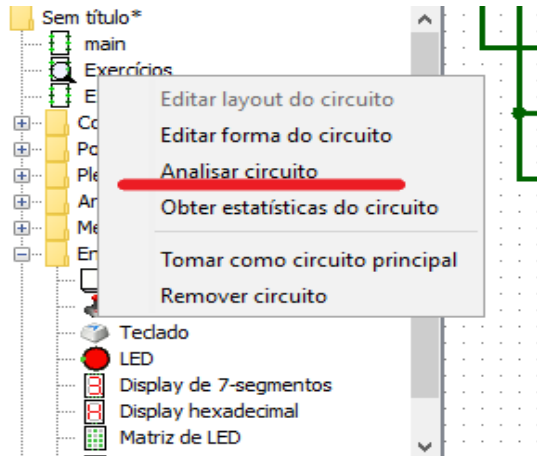
Tendo em vista que nesse ponto, quem esteja utilizando o Logisim, saiba o suficiente para executar apenas com algumas ferramentas dentro do programa. Então para a execução do exercício, vamos utilizar a forma rápida para resolver, vamos montá-lo no Logisim, depois iremos analisar o circuito para saber a resposta.

Figura 6

Circuito lógico 3



Nota: Screenshot do programa Logisim, 2019

Figura 7*Demonstração de ferramentas*

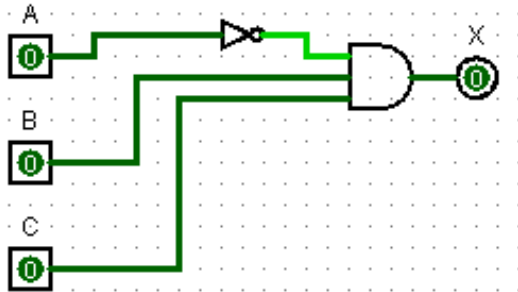
Nota: Screenshot do programa Logisim, 2019

Com ele devidamente “montado” iremos apenas analisar o circuito (figura 6). Com o botão direito do mouse clique no circuito, e selecione a opção marcada (Figura 7). Com isso irá aparecer a seguinte janela:(Figura 8) Note que, se pode navegar nos outros tipos de Análises, desde a tabela até ao mapa de K, mas nosso foco agora é a tabela (Figura 8). Olhando a tabela já se sabe a resposta do exercício, ou seja, só terá a saída “x=1” quando “A” estiver desligado, “B” e “C” estiverem ligados. Mas e se por curiosidades pedirmos para o Logisim construir outro circuito com essas informações?

Figura 8*Tabela Verdade*

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Fonte: Screenshot do programa Logisim, 2019.

Figura 9*Circuito lógico 4*

Fonte: Screenshot do programa Logisim, 2019.

Ele mostrará uma resposta simplificada (Figura 9). Ou seja, todo aquele circuito com portas combinacionais mais complexas, pode ser resumido em um circuito simples $A'BC$

Flip Flop

Uma porta lógica em si não tem capacidade de armazenamento, porém quando se é combinada e remanejada no conceito de realimentação, de certa forma, isso se torna possível, isso é o que chamamos de Flip Flop. Segundo (Tocci, 2008) flip flop é o elemento de memória mais importante, composto por um conjunto de portas lógicas.

Como o intuito desse trabalho é, além de mostrar a toda a praticidade e utilidade do programa Logisim, é também, mostrar seus limites, e como será mostrado posteriormente, o programa apresenta algumas dificuldades ao analisar um elemento de memória.

Existem modelos prontos de flip flops, localizados na pasta de memória (Figura 10).

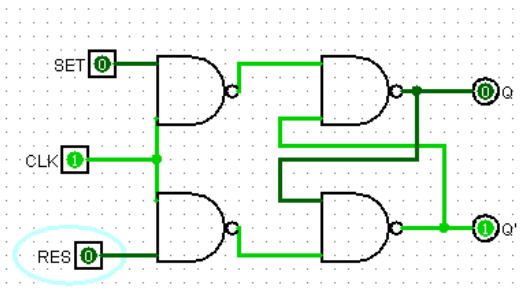
Figura 10*Alguns circuitos integrados existentes no Logisim*



Fonte: Screenshot do programa Logisim, 2019

Figura 11

Circuito lógico 5

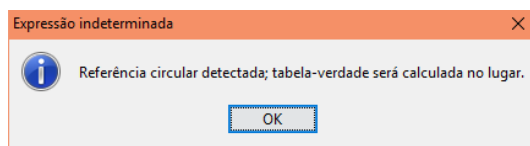


Fonte: Screenshot do programa Logisim, 2019

Porém, também é possível “cria-lo do zero”, usando algumas portas NAND (Figura 11), e para simular o clock, colocarmos um pino como entrada. Ao analisarmos o circuito, encontramos o seguinte quadro de aviso:

Figura 12

Notificação de erro



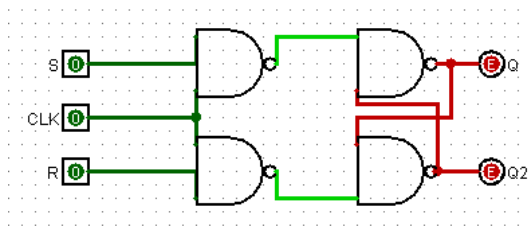
Fonte: Screenshot do programa Logisim, 2019

Basicamente, esse erro aparece, pois, dependendo da combinação o circuito estará logicamente errado, é impossível ele realizar essa sequência lógica, pois a mesma não existe.

Figura 13*Tabela verdade com erros*

SET	CLK	RES	Q	Q2
0	0	0	!!	!!
0	0	1	!!	!!
0	1	0	!!	!!
0	1	1	0	1
1	0	0	!!	!!
1	0	1	!!	!!
1	1	0	1	0
1	1	1	1	1

Fonte: Screenshot do programa Logisim, 2019.

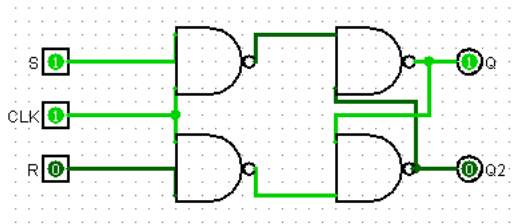
Figura 14*Circuito lógico 6*

Fonte: Screenshot do programa Logisim, 2019.

A tabela verdade acima mostra o circuito no momento em que ele é “criado”, ou seja, até que faça alguma combinação de entradas que dão uma saída, que não seja erro, ele continuará estático (Figura 14).

Ao usar uma das combinações, o circuito começa a se comportar conforme um flip-flop, onde S é o SET, ou seja, o botão que força o a saída Q ligar e desligar a Q2. R é o RESET, que é o inverso do SET, ou seja, desliga Q e liga Q2. CLK é a simulação do CLOCK, só que nesse exemplo, eu controlo quando ele estará emitindo valores altos ou baixos, diferente de um clock usual, que alterna esses valores automaticamente.

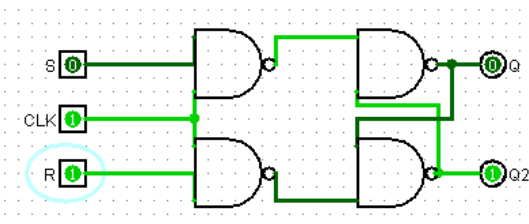
Figura 15*Circuito lógico 7*



Fonte: Screenshot do programa Logisim, 2019.

Figura 16

Circuito lógico 8

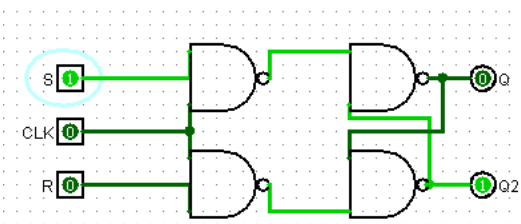


Fonte: Screenshot do programa Logisim, 2019.

Esse flip-flop, é um dos vários tipos, e alteram o resultado apenas quando o clock está em valor alto.

Figura 17

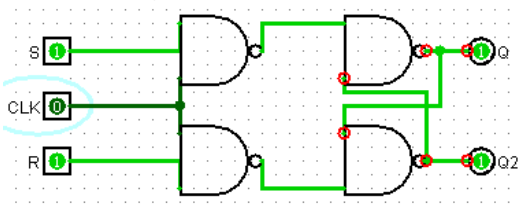
Circuito lógico 9



Fonte: Screenshot do programa Logisim, 2019

Figura 18

Circuito lógico 10



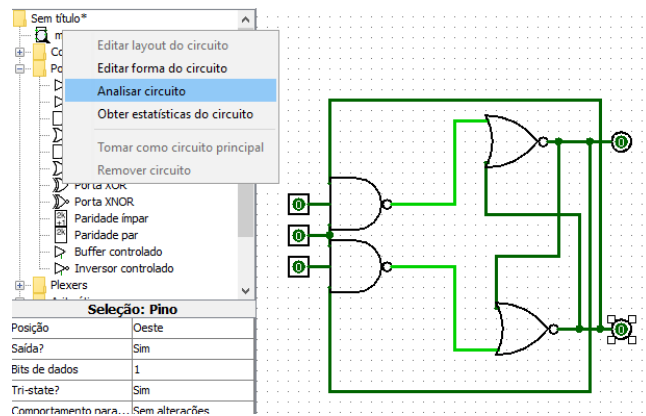
Fonte: Screenshot do programa Logisim, 2019.

Se for deixado em valor baixo, a saída não modificará até independente de S e R. Ao desligar o CLK, quando S e R estiverem ligados, o programa irá parar de funcionar, pois houve uma incompatibilidade lógica (não achei expressão melhor). Sendo necessário a reinicialização do mesmo.

Circuitos muito complexos como o flip-flop JK, podem ser montados, porém não analisados.

Figura 19

Circuito lógico 11



Fonte: Screenshot do programa Logisim, 2019.

Ao tentarmos analisá-lo, o programa trava. Este travamento foi verificado em mais de um computador na utilização dessa versão do Logisim. Mais do que um evento que possa gerar frustração, abre-se uma oportunidade para se ter uma oportunidade futura de elucidar a razão desses travamentos.

CONSIDERAÇÕES FINAIS

Conclui-se que apesar de ter este limite com o flip-flop, como o Logisim é um software gratuito e livre, existem várias versões do programa, onde alguém acrescentou algo diferente, seja uma correção e otimização de todos os elementos, ou até mesmo, um elemento a mais visto que sua biblioteca é expansível. Então levando em conta o que vários autores Pedgen (1990), Schriber (1974), Mendez (2004) dizem sobre simulação, e analisando todos principais pontos do Logisim, ele cumpre bem sua função de simulador lógico.

REFERÊNCIAS

FREINET, C. (2011). **Para uma escola do povo: guia prático para a organização material, técnica e pedagógica da escola popular.**

LÉVY, P. (1993). **Tecnologias da inteligência**, As. Editora 34.

LogiSim. Disponível em: <http://www.cburch.com/logisim/pt/index.html>. Acesso em 13/09/2020.

MENDES, M. A., & FIALHO, F. A. (2004, April). **Avaliação de Simuladores Aplicados na Educação Tecnológica a Distância**. Salvador, BA: XI Congresso Internacional de Educação a Distância.

PEDROSA, E. F. (2010). **Simulated environment for robotic soccer agents**. Master, Aveiro University, Portugal.

PEGDEN, C.D., SHANNON, R.E., SADOWSKI, R.P. **Introduction to Simulation Using SIMAN**, McGraw-Hill, New York, USA. v. 2. 1990.

SCHRIBER, Thomas J. **Simulation Using Gpss**. Nova Iorque: John Wiley & Sons, 1974.

TOCCI, R. J., Widmer, N. S., & Moss, G. L. (2008). **Sistemas Digitais, princípios e aplicações**. 10a edição.

AUTORES

Rafael Augusto Albuquerque Miquelini, graduando em engenharia elétrica, UEMG – Unidade Ituiutaba.

Rafael.miquelini@gmail.com

Hélio Oliveira Ferrari, graduado em engenharia elétrica (UNIFEB), doutor em engenharia elétrica (UFU), professor da UEMG – Unidade Ituiutaba, pesquisador do GPETEX - Grupo de Pesquisa em Engenharia, Tecnologia e Ciências Exatas.

helio.ferrari@uemg.br