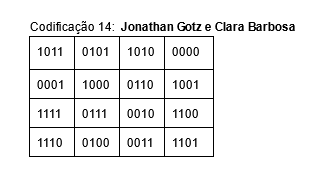
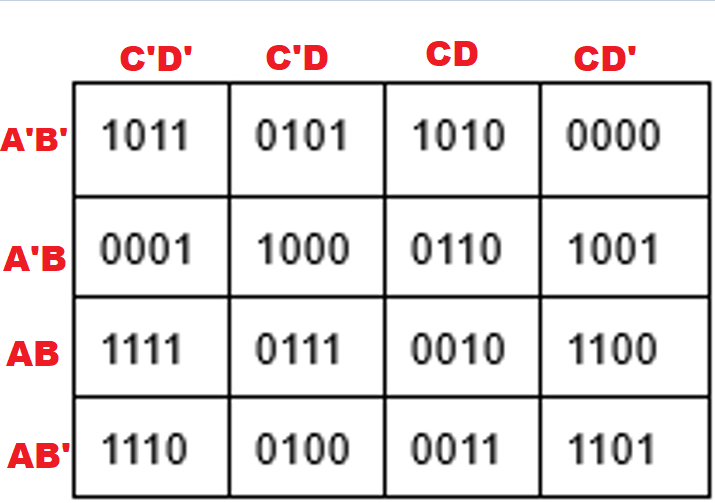
**Apresentação**

Eu, **Jonathan Gotz Correa** (2121101052) junto com **Ana** **Clara Barbosa Brusamarello** (2121101054), trazemos o projeto de simulação do famoso jogo de batalha naval, proposto em sala de aula como trabalho final e implementado primeiramente via Logisim ¹, e por fim através do Tinkercad ².

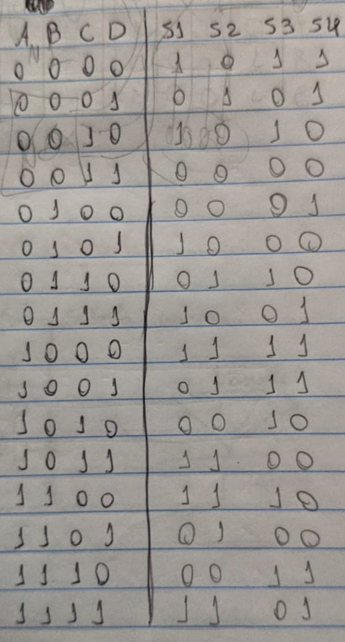
A solução do começo ao final da produção do projeto começa com a produção da tabela verdade em papel usando como referência a codificação de onde estariam os navios e a localização deles, presente na figura\_1.

Figura\_1:

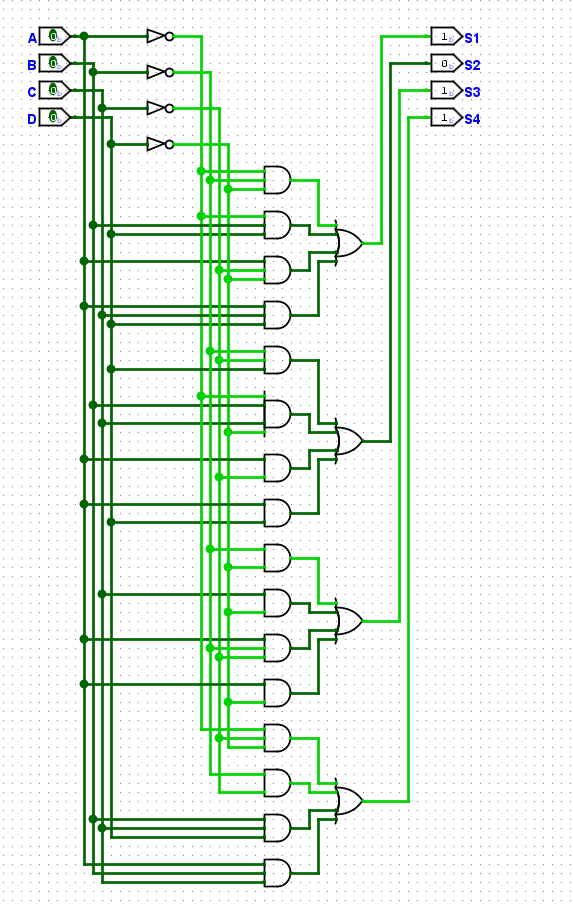
Pegando a lógica desta tabela e comparando com o mapa de karnaugh de 4 elementos as linhas se equivalem aos valores figurativos “AB” e as colunas, respectivamente, aos valores “CD”, conforme figura\_2.

Figura\_2:

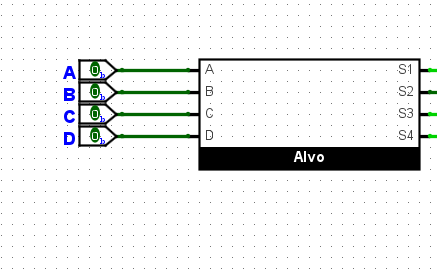
Formulando então a tabela verdade, temos:

Figura\_3:

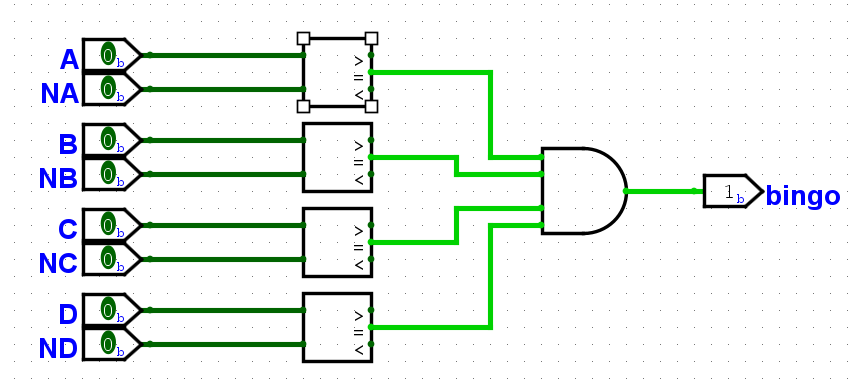
Colocando a seguinte tabela no logisim, é apresentado o seguinte circuito com 4 bits cada, que representa no circuito num todo o sistema de alvo ³, conforme figura\_4.

Figura\_4:

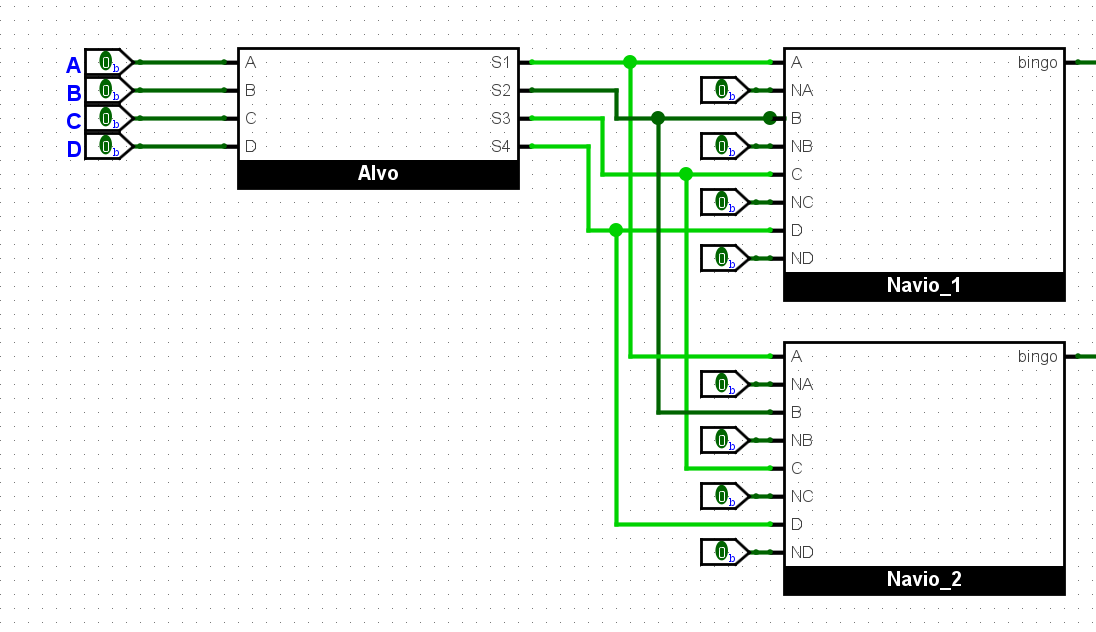
Para passar tal circuito para o tinkercad da forma mais prática é pegar bit por bit e refazer as portas AND e OR com mais de 3 entradas de forma prática visto que não há CI’s com 4 entradas, por exemplo. No logisim essa parte ficaria intocada dentro de um circuito único, que se une ao circuito principal, denominado “main” no arquivo, logo é necessário adicionar o sistema dos navios que receberão tal informação do alvo e a informação de onde está o navio, enviando essas duas informações para um circuito que tem como função informar se acertou ou errou, a partir de leds. Até agora temos o seguinte circuito, presente na figura\_5:

Figura\_5:

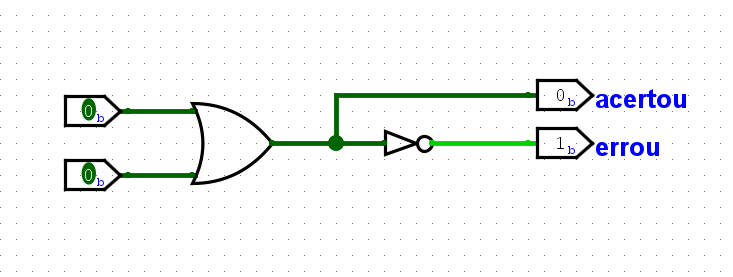
O circuito do navio é particularmente simples, utilizando quatro comparadores, que, cada um, recebe duas entradas, dentre essas uma é a saída do alvo de um dos bits e a outra entrada é a coordenada do navio, todos combinados enviam a informação a uma saída passando por um OR, denominamos no projeto como “NX” porque “CX” poderia confundir nas portas C, circuito explicado presente na figura\_6.

Figura\_6:

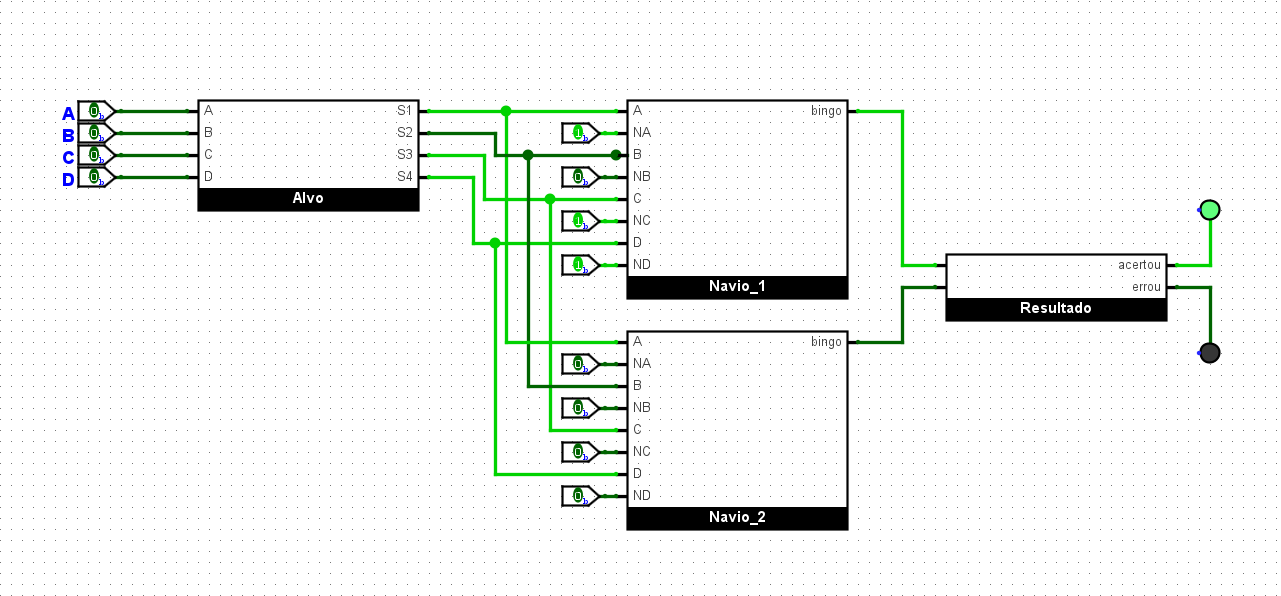
Temos então o seguinte circuito na figura\_7:

Figura\_7:

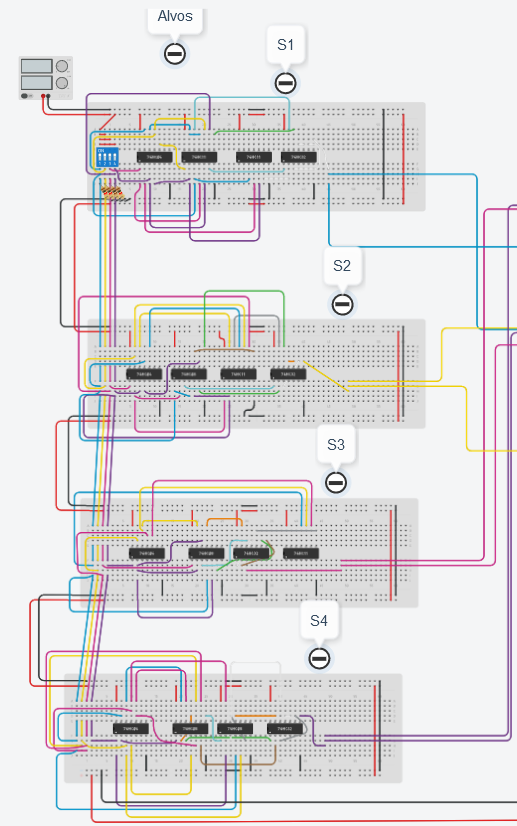
Por fim, essa saída dos dois navios com a informação de qual foi o lugar mirado para acertar um barco e qual era a localização no mapa em que o barco se encontra, basta fazer o circuito que informa através de LED’s se houve um acerto ou não. Bastando pegar um OR nas duas entradas dadas pelos navios, já que se só é preciso acertar um dos navios, a LED de acerto recebe o valor puro, a LED de erro converte negativamente o valor, assim dando o resultado coeso, presente na figura\_8.

Figura\_8:

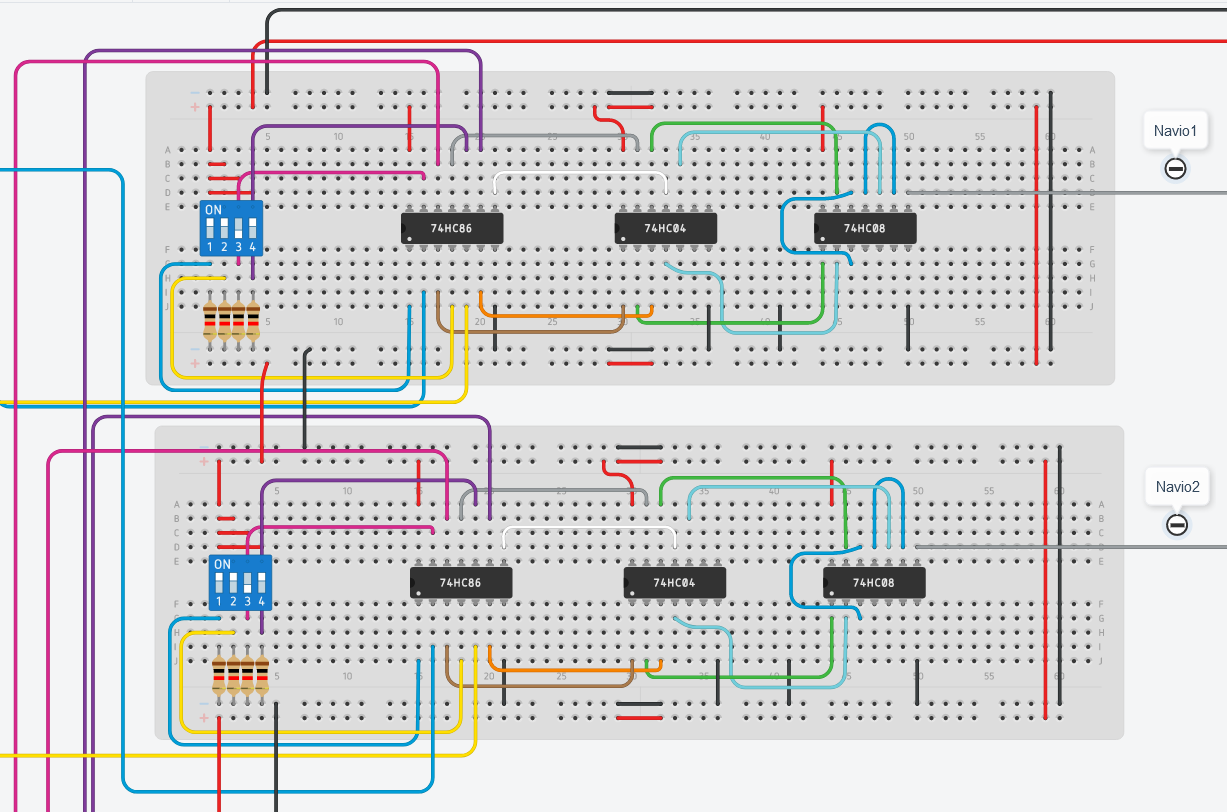
Por fim, o circuito final é o seguinte:

Figura\_9:

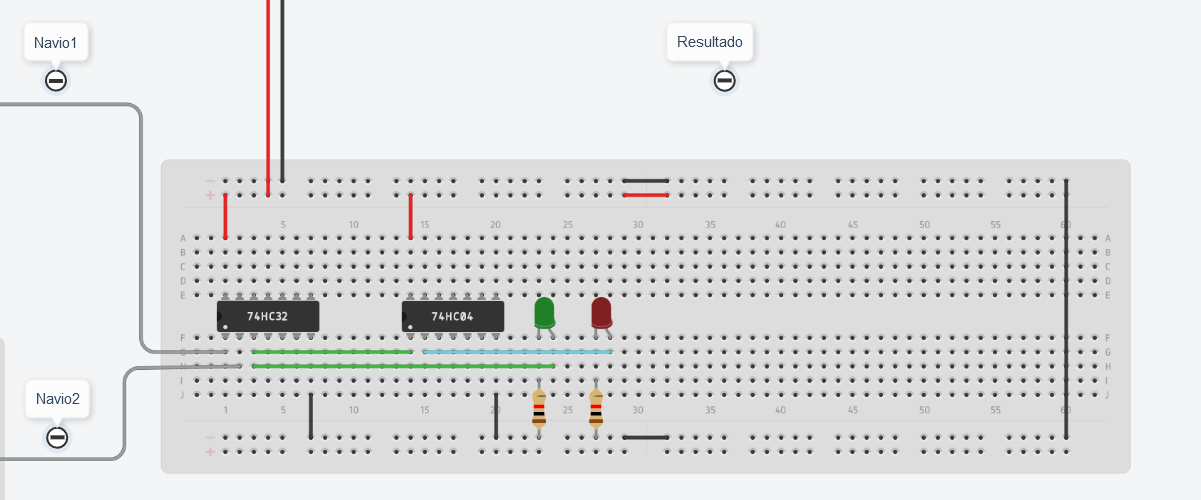
Faltando apenas agora transcrever tal circuito para protoboards “físicas” pelo tinkercad, de começo, é necessário criar as entradas e saídas de cada dos bits do alvo, nomeados no projeto como S1, S2, S3 e S4 ², um ligado ao outro após testes pela tabela verdade, tendo o seguinte resultado:

Figura\_10:

Seguindo a lógica do circuito feito através do logisim, esses resultados do alvo são enviados aos navios, que recebem dentro da protoboards deles mesmos as coordenadas, conforme imagem:

Figura\_11:

Por fim, o circuito do resultado recebe o resultado das protoboards navio acionando as LEDs, segue:

Figura\_12:

Parte física do bit S1 feita em laboratório e aprovada após executados testes da :

Conclusão

* Arquivo do projeto Logisim

\*¹ = Arquivo presentes no caminho “**\jonathan\_gotz-clara\_barbosa\Logisim**” para visualização, incluindo as tabelas verdades e simplificações, separadas por bits do “Alvo” e do circuito completo em si;

* Arquivo do projeto Tinkercad

\*² = Infos de acesso presente no caminho: “**\jonathan\_gotz-clara\_barbosa\Tinkercad**”;

\*³ = Print com mais detalhes presente no caminho: “**\jonathan\_gotz-clara\_barbosa\Logisim\Battle**”;

= Informações completas presentes no caminho: “**\jonathan\_gotz-clara\_barbosa\Montagem de um bit**”

* Link para um vídeo mostrando as funcionalidades e a execução correta do jogo

= Presente no link: https://youtu.be/KmXemo2-j8c

* Simplificações

= Presente no caminho: “\jonathan\_gotz-clara\_barbosa\Logisim”