



**UNIVERSIDADE
FEDERAL DO CARIRI**

Ciência da Computação

**Jonathas Levi Pascoal Palmeira
Islânio Gomes Feitosa**

Projeto de Circuitos Digitais 2024.1

**Juazeiro do Norte - CE
2024**

SUMÁRIO

1. INTRODUÇÃO.....	3
2. DESENVOLVIMENTO	3
2.1. Movimentos Dentro do Tabuleiro.....	4
2.2. Sistema de Escolha dos Displays.....	3
2.3. Sistema de Pontuação.....	7
3. CONCLUSÃO	10

1. INTRODUÇÃO

O Projeto proposto é a construção de um jogo da memória para dois jogadores utilizando circuitos sequenciais no software simulador Logisim. Neste software o usuário 01 poderá escolher duas posições dentro do jogo, caso essa escolha tenha como resultado dois números iguais, o software pontuará e passará a vez para o usuário 02. Se como resultado o usuário 01 tiver dois números diferentes, não marcará pontos e a vez passara para o usuário 02.

Serão utilizados 16 LEDs de display hexadecimal formando um tabuleiro 4x4. Cada par de displays terá um número distribuído aleatoriamente no tabuleiro (figura 01), formando 8 pares de displays com o mesmo número. No início do jogo os displays ficarão ocultos e à medida que os jogadores fizerem suas escolhas os números serão revelados. Se acertados os displays permanecerão visíveis durante todo o jogo se não, voltarão ao estado de oculto e caberá aos jogadores lembrarem suas posições.

A dinâmica do jogo é ver qual usuário consegue acertar mais pares de números.

2. DESENVOLVIMENTO

Iniciamos a etapa de desenvolvimento criando o tabuleiro. Para isso foi utilizado 16 displays de display hexadecimal posicionados lado a lado formando um tabuleiro (figura 01). Para facilitar a criação e organização do tabuleiro foi utilizado tuneis dentro do logisim.

Cada display deve ter seu túnel referenciado para que seja possível guardar a sua posição. Por exemplo o primeiro display tem como posição P00 assim, seu túnel deve ter como rótulo o texto P00, o segundo display tem como posição e texto de rótulo P01 e assim sucessivamente. Para inserir valores nos displays dos tabuleiros foi utilizado os tuneis onde, cada par de display deve receber dados de 4bits de um número específico pré-determinado.

Para esconder os números que cada display possui foi utilizado multiplexadores de 4 bits. Dessa forma sempre que o multiplexador receber um sinal baixo "0" o display em específico ficará escondido e quando receber um sinal alto "1"

o display será exposto. Foram utilizados 16 multiplexadores, cada um interligado por tuneis ao seu display específico dentro do tabuleiro.

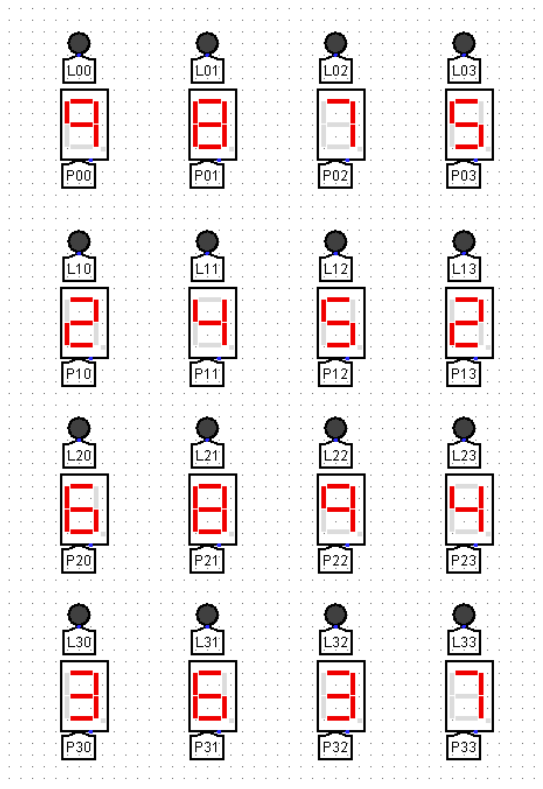


Figura 01 – Tabuleiro de LEDs de display hexadecimal.

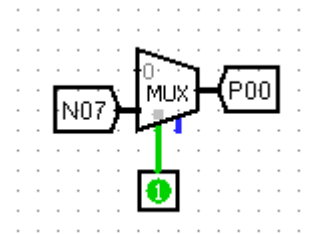


Figura 02 – Sistema com multiplexadores.

2.1. Movimentos Dentro do Tabuleiro.

Buscando sinalizar em que posição o usuário estar localizado dentro do tabuleiro, foram utilizados Leds referenciados posicionados em cima de cada display hexadecimal. Seguindo o mesmo raciocínio do posicionamento dos displays os leds possuem tuneis referenciados de acordo com sua posição. Por exemplo, o led da posição 00 tem seu túnel referenciado com o rótulo L00, o led da posição 01 tem seu túnel com o rótulo L01, e assim sucessivamente.

Para que os usuários possam se movimentar dentro do tabuleiro foram criados dois botões (figura 03), um indicando o movimento na horizontal (LINHA) e o outro o movimento vertical (COLUNA).

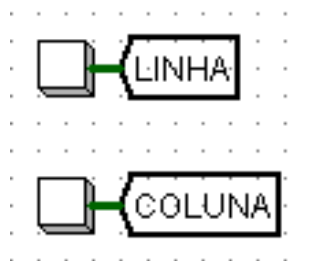


Figura 03 – Botões de deslocamento no tabuleiro.

Para que o movimento dentro do tabuleiro seja possível, foi utilizado dois contadores em conjunto com demultiplexadores (figura 04) formando um circuito posicional. A lógica segue o seguinte raciocínio: à medida que o usuário pressiona o botão “LINHA” o contador ligado ao túnel do botão soma um, partindo do zero até o número 3. Quando o contador chega ao número 3, ele retorna para 0, reiniciando o ciclo. Por estar ligado a um demultiplexador 1x4 quando o contador for 0, a primeira porta do demultiplexador ficará com sinal alto “1”, quando o contador for 2, a segunda porta do demultiplexador ficara com sinal alto, quando for 2 a terceira porta recebe sinal alto e quando for 3 a quarta porta recebe sinal alto. A mesma lógica é utilizada para o botão COLUNA.

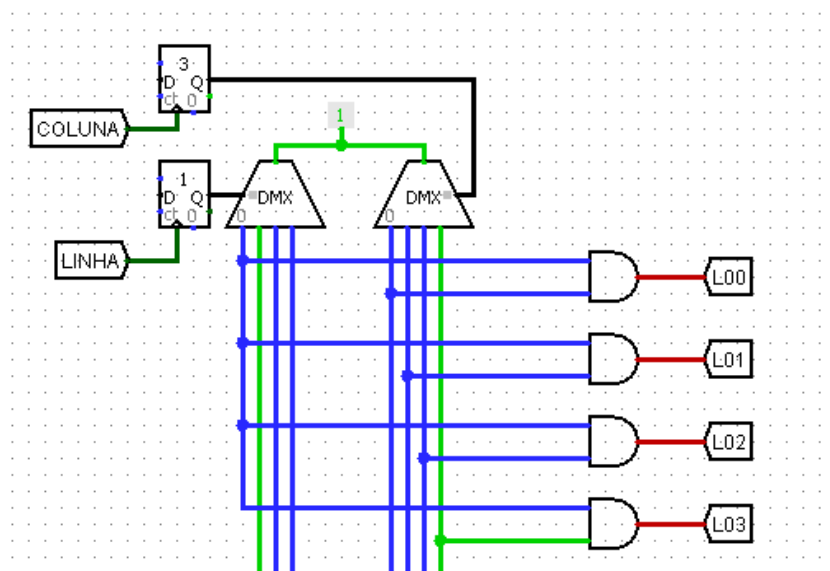


Figura 04 – Circuito posicional.

Percebe-se que com a ajuda dos demultiplexadores podemos posicionar os usuários dentro do tabuleiro seguindo os conceitos de plano cartesiano. Utilizando portas AND, é possível posicionar os usuários dentro de um plano cartesiano 4x4 em que cada coordenada é o resultado da saída dos demultiplexadores.

Por exemplo, quando o demultiplexador referente a LINHA estiver com a porta 1 com sinal alto “1” e o demultiplexador referente a COLUNA, estiver com a porta 3 com sinal alto. A posição do jogador será Linha 1 Coluna 3 dentro do plano cartesiano 4x4 do tabuleiro.

Para que os jogadores possam ver em que posição estão, os leds posicionados em cima dos displays estão referenciados de acordo com a sua posição vinda do circuito posicional.

2.2. Sistema de Escolha dos Displays.

Sabemos que para que o jogo da memória funcione todos os displays devem ficar ocultos e no momento que um dos jogadores escolher um determinado display, sua numeração deve ser revelada, caso a segunda escolha seja a mesma numeração, os dois devem permanecer revelados durante toda a partida. Caso a segunda escolha seja diferente da primeira os dois devem voltar ao estado de oculto. Para implementar esse sistema foi utilizado o conceito de máquinas de estado.

A máquina de estados funciona como um contador dessa forma, sempre que um número é somado o sistema muda de estado. Para esse caso serão necessários quatro estados:

Estado 00 – Momento em que todos os displays estão ocultos.

Estado 01 – Momento em que o primeiro display é escolhido, revelando a sua numeração.

Estado 02 – Momento em que o segundo display é escolhido, revelando a sua numeração.

Estado 03 – Momento de transição para o estado 00.

Para a montagem do circuito de mudança de estados foi utilizado um botão confirma, que sempre que pressionado tem a função de mudar o estado do sistema. Esse botão manda um sinal alto “1” para o contador que por sua vez soma um número a contagem. Com um conjunto de portas lógicas AND é possível que cada saída do contador seja um estado do sistema. Assim, quando o contador for 00 o sistema estará no estado 00, quando for 01 o sistema estará no estado 01, 02 estará no estado 02 e 03 estará no estado 03. É importante frisar que sempre que alcançado o estado

03 o circuito voltará ao estado 00, reiniciando a contagem à medida que pressionado o botão confirma. Foi adicionado um botão RESET, caso os jogadores desejem reinicializar o jogo antes do fim da partida. Este botão RESET está ligado a um porta OR junto com um túnel do estado 03, assim em qualquer das ocasiões o contador e reinicializado.

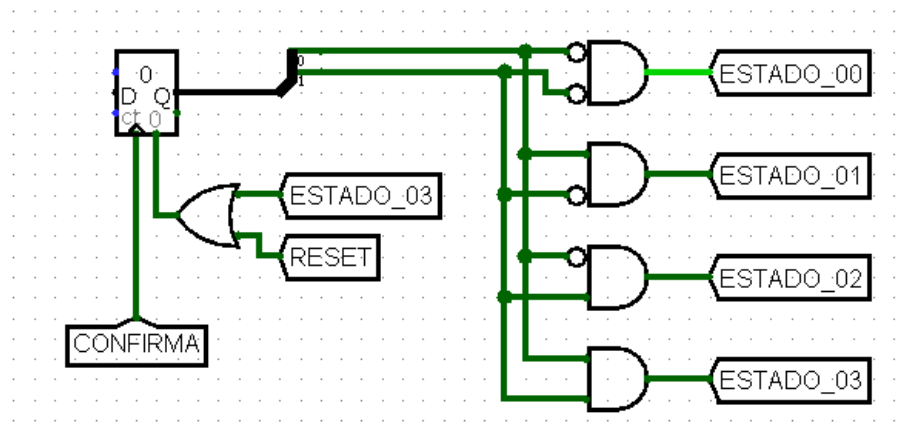


Figura 05 – Circuito de mudança de estados.

Para guardar a posição do display no momento da primeira e segunda escolha foi utilizado um circuito com dois contadores onde, um contador terá como saída a posição da linha e o outro a posição da coluna (figura 06).

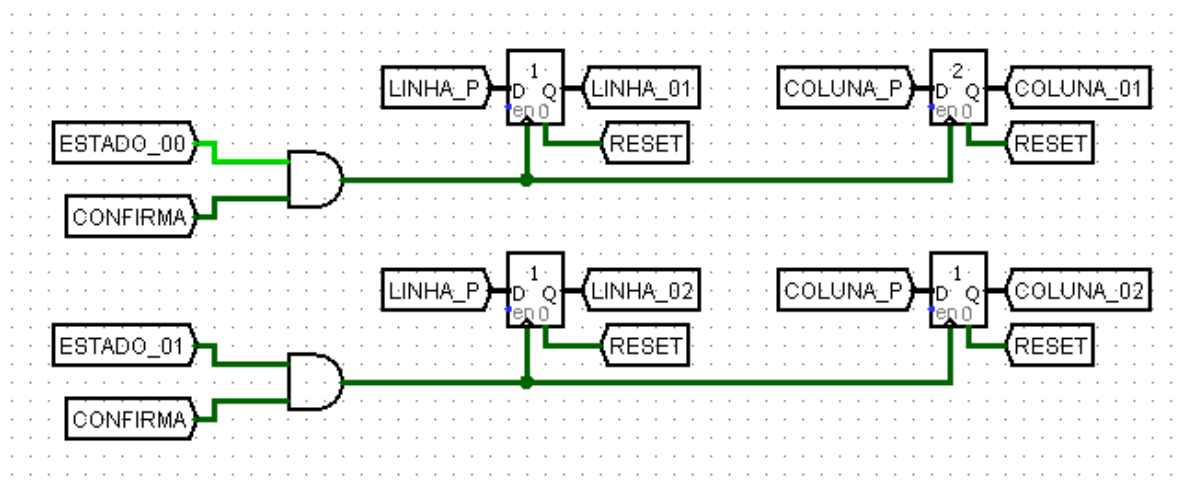


Figura 06 – Circuito para guardar a posição da primeira e segunda escolha.

No instante em que o usuário pressionar o botão confirma o circuito é ativado, se o jogo estiver no estado 00 a primeira escolha será guardada (LINHA_01 e COLUNA_01), caso o jogo esteja no estado 01 a segunda escolha será guardada

(LINHA_02 e COLUNA_02). Foi adicionado um túnel do botão reset para caso os jogadores necessitem que o jogo seja reiniciado.

Durante a jogatina o display que foi revelado na primeira escolha do jogador deve permanecer visível até que seja confirmada a segunda escolha, caso essa segunda escolha seja o mesmo número da primeira, os dois displays em questão devem permanecer visíveis até que o jogo finalize, caso não seja o mesmo número, os dois displays devem voltar ao estado de oculto.

Para que essa lógica funcione devemos implementar dois circuitos, um que analise se as duas escolhas são iguais ou não e outro que mantenha a primeira e a segunda escolha visíveis até o estado em que seja feita a verificação de sua paridade, permanecendo visíveis se forem iguais e voltando ao seu estado oculto se forem diferentes.

Para verificar se os displays escolhidos são iguais foi desenvolvido um circuito formado por multiplexadores 16x1 que recebem 16 entradas correspondentes aos números existentes em cada display dentro do tabuleiro. Como seletor é utilizado a posição guardada na primeira escolha assim, quando o usuário realizar sua primeira escolha o multiplexador passara como resultado o número correspondente a posição escolhida. O mesmo raciocínio é utilizado para a segunda escolha.

A saída dos multiplexadores seguirá para um comparador que fará a análise se os números selecionados são iguais ou não. Para que a lógica funcione corretamente essa análise deve ser feita dentro do estado 03, dessa forma, antes do resultado ser levado para o circuito responsável por manter o display visível, deve-se verificar se de fato a análise está sendo feita no estado 03. Para isso foi adicionado uma porta AND que terá saída alta “1” sempre que estiver no estado 03 e os números selecionados forem iguais.

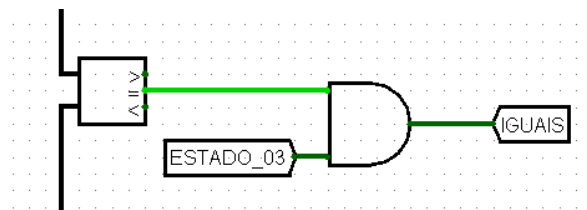


Figura 07 – Análise final da comparação.

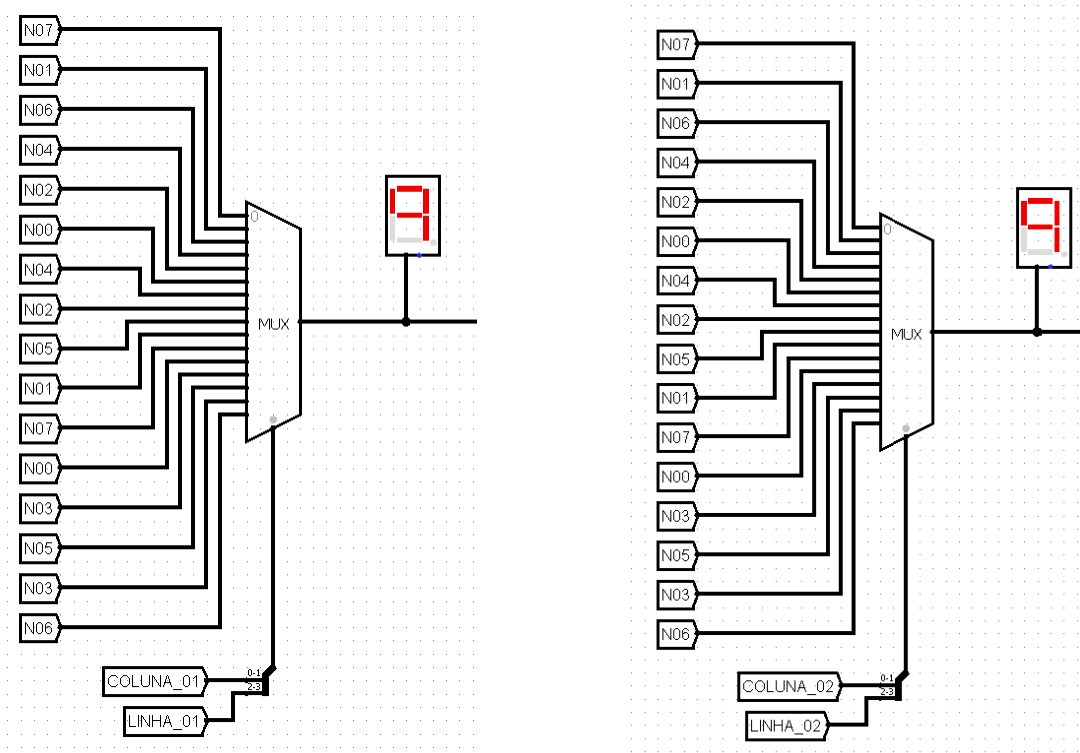


Figura 08 – Multiplexadores do circuito comparador.

Com o circuito responsável por realizar a comparação no estado 03 dos displays selecionados podemos implementar o circuito que os manterá visíveis no decorrer do jogo.

A exibição de um display ocorre em três situações específicas:

- Se o jogo estiver em um estado diferente do estado 00 e a posição LINHA_01 e COLUNA_01 estiverem com a sua posição, indicando que foi a primeira escolha.
- Se o jogo estiver em um estado 02 e a posição LINHA_02 e COLUNA_02 estiverem com a sua posição, indicando que foi a segunda escolha.
- O display será exibido se já tiver sido acertado. Para isso, precisaremos de um contador que receberá como entrada “1” caso o sistema esteja no estado 3, o display tenha sido escolhido na primeira ou na segunda opção e ambos os números escolhidos forem iguais.

Essas três condições ativam os multiplexadores que controlam a exibição do valor no display, conforme explicado na seção anterior sobre a criação do tabuleiro de displays.

No decorrer dos testes percebeu-se que quando se confirmava novamente uma alternativa já acertada, os displays voltavam a ficar oculta. Para evitar esse erro

utilizamos um artifício, um túnel que liga a saída do contador ate a porta lógica AND negada, possibilitando que mesmo que a opção acertada seja escolhida novamente a mesma não fique oculta.

Para facilitar a reinicialização do jogo foram incluídos tuneis do botão reset no contador.

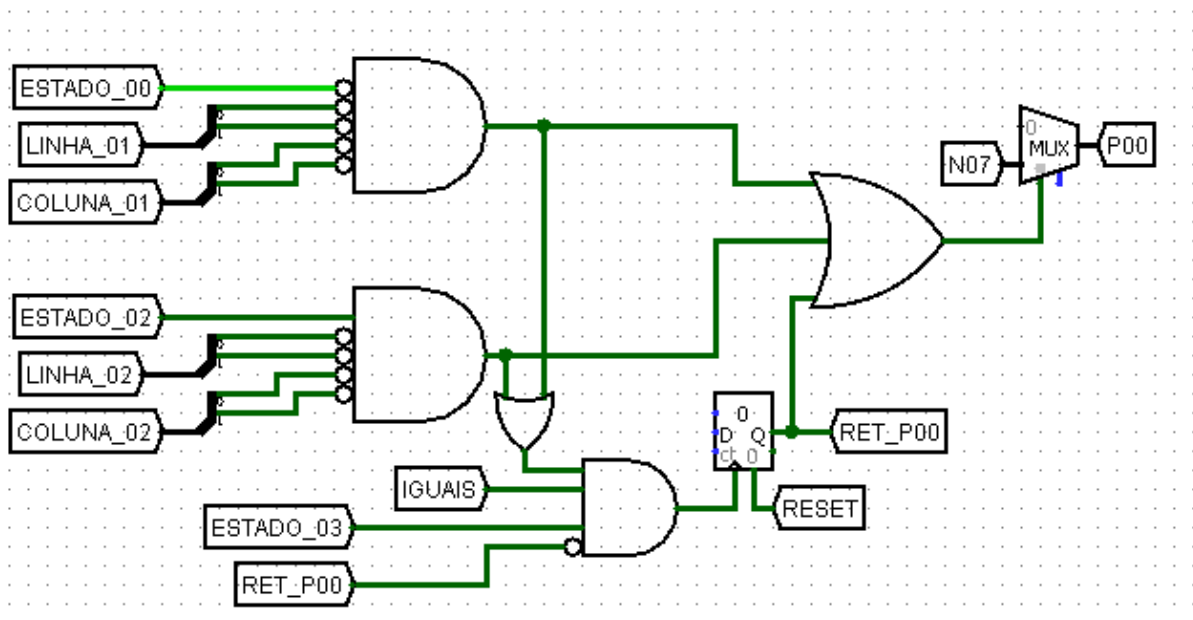


Figura 09 – Exemplo do sistema responsável por manter os displays visíveis (Posição 00).

2.3. Sistema de Pontuação.

Nesta etapa do projeto precisamos desenvolver um sistema de pontuação que registre o número de acertos que cada usuário realizou durante o jogo. Primeiramente montamos o quadro com dois displays hexadecimais responsáveis pela pontuação de cada jogador. Um LED em cima dos displays sinalizará a vez de cada jogador (figura 10).

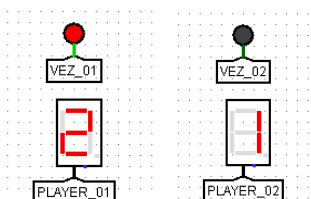


Figura 10 – Painel de pontuação dos jogadores.

Para que esse sistema de pontuação possa funcionar corretamente ele deve ser acionado quando o jogo estiver no estado 03. Assim um contador recebera como

A mesma lógica se aplica para o registro da pontuação de cada jogador. Sempre que a vez for do player_01 e os números forem iguais, um contador acionará a sua pontuação. A mesma lógica se aplica para o player_02.

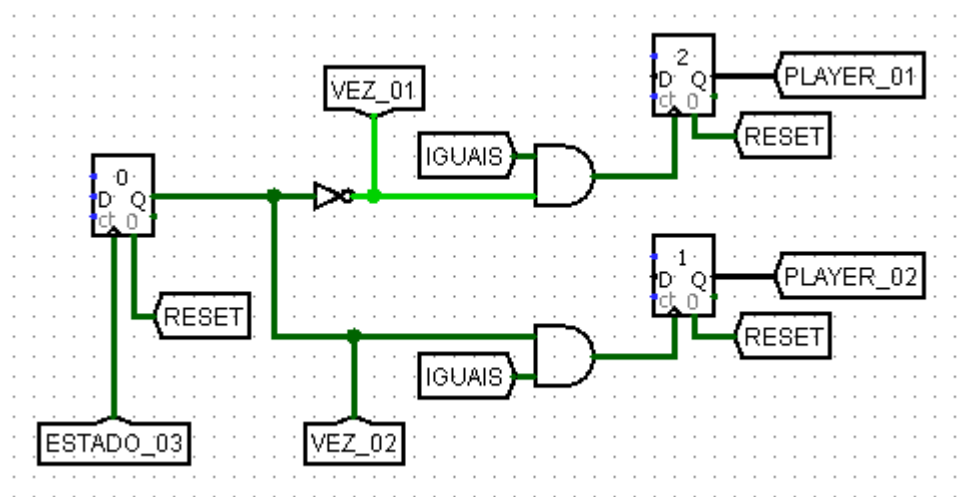


Figura 11 – Sistema de pontuação.

Durante os testes observou-se que se um dos jogadores quisesse utilizar um número já acertado o sistema de pontuação contabilizaria como um novo acerto. Para evitar esse erro foi implementado uma verificação antes do sistema pontuar a jogada. Essa verificação se divide em duas etapas.

A primeira é implementar uma verificação de acerto nos circuitos que mantem os displays acertados visíveis, como mostrado na figura 12. Um túnel com o título ACERTO mais a posição do número, por exemplo o número 00, recebe como entrada a saída de uma porta AND que é ativada sempre que o número é escolhido em algumas das duas opções dos jogadores e durante essa escolha houve um acerto.

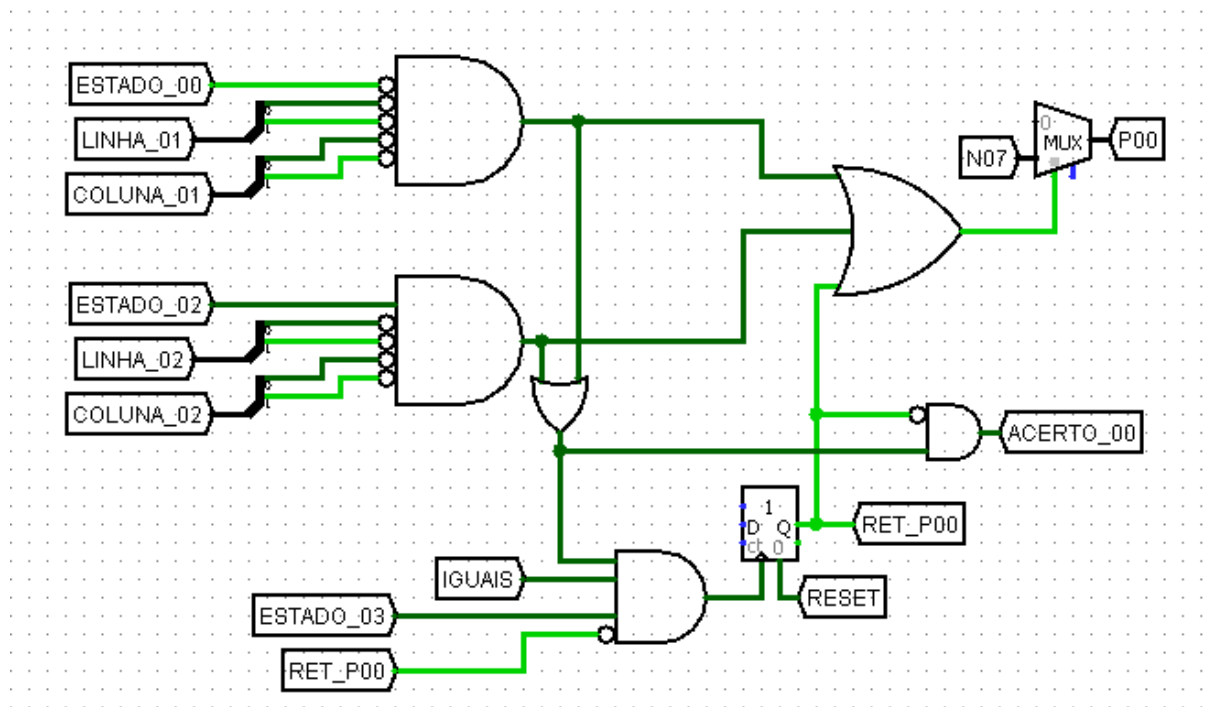


Figura 12 – Melhoramento do sistema de visualização dos displays.

A segunda etapa é realizar a verificação se as escolhas do jogador não foram acertadas em uma jogada anterior. Para isso, foi utilizado um circuito com um conjunto de portas AND que recebe os acertos de cada par de números dentro do tabuleiro.

Neste caso, sempre que um número estiver visível é porque foi acertado e logicamente se foi acertado é porque os dois pares existentes no tabuleiro estão visíveis.

Após essa análise inicial o circuito recebe um sinal do comparador (CONDICIONAL) mostrando que foi feita uma escolha de dois displays iguais. Se essa escolha não estiver ativa o sistema libera a pontuação (OP_VALIDA), caso essa escolha esteja ativa (visível) o sistema não pontuará e como punição ao jogador passará a vez.

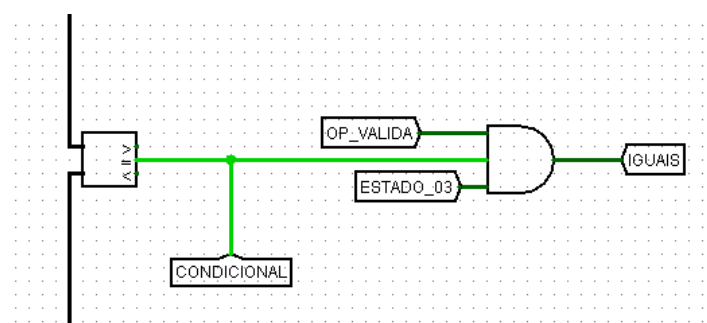


Figura 13 – Implementação da validação para o sistema de pontuação.

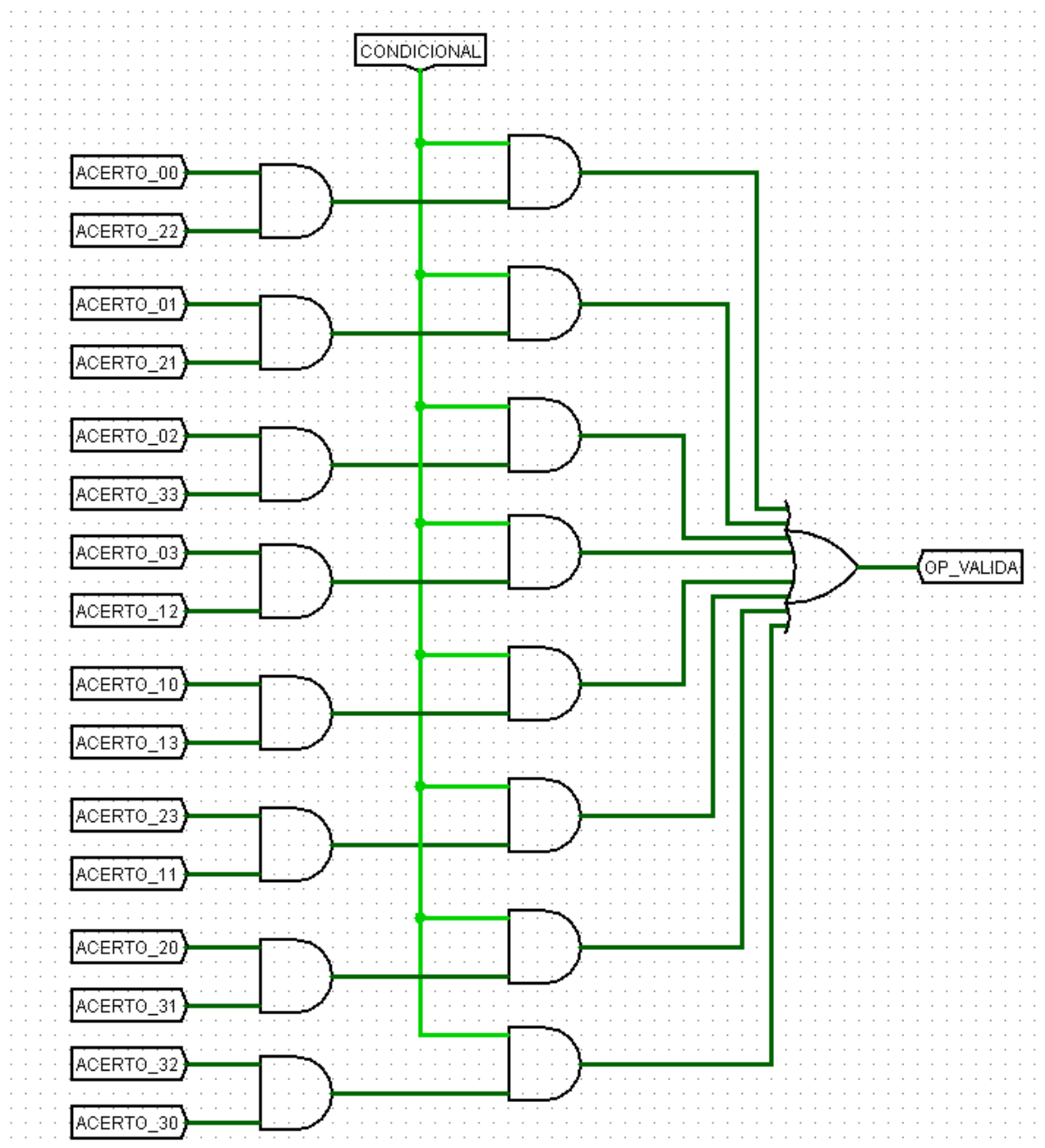


Figura 13 – Sistema que verifica se a opção escolhida não foi acertada anteriormente

3. CONCLUSÃO

As dificuldades enfrentadas durante a construção dos circuitos foram extremamente importantes para o entendimento da disciplina Circuitos Digitais, mostrando na prática todo o conhecimento passado em sala de aula. Projetar e

entender como funciona os sistemas sequenciais dentro da lógica de circuitos digitais, nós incentivou a buscar soluções e entender que construir coisas não é apenas seguir manuais prontos, mas buscar criar soluções práticas.

Após a finalização dos circuitos foram feitos vários testes comprovando a assertividade do circuito principal e o bom funcionamento do jogo de memória. Lembrando que durante a construção de todos os circuitos foram feitos testes preliminares.