3.3 - Sistema eletrónico para o controlo dos motores (pcbs e ligações)

De forma a ser possível controlar o estado de funcionamento do robô, ou seja, fazê-lo seguir a linha, parar ou rodar 180 º, desenvolveu-se uma máquina de estados sequencial síncrona, mais especificamente uma máquina de Moore.

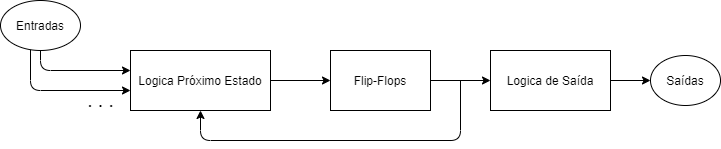


Figura 1 - Máquina de Moore

Numa máquina de estados de Moore, o objetivo é determinar a “Lógica do Próximo Estado” e a “Lógica de Saída” do circuito. Como se pode ver pela figura, “Lógica de saída” depende apenas do estado atual dos Flip-Flops (FF). Já as entradas dos FF (a “Lógica do Próximo Estado”) dependem do estado atual e das entradas do circuito.

Como referido, o objetivo do robô é seguir uma linha, devendo parar sobre uma linha horizontal colocada no final do percurso (figura 2) e, nesse momento, o robô deve dar uma volta de 180 º sobre o seu eixo, voltando a seguir a linha em direção ao ponto de partida, após o acionamento de um botão.

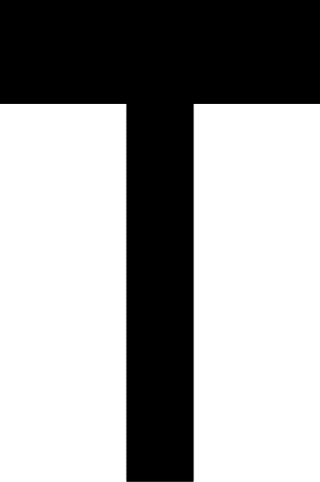


Figura 2 - Linha Preta Usada para os Testes

De modo a detetar o final de percurso, usaram-se os dois sensores das pontas do array de sensores (S1 e S8, rodeados a amarelo na figura 3). Para o robô rodar, usou-se também o sensor do meio (S5, rodeado a verde na figura 3) de forma a interromper a rotação quando este encontrar a linha preta.



Figura 3 - Botão de Pressão

Nesse momento, o robô poderá ser acionado através de um botão de pressão (B).

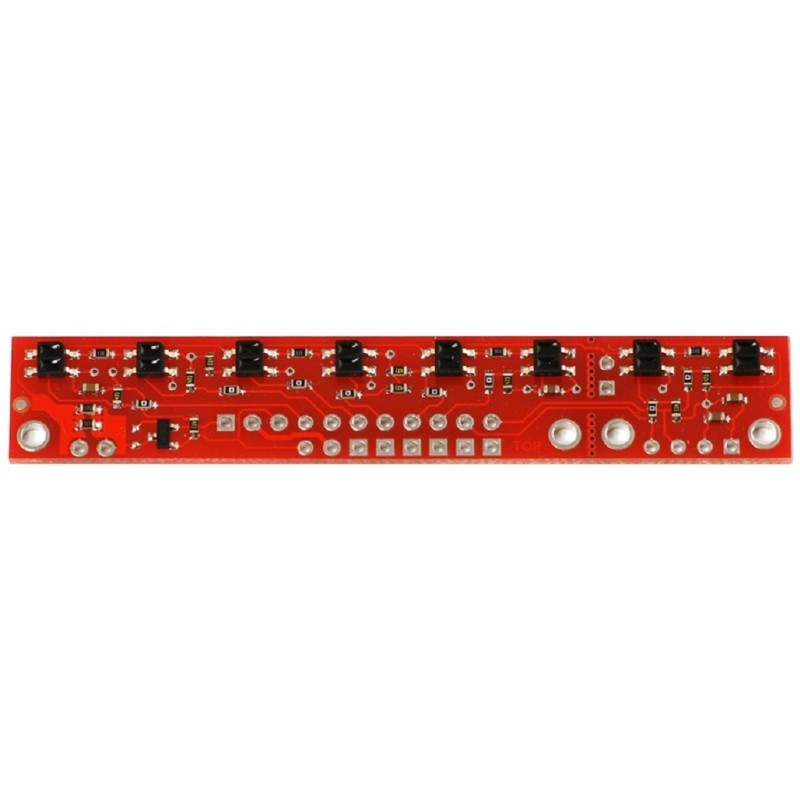


Figura 4 - Array de Sensores QTR-8A (Sensores de Linha - S1 a S8)

O controlo do estado de funcionamento do robô é feito através do controlo dos motores. Os motores são controlados por meio de um *driver*, podendo estar parados ou a rodar no sentido horário ou anti-horário.

Para isso, na máquina de estados tem-se duas saídas para cada motor, sendo que uma controla o movimento no sentido horário e outra no sentido anti-horário.

Assim sendo, temos quatro saídas, que quando estão ativas (nível lógico alto) despoletam as seguintes respostas:

* FWR – motor da direita roda no sentido horário;
* FWL – motor da esquerda roda no sentido horário;
* BWR – motor da direita roda no sentido anti-horário;
* BWL – motor da esquerda roda no sentido anti-horário.

Na figura seguinte, mostra-se o diagrama da máquina de estados, usada para controlar o modo de funcionamento do robô, em que FW representa os valores de FWR e FWL e BW representa os valores de BWR e BWL.

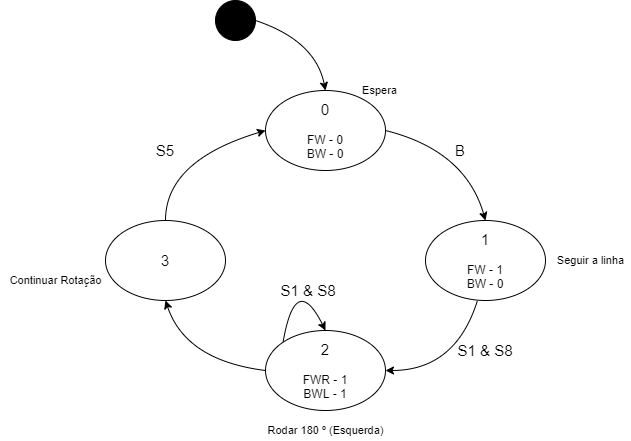


Figura 5 - Diagrama da Máquina de Estados

Inicialmente, no estado 0 o robô espera, parado (FW = 0 e BW = 0), pelo acionamento do botão para iniciar a marcha. Após o acionamento do botão, no estado 1, o robô inicia a trajetória, seguindo a linha com os dois motores no sentido horário (FW = 1 e BW = 0). Quando o *array* de sensores estiver posicionado sobre a linha preta, o robô estará no fim da sua trajetória, pelo que os sensores S1 e S8 estarão a nível lógico alto, transitando de estado. No estado 2, o robô inicia a rotação para a esquerda (FWR = 1 e BWL = 1) e não transita de estado enquanto os sensores S1 e S8 estiverem a nível lógico alto. Quando o *array* de sensores sair da linha preta, S1 e S8 estarão a nível lógico baixo, avançando para o estado 3 em que o robô continua a sua rotação (FWR = 1 e BWL = 1) até que o sensor do meio (S5) esteja sobre a linha. Assim posicionado, o robô volta ao estado inicial.



Tabela 1 - Tabela Completa para Determinação da Lógica Completa da Máquina de Estados

Com a ajuda do diagrama de estados, determinou-se toda a lógica necessária à implementação da máquina de estados. Para esse efeito foi necessário seguir um conjunto de passos, com os quais foi construída a tabela 1.

Adicionaram-se duas colunas que indicam os bits do estado atual (B1 e B0). Perfiladas, foram colocadas as entradas da máquina de estados (S1, S5, S8 e B). Como existem quatro entradas, tem-se dezasseis (24) possibilidades combinacionais para cada estado, o que resulta em sessenta e quatro possibilidades no total (24+2 = 26 = 64).

Tabela 2 - Atribuição de Estados



A seguir, “E” representa o estado atual e “E\*” o estado seguinte. “B1\*” e “B0\*” representam os bits do estado seguinte. Estas colunas devem ser preenchidas de acordo com a tabela ao lado.

Sabendo que a cada dois estados, deve ser usado um flip-flop e que existem quatro estados, selecionaram-se dois *flip-flops* do tipo J-K, o qual apresenta a tabela de excitação ao lado. De acordo com esta tabela, foram preenchidas as colunas “J” e “K” para cada *flip-flop* (Ja e Ka; Jb e Kb). Através de mapas de Karnaugh, determinou-se a lógica do próximo estado.



Figura 6 - Lógica do Próximo Estado



Tabela 3 - Tabela de Excitação do Flip-Flop JK

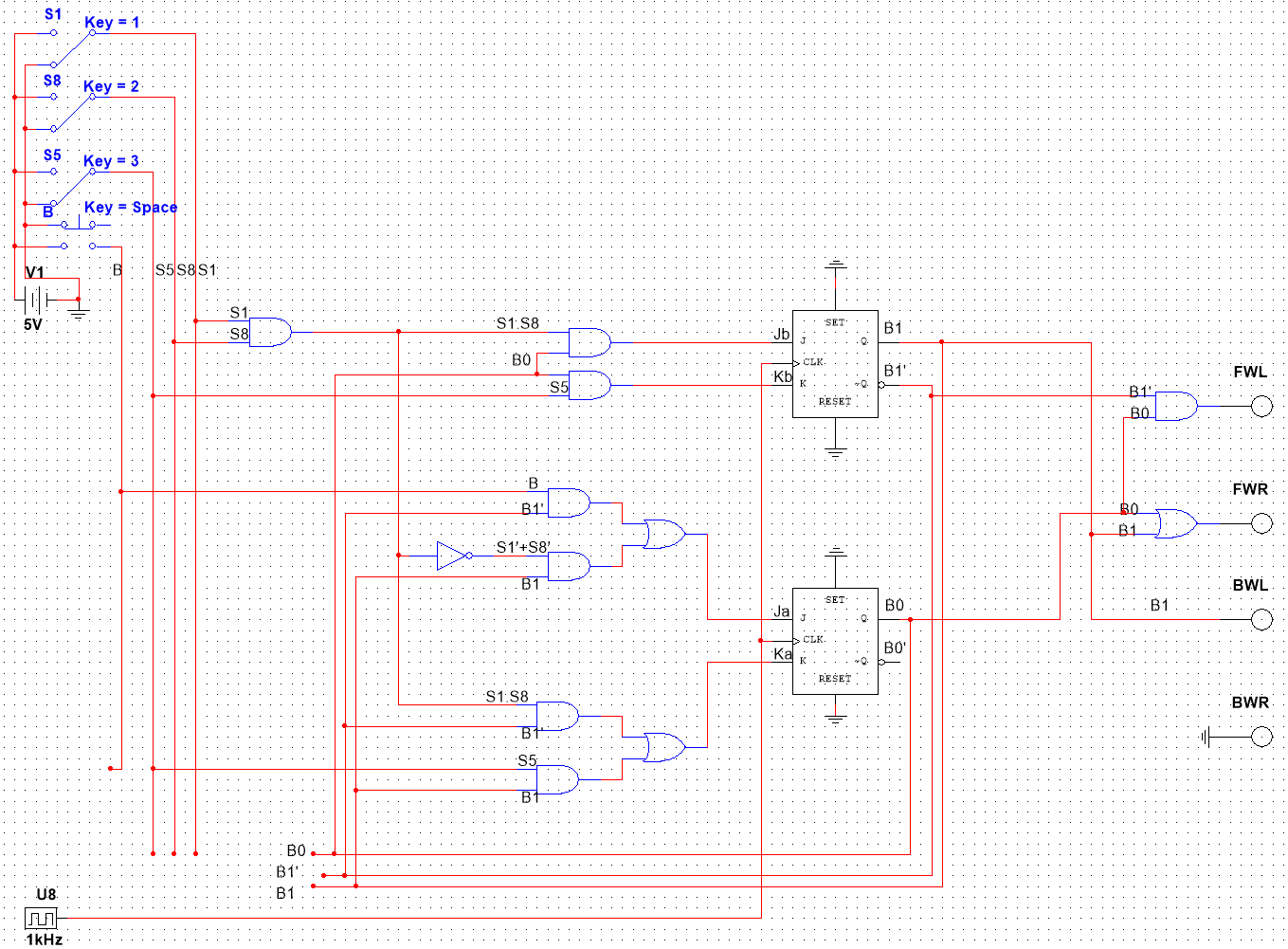
Adicionou-se as colunas das saídas (FWR, FWL, BWR e BWL), as quais foram preenchidas de acordo com o estado atual. Para cada saída criou-se a respetiva lógica de saída, através de mapas Karnaugh.



Figura 7 - Lógica de Saída

De acordo com a lógica determinada, desenhou-se o circuito que implementa a máquina de estados, a qual foi testada usando Multi-Sim, verificando-se a sua funcionalidade.

Figura 8 - Desenho da Máquina de Estados



Tal como mostra a figura 9, cada Flip-Flop J-K necessita de um sinal de *clock* (oscilador) como entrada. Um *clock* é um sinal em forma de onda quadrada com um *duty-cycle* de, aproximadamente, 50 %. Dá-se uma transição entre estados sempre que, a cada pulso de *clock*, uma das entradas (J ou K) estiver a nível lógico alto.

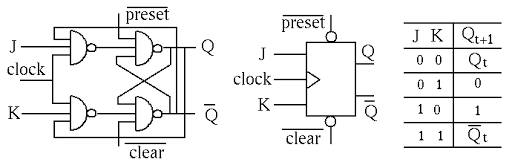


Figura 9 - Esquema do Flip-Flop J-K

Para implementar o oscilador, usou-se o integrado NE555 numa montagem como multivibrador astável:

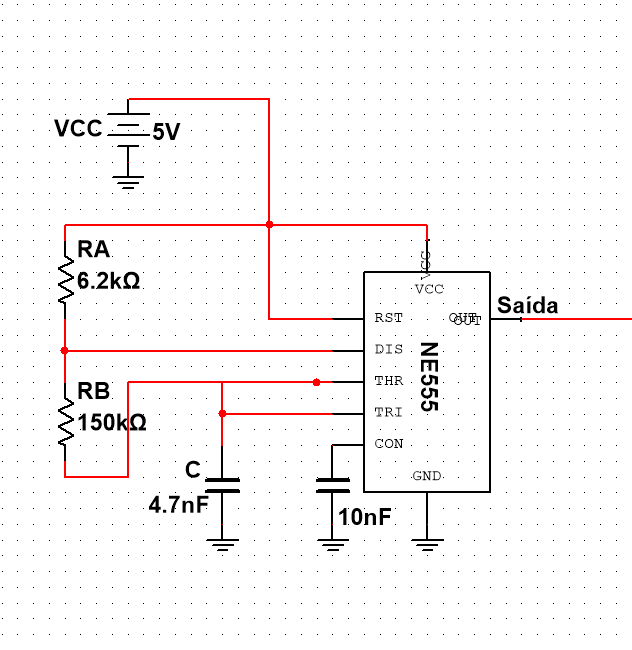


Figura 10 - Implementação do Oscilador

O condensador C1 carrega até 2\*VCC/3 através das resistências RA e RB (saída = 1). Quando C1 atinge 2\*VCC/3, o condensador descarrega por RB (saída=0). O condensador descarrega até VCC/3 e o ciclo repete-se.

Para e :

Valores standard de resistências (Série E24) e condensador (Série E12):

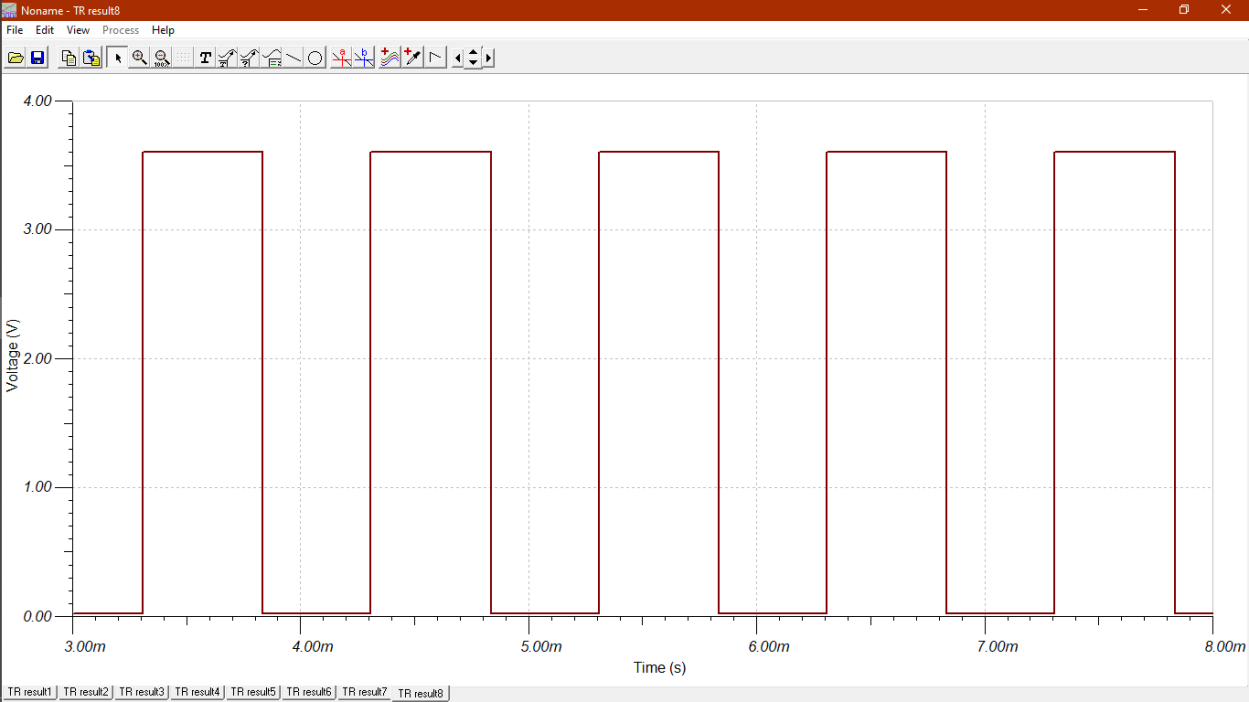


Figura 11 - Resposta do Oscilador de 1 kHz

As entradas “*Clear”* dos Flip-Flops J-K utilizados são ativas a nível lógico baixo, isto é, se ligarmos esta entrada ao *“ground”*, a saída “Q” fica a nível lógico baixo. Assim, se na máquina de estados as entradas “*Clear”* forem ligadas *ao* GND*,* B0 e B1 ficarão a nível lógico baixo, que corresponde ao primeiro estado*.*

Experimentalmente, verificou-se que, quando alimentados, os flip-flops mantêm as saídas do estado anterior. Para que o robô inicie no primeiro estado de funcionamento, utilizou-se um circuito para efetuar um *reset* aos flip-flops, tal como mostra a figura 10.

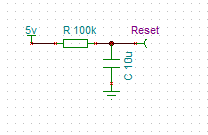
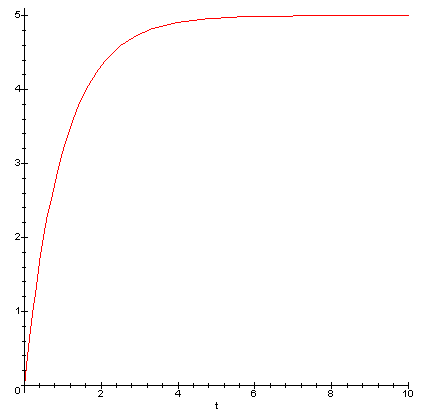


Figura 12 - Circuito RC de Reset e a sua resposta no tempo.



Sabendo que a saída deste circuito estará a nível lógico baixo enquanto o condensador não carregar, a entrada *reset* dos flip-flops estará também a nível lógico baixo, o que resulta na ação de *reset* destes. Assim, logo que se alimentar a máquina de estados, os flip-flops apresentarão nas suas saídas “Q” o nível lógico baixo que representa o primeiro estado de funcionamento do robô. Após o condensador carregar, a entrada *reset* estará a nível lógico alto durante o resto do tempo de funcionamento, o que não irá provocar uma ação de *reset* indesejada.

Foi usado um botão de pressão para que o robô inicie a marcha. No entanto, sabe-se que quando uma saída provém diretamente de um botão normal, esta é afetada por problemas de vibração mecânica (*bounce*) quando o contacto se fecha. De forma a garantir que a mudança de nível lógico se faça sem oscilações, torna-se necessário implementar um circuito de *debounce* como, por exemplo, o apresentado na figura 13.

O condensador em conjunto com a resistência funciona como filtro passa-baixo, que ajuda a reduzir o ruido provocado pelas vibrações mecânicas do botão. Assim, quando o botão não está pressionado, os terminais do condensador apresentarão 5 V e, pelo contrário, quando pressionado apresentarão 0 V. Uma vez que, num circuito sequencial, o nível lógico alto corresponde ao acionamento de uma entrada (botão, neste caso), será necessário utilizar uma lógica inversora na saída deste circuito. Para isso, a porta Schmitt-Trigger (foi usado o integrado 74HCT14) funciona como um circuito comparador inversor com histerese centrado em 0 V, e que aumenta a imunidade ao ruído e, como inverte, a saída do circuito estará a nível lógico alto quando o botão for pressionado.

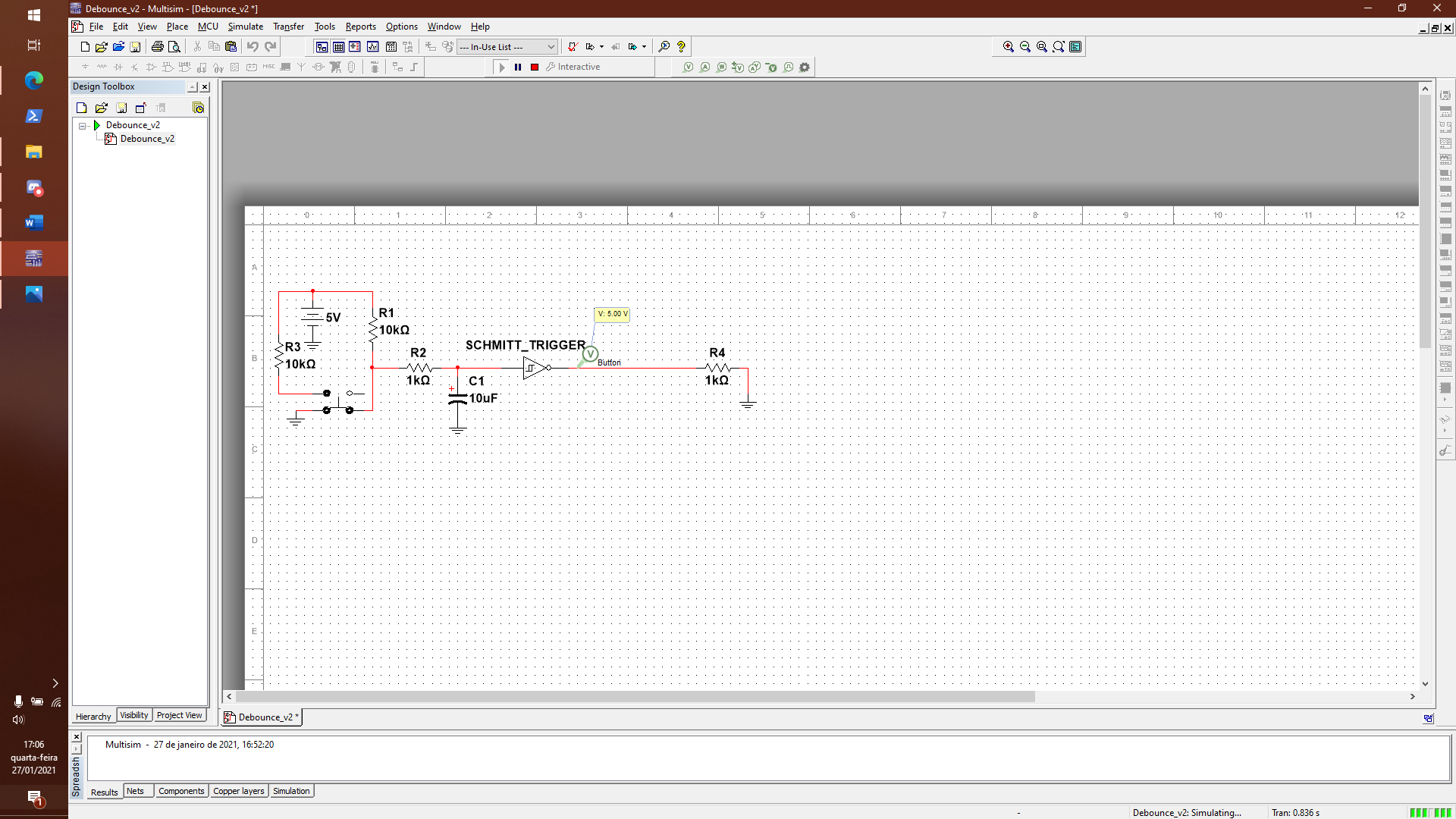


Figura 13 - Circuito de Debounce do Botão de Pressão (quando premido)

OUTRO PONTO EXPERIMENTAL

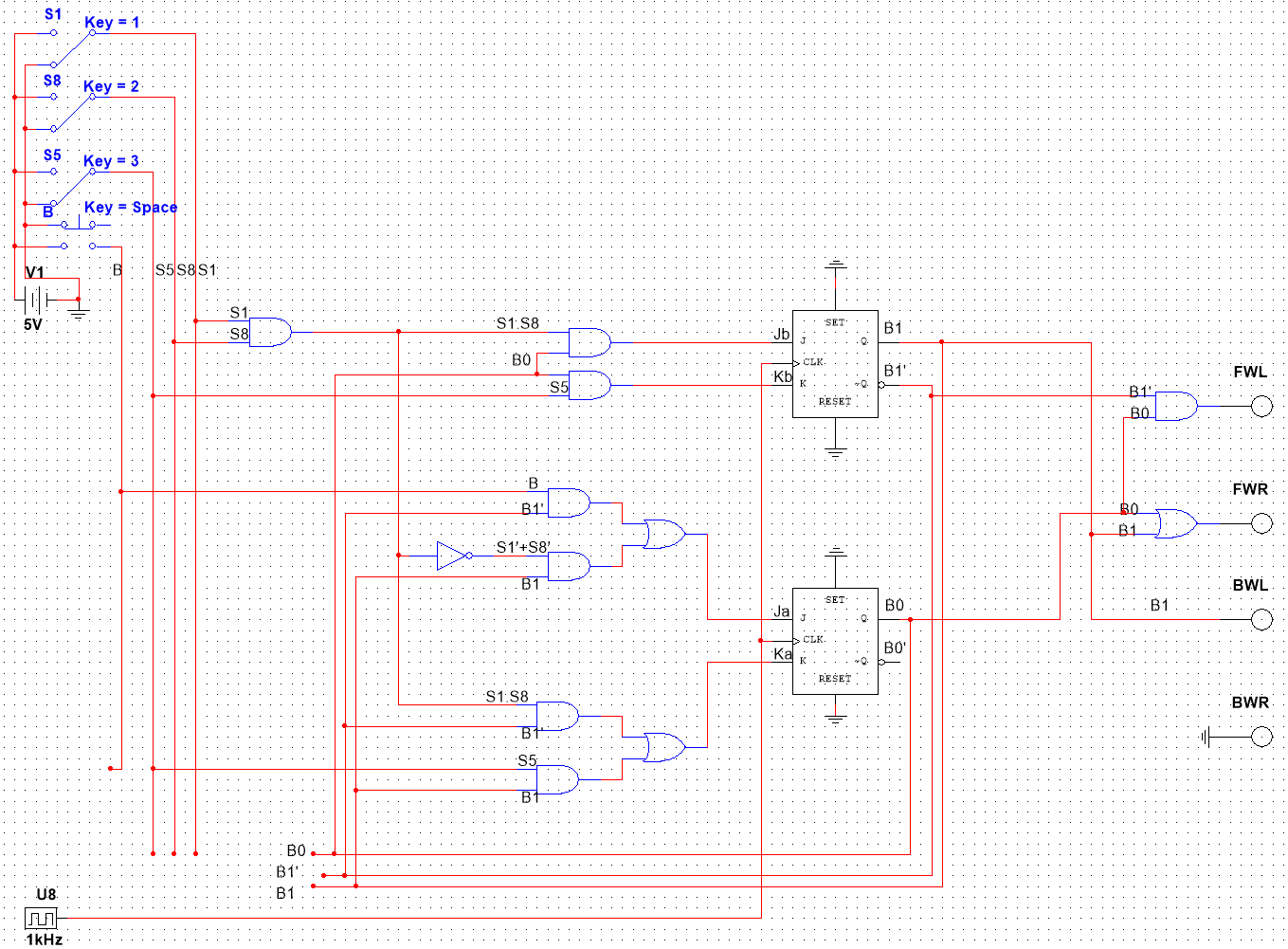


Figura 14 - Estado 0 (Parado)

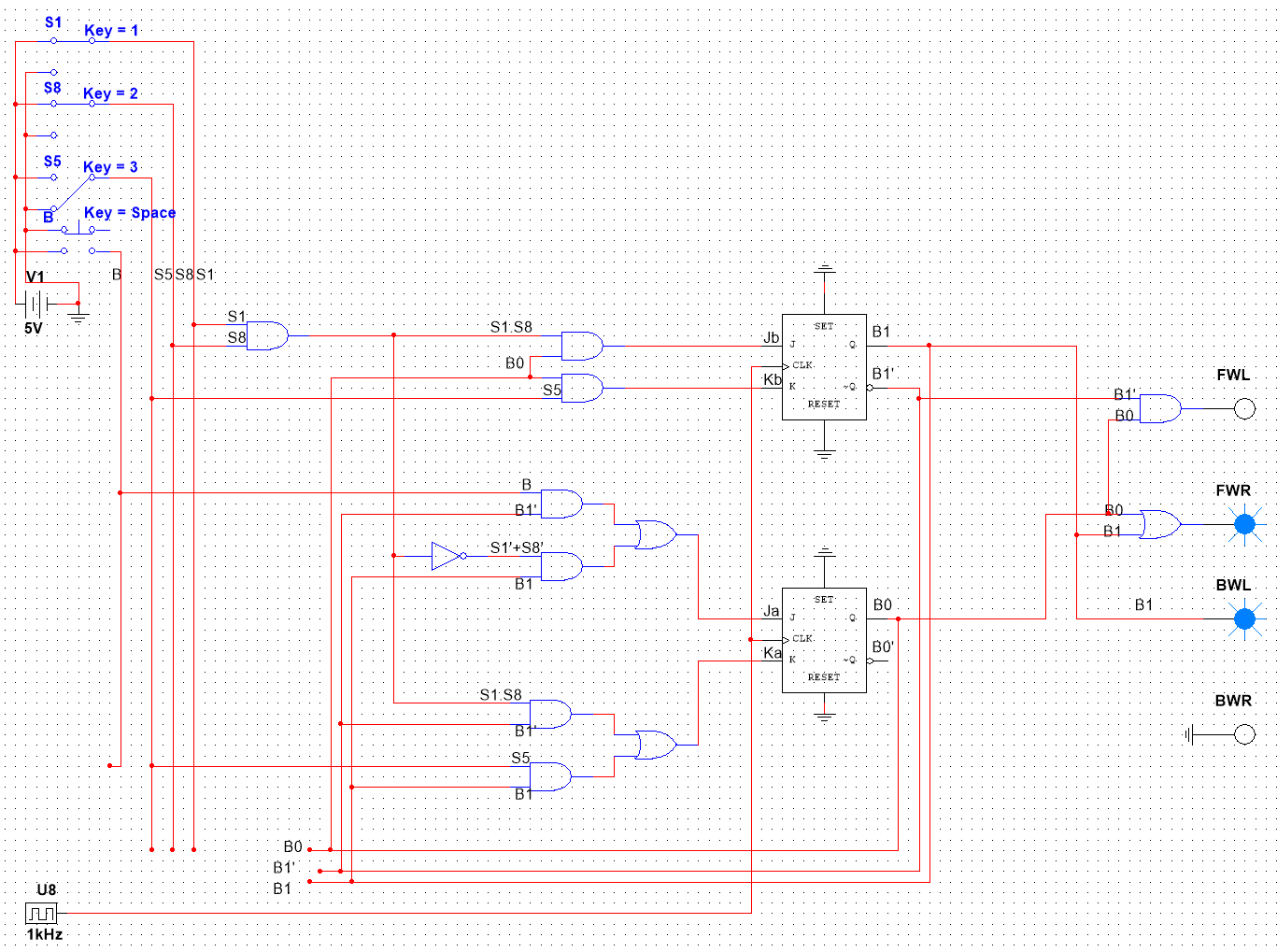


Figura 15 - Estado 2 (Dá a volta para a esquerda) – S1 e S8 a nível lógico alto

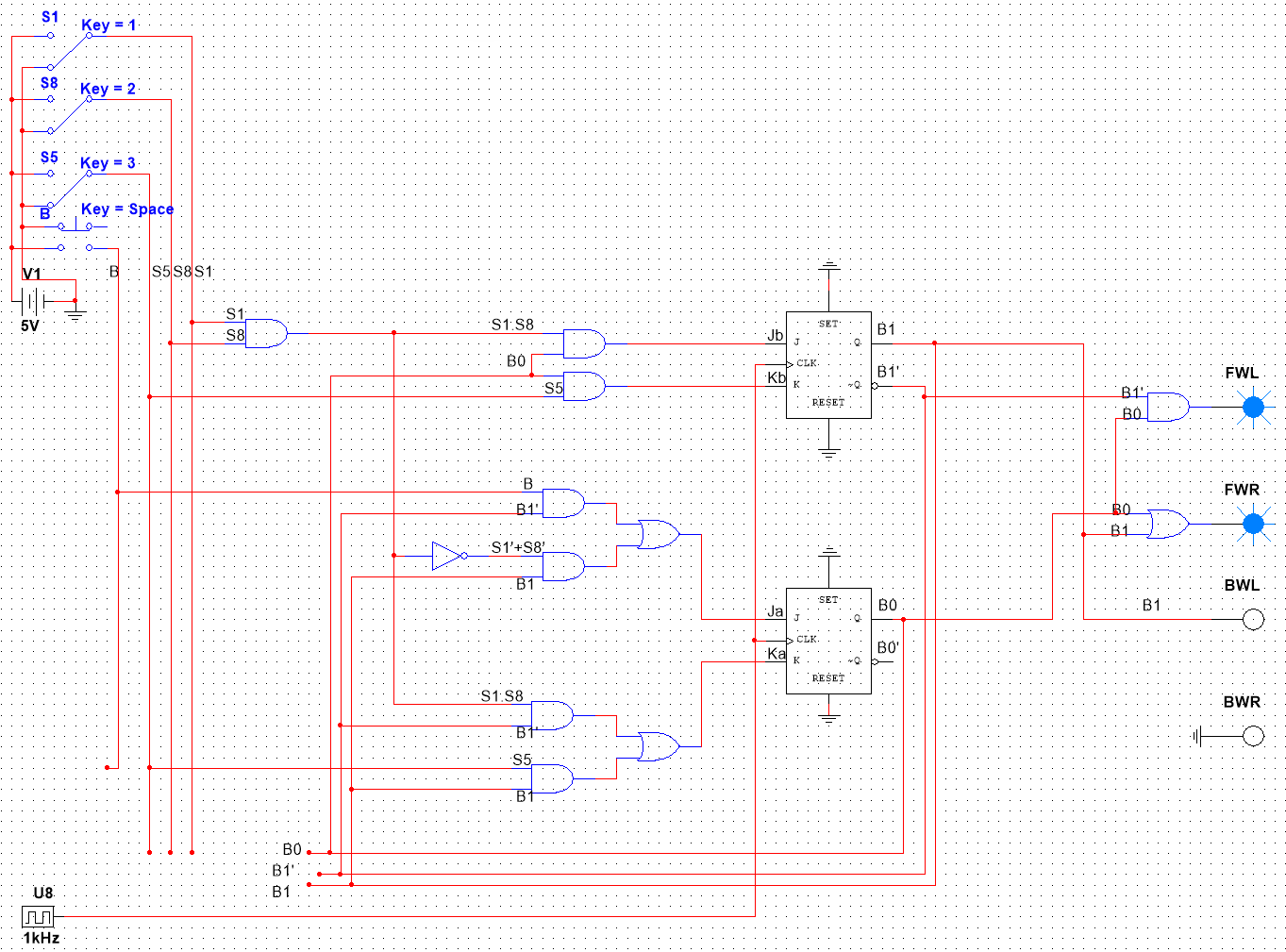


Figura 16 - Estado 1 (Anda para a frente) – Botão foi pressionado

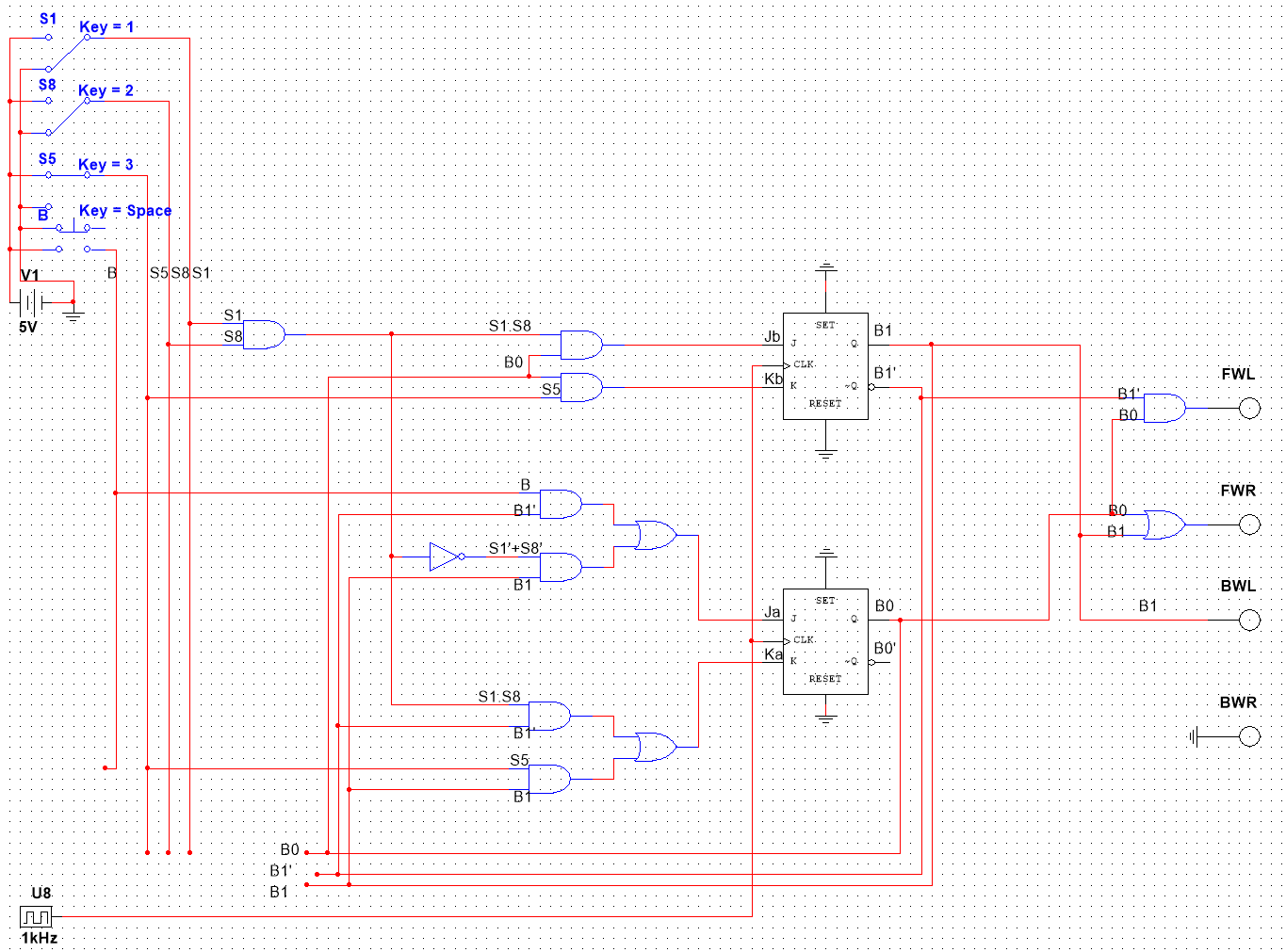


Figura 18 - Estado 0 (Parado novamente) - S5 = 1

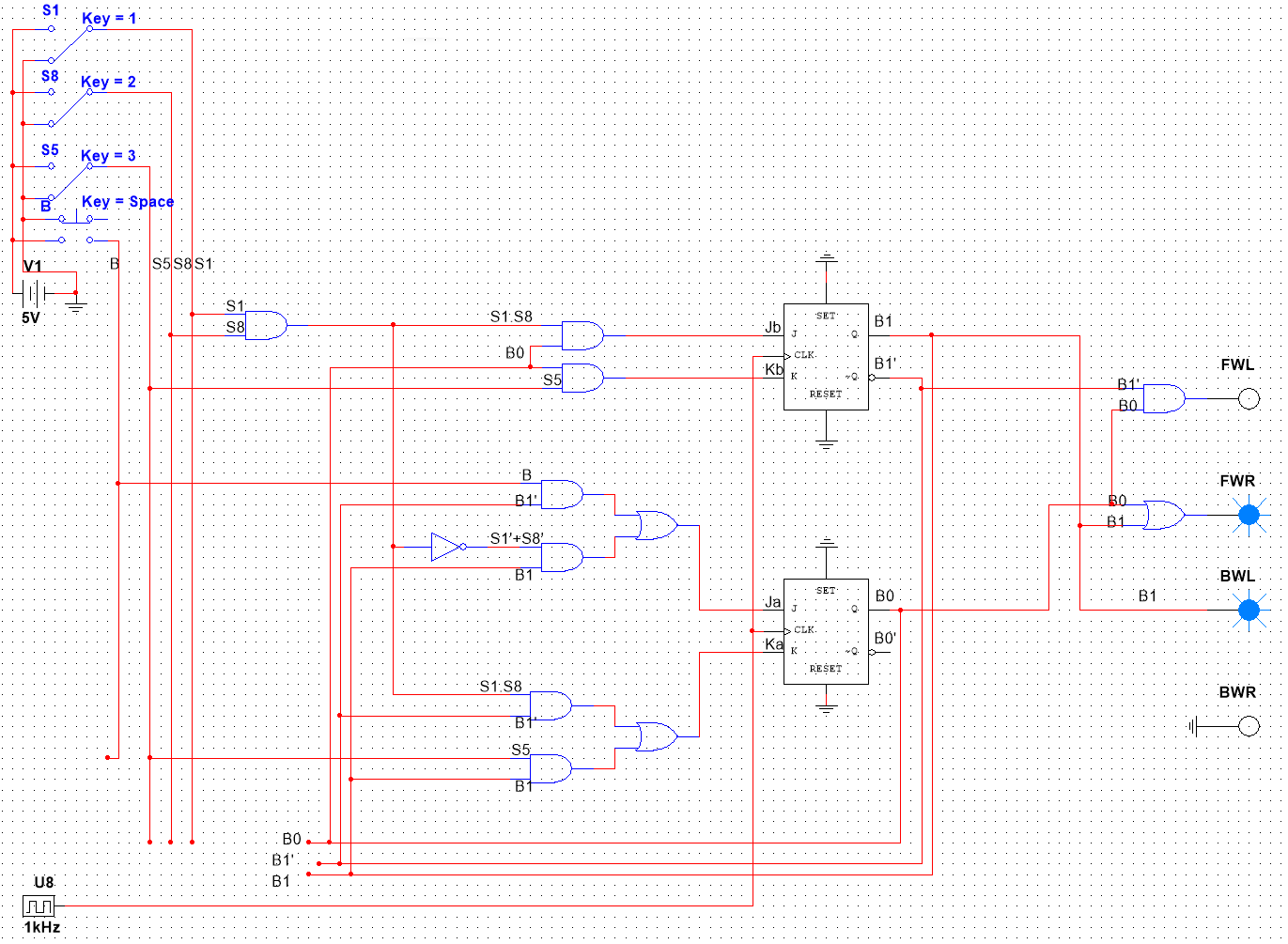


Figura 17 - Estado 3 (Dá a volta) - S1 e S8 a nível lógico baixo