

UFPR – DEPARTAMENTO DE INFORMÁTICA
CI068 CIRCUITOS LÓGICOS — TRABALHO 2 — ANOTAÇÕES
ANDRIELI LUCI GONÇALVES

1 DESCRIÇÃO DO PROJETO

Este trabalho apresenta um sistema de controle de semáforos em um cruzamento, em que os semáforos S1, S2 e S3 regulam o fluxo de veículos e o semáforo P controla a travessia de pedestres. Os semáforos de carros podem ser verdes, amarelos ou vermelhos, enquanto o semáforo de pedestres pode ser verde ou vermelho.

A troca de cores dos semáforos ocorre a cada pulso de *clock*, seguindo uma sequência específica. Quando um pedestre pressiona o botão de travessia, o semáforo P fica verde, garantindo a travessia segura. Para isso, todos os semáforos de veículos devem ficar vermelhos antes que o semáforo P mude para verde.

Para desenvolver o projeto, foi necessário definir: um diagrama de máquina de estados (seguindo o padrão Moore), a codificação dos estados e das cores dos semáforos, a tabela de transições de estados, os mapas de Karnaugh, tabelas-verdade e, por fim, o circuito propriamente dito.

2 DIAGRAMA DE MÁQUINA DE ESTADOS

O diagrama foi elaborado considerando um total de nove estados, que são: E1 (0000), E2 (0001), E3 (0010), E4 (0011), E5 (0100), E6 (0101), E7 (0110), E8 (0111), E9 (1000). O primeiro estado, E1, é o inicial, enquanto os três últimos estados, E7, E8 e E9, são específicos para a travessia de pedestres

Caso o pedestre esteja nos estados E1, E3 ou E5 (momentos em que um dos semáforos está verde), o circuito sempre transita para estados onde a cor de um dos semáforos é amarela, como E2, E4 e E6, independentemente de o botão de pedestres ter sido pressionado ou não. Caso o botão tenha sido clicado, o circuito armazena essa informação e aguarda a transição do verde para o amarelo, para então seguir para o vermelho.

Caso o botão tenha sido pressionado em E2, E4 ou E6, os próximos estados são, respectivamente, E7, E8 e E9; caso contrário, E3, E5 e E1. No que tange aos momentos E7, E8 e E9, não importa a ação do botão, seu próximos estados sempre serão E3, E5 e E1.

A partir dessa lógica, os ciclos de deslocamento de veículos e de pedestres passa a ter um fluxo inteligente. Isso evita, por exemplo, que uma pessoa pressione várias vezes ou mantenha o botão de pedestres pressionado, bloqueando a travessia de veículos por um período indeterminado.

3 COMPONENTES PRINCIPAIS DO CIRCUITO

3.1 LATCH S-R

O *latch S-R* possui duas entradas principais: S (*Set*) e R (*Reset*), e duas saídas Q0 e Q1 (sendo Q1 a versão negada de Q0). Para desenvolvê-lo, foram utilizadas quatro portas lógicas *NAND*.

Além dessas entradas e saídas principais, o *latch* possui uma entrada adicional chamada E (*Enable*), que indica se o *latch* estará ativo ou não para suas ações. Quando E está ativado, o *latch* responde às entradas S e R, ajustando as saídas Q0 e Q1 de acordo com os sinais recebidos. Quando E está desativado, o *latch* ignora as entradas S e R, mantendo o estado atual de Q0 e Q1.

3.2 FLIP-FLOP D (COM RESET)

O *flip-flop D*, por sua vez, foi desenvolvido a partir da combinação de dois *latches S-R*: a entrada S do primeiro *latch* depende da saída de um multiplexador 2:1 e a entrada R equivale ao S negado. Enquanto isso, o segundo *latch* possui como entradas as próprias saídas Q0 e Q1 do primeiro *latch*.

Para uma ação importante do circuito, que é a redefinição para o seu estado inicial, foi implementado um mecanismo de *reset*. Por meio do multiplexador 2:1 com seletor R, é indicado se o *flip-flop* armazenará 0 (se R estiver ativado) ou o valor da próxima entrada (se R estiver desativado). Essa configuração é do tipo síncrona, ou seja, o sinal de reset só tem efeito quando um evento de *clock* ocorre. O *flip-flop* só é

resetado quando a borda do *clock* está presente e o sinal de reset é ativado ao mesmo tempo.

Neste contexto, as entradas *E (Enable)* dos *latches* são substituídas por entradas *Clk* (referentes ao *clock* do *flip-flop*). O primeiro *latch* recebe *Clk* em sua versão negada, enquanto o segundo *latch* recebe a versão normal de *Clk*.

3.3 MULTIPLEXADOR

O multiplexador possui duas entradas, *D0* e *D1*, e um bit de seleção, identificado como *S*. Sua saída é representada como *Y*. Com base nas restrições do circuito, o multiplexador foi simplificado em relação à sua estrutura tradicional. Em vez de utilizar múltiplas portas lógicas, ele foi implementado com uma única porta lógica *AND*, ativada quando *S* e *D1* estão negados e *D0* está em sua forma normal. Isso garantirá que sempre que *S* seja 0, o valor de *D0* será selecionado, e quando *S* seja 1, o valor de *D1* será selecionado.

3.4 DECODIFICADOR

Assim como o multiplexador, o decodificador foi simplificado com base nas restrições do projeto, uma vez que nem todos os casos serão expressos no circuito — possibilitando simplificações de lógicas combinacionais e remoção de saídas.

O decodificador é composto por duas entradas, *S0* e *S1*, e três saídas: *D0* (representada pela negação das duas entradas), *D1* (equivalente ao valor de *S0*) e *D2* (equivalente ao valor de *S1*).

O objetivo, neste cenário, é decodificar as cores dos semáforos de veículos. Para cada uma das combinações 00, 01 e 10, uma das saídas é ativada, permitindo que apenas seu respectivo *LED* seja aceso.

4 CIRCUITO COMPLETO

Por fim, tem-se de fato o circuito de controle dos semáforos. Ele é composto por quatro *flip-flops D*, representando cada um dos bits do estado atual (*A0*, *A1*, *A2* e *A3*). A entrada *Clk* de cada *flip-flop* é conectada ao componente padrão de *clock* da ferramenta *Logisim Evolution*. A entrada *R* é conectada a um botão denominado

“Reset”, enquanto as entradas D são conectadas às saídas C0, C1, C2 e C3 de um circuito combinacional, cujo objetivo é definir o próximo estado do circuito. Além disso, este mesmo circuito combinacional também depende de uma entrada B, indicada no projeto como o botão de travessia de pedestres.

As saídas dos *flip-flops*, além de serem conectadas às entradas do circuito do próximo estado, também estão conectadas às entradas do circuito das saídas. Este circuito possui sete saídas: S1_0, S1_1, S2_0, S2_1, S3_0, S3_1 e P. Cada bit dos semáforos S1, S2 e S3 é conectado a uma entrada do decodificador, totalizando três decodificadores. Suas respectivas saídas são: (S1R, S1Y, S1G), (S2R, S2Y, S2G) e (S3R, S3Y, S3G), em que R, Y e G indicam, respectivamente, *red* (vermelho), *yellow* (amarelo) e *green* (verde).

Há, também, um *latch* S-R que avalia o comportamento do botão de travessia de pedestres. Sua entrada S equivale a um *AND* entre o botão B e a negação da saída P, evitando que as duas entradas S e R fiquem ativas ao mesmo tempo. Enquanto isso, R equivale a um *OR* entre P e o botão Reset.

No lado direito do circuito, há uma simulação das ruas do trabalho, bem como dos semáforos de carros (S1, S2 e S3) e do semáforo de pedestres, com seus respectivos LEDs. O propósito dessa parte é apenas estética, com vistas a promover uma visualização menos abstrata das etapas realizadas pelo circuito.

5 COMO FAZER O CIRCUITO FUNCIONAR?

Além de ativar as configurações do *Logisim Evolution* na seção de simulação, é necessário manter o botão de Reset pressionado por alguns segundos para inicializar o circuito, devido ao seu caráter síncrono. Esta etapa é importante, pois, inicialmente, alguns fios do circuito podem estar com valores indefinidos, o que tende a impactar na execução correta das sequências.

Para observar o momento de travessia de pedestres, basta clicar sobre o botão B, localizado abaixo do botão de Reset.

Para “reiniciar” as etapas do circuito, deve-se apenas seguir o passo a passo descrito no primeiro parágrafo (de pressionar o botão de Reset).