Sincronização de Semáforos

Daniel C. de Lins Neto
Ciência da computação
Universidade Federal do Paraná – UFPR
Curitiba, Brasil
daniellins@ufpr.br

I. INTRODUÇÃO

O circuito foi produzido com o propósito de organizar o cruzamento hipotético da Figura 1.

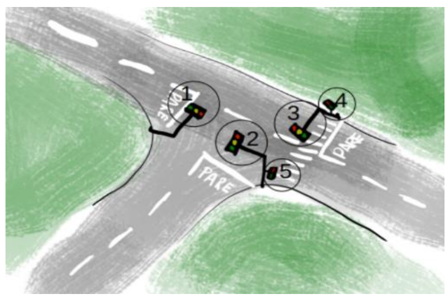

Figura 1. Retirada de [1].

O objetivo era fazer com que os semáforos 1 e 3 funcionassem de maneira contrária ao 2, e quando botão do semáforo de pedestre fosse pressionado, após o próximo sinal amarelo todos deveriam ficar vermelhos, exeto o 4 e 5, que deveriam ficar verdes. Todo o circuito, desde da prototipação da máquina de moore, até a diagramação das tabelas verdades foram feitas pelo autor no *Digital* [2].

II. DESENVOLVIMENTO

Por início procurou-se compreender bem do que se tratava o problema. Notou-se que os semáforos 1 e 3 trabalhavam de forma igual em qualquer caso, de maneira análoga, os semaforos de pedestre 4 e 5 também, fazendo com que assim, nos dois casos, ambos pudessem ser tratados como um semáforo só. Com isso em vista, foi escolhida com qual tipo de máquina de estados o circuito funcionária, sendo selecionada a de Moore por ser mais fácil de ser sincronizada por um clock.

Para a coonfecção do sistema os semáforos 1 e 3 foram chamados de "A", o 2 de "B", os 4 e 5 de "C" e o botão de "D". Então primeriramente trabalhou-se o sistema como se não existisse botão, obtendo a Figura 2.

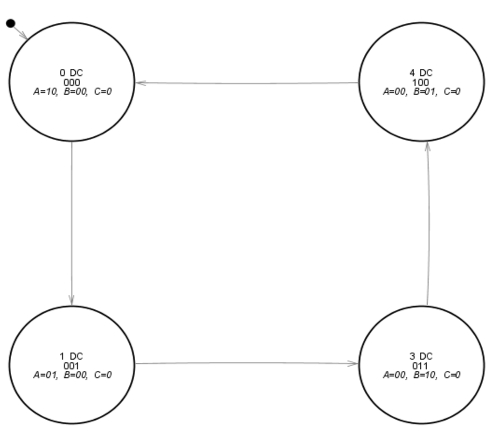

Figura 2. Sendo 10 verde, 01 amarelo e 00 vermelho.

Após isso, criou-se mais 2 estados, que transformariam "C" em 1 quando o botão fosse apertado e "A" ou "B" fossem amarelos, e mais dois estados que fariam com que "C" se transformace em 1 após um estado de quando o botão fosse apertado, caso "A" ou "B" fossem verdes, resultando na Figura 3.

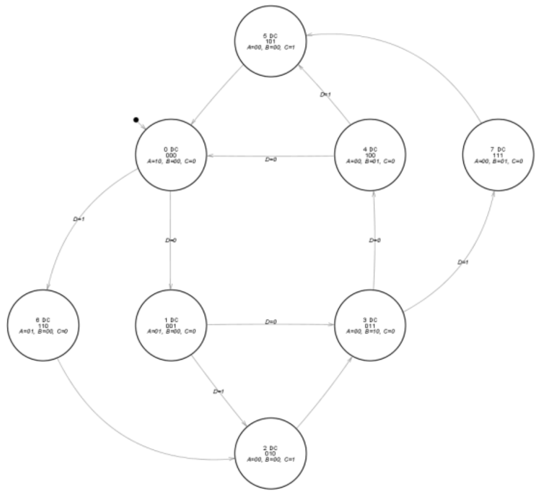

Figura 3. Resultado final.

Após isso foi criada a tabela verdade, presente na figura 4.

Z2n	Z1n	Z0n	D	Z2n+1	Z1n+1	Z0n+1	A1	A0	B1	B0	C
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	1	0	0	0	0
0	0	1	0	0	1	1	0	1	0	0	0
0	0	1	1	0	1	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	0	0	1	0	0
0	1	1	1	1	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	1	0	1	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0	0	1
1	1	0	0	0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0	1	0	0	0
1	1	1	0	1	0	1	0	0	0	1	0
1	1	1	1	1	0	1	0	0	0	1	0

Figura 4. Sendo Zxn+1 o próximo estado.

Gerando as simplificações dos circuitos foi obtido a Figura 5.

$$Z_2^{n+1} = (D \wedge \overline{Z_0^n} \wedge \overline{Z_1^n}) \vee (Z_0^n \wedge Z_1^n)$$

$$Z_1^{n+1} = (D \wedge \overline{Z_2^n}) \vee (Z_0^n \wedge \overline{Z_1^n} \wedge \overline{Z_2^n}) \vee (\overline{Z_0^n} \wedge Z_1^n)$$

$$Z_0^{n+1} = (\overline{D} \wedge \overline{Z_1^n} \wedge \overline{Z_2^n}) \vee (D \wedge \overline{Z_0^n} \wedge \overline{Z_1^n} \wedge Z_2^n) \vee (\overline{D} \wedge \overline{Z_0^n} \wedge \overline{Z_2^n}) \vee (D \wedge Z_1^n \wedge \overline{Z_2^n}) \vee (Z_0^n \wedge Z_1^n \wedge Z_2^n)$$

$$Z_0^{n+1} = (\overline{D} \wedge \overline{Z_1^n} \wedge \overline{Z_2^n}) \vee (D \wedge \overline{Z_0^n} \wedge \overline{Z_1^n} \wedge Z_2^n) \vee (D \wedge Z_0^n \wedge Z_1^n) \vee (Z_0^n \wedge Z_1^n \wedge Z_2^n) \vee (\overline{Z_0^n} \wedge Z_1^n \wedge \overline{Z_2^n})$$

$$Z_0^{n+1} = (\overline{D} \wedge \overline{Z_1^n} \wedge \overline{Z_2^n}) \vee (D \wedge \overline{Z_0^n} \wedge \overline{Z_1^n} \wedge Z_2^n) \vee (D \wedge Z_1^n \wedge \overline{Z_2^n}) \vee (Z_0^n \wedge Z_1^n \wedge Z_2^n) \vee (\overline{Z_0^n} \wedge Z_1^n \wedge \overline{Z_2^n})$$

$$A1 = \overline{Z_0^n} \wedge \overline{Z_1^n} \wedge \overline{Z_2^n}$$

$$A0 = (Z_0^n \wedge \overline{Z_1^n} \wedge \overline{Z_2^n}) \vee (\overline{Z_0^n} \wedge Z_1^n \wedge Z_2^n)$$

$$B1 = Z_0^n \wedge Z_1^n \wedge \overline{Z_2^n}$$

$$B0 = (\overline{Z_0^n} \wedge \overline{Z_1^n} \wedge Z_2^n) \vee (Z_0^n \wedge Z_1^n \wedge Z_2^n)$$

$$C = (\overline{Z_0^n} \wedge Z_1^n \wedge \overline{Z_2^n}) \vee (Z_0^n \wedge \overline{Z_1^n} \wedge Z_2^n)$$

Figura 5.

Apartir disso foi criado o circuito, representado pela Figura 6.

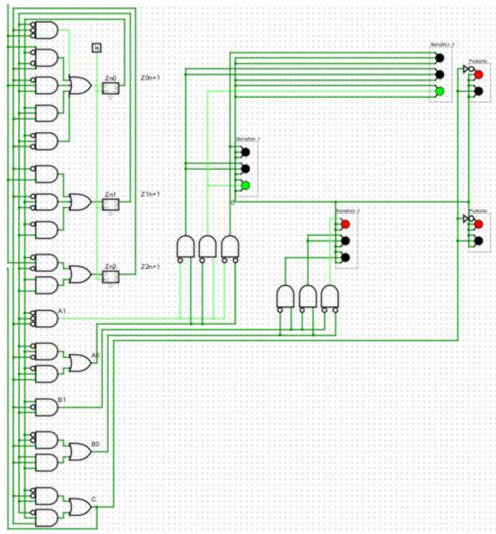

Figura 6.

E por último, foi criado um latch tipo SR, que pode ser visto na Figura 7, usado para armazenar a informação de quando o botão for pressionado, sendo resetado quando o semáforo de pedestre estiver verde.

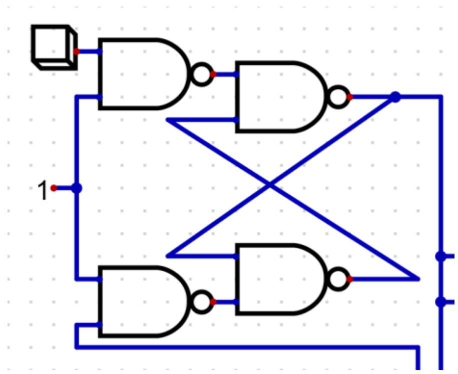

Figura 7. Latch SR com enable fixo em 1.

III. CONCLUSÃO

Com a construção do circuito, foi desenvolvido ainda mais o domínio do autor sobre portas lógicas e máquinas de estados, além da percepção da já possibilidade de programar objetos que tenham uso prático no dia a dia.

IV. REFERÊNCIAS

[1] Disponível em: https://ufprvirtual.ufpr.br/pluginfile.php/1678571/mod_resource/content/0/Trabalho2.pdf

[2] Disponível em: https://ufprvirtual.ufpr.br/pluginfile.php/1678571/mod_resource/content/0/Trabalho2.pdf