

Universidade Tecnológica Federal do Paraná
Campus Apucarana
Engenharia de Computação

Luis Henrique Ferracciu Pagotto Mendes, RA: 2272016

Relatório 2

Introdução à Placa DE10-Lite

LRCO7A - Lógica Reconfigurável
Professor: Marcelo de Oliveira

Apucarana - PR
2024

1 Introdução:

Este projeto tem como objetivo desenvolver um circuito lógico combinacional para controlar um LED com base no estado de quatro switches (chaves). O circuito é projetado para acionar o LED sempre que duas ou mais chaves estiverem fechadas. Utilizando o software Quartus para simulação em VHDL e o Logisim para modelagem visual, o circuito foi construído com portas lógicas básicas, como AND e OR, de forma a atender às condições especificadas.

A convenção adotada considera que um switch fechado representa o valor lógico 0, enquanto um switch aberto é 1. Esta lógica foi aplicada para garantir que o circuito funcione corretamente, acendendo o LED nas combinações desejadas de switches. O projeto inclui uma tabela verdade para validar os estados possíveis das chaves e suas respectivas saídas, assegurando que o comportamento do circuito esteja conforme o esperado.

2 Tabela Verdade do Circuito:

A tabela a seguir representa a combinação para as entradas dos switches 1 a 4, em que a combinação deles, na utilização das portas lógicas gera as possíveis saídas representadas na tabela pela coluna z.

sw1	sw2	sw3	sw4	z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Tabela 1: Entradas sw1, sw2, sw3 e sw4 e saída z

3 Mapa de Karnaugh e Equação Booleana:

A partir da Tabela Verdade descrita na seção anterior, é possível definir a criação do Mapa de Karnaugh, que permite a combinação das variáveis do circuito, facilitando a visualização de identificações de padrões presentes no mesmo e, assim, contribuindo na remoção de redundâncias do projeto.

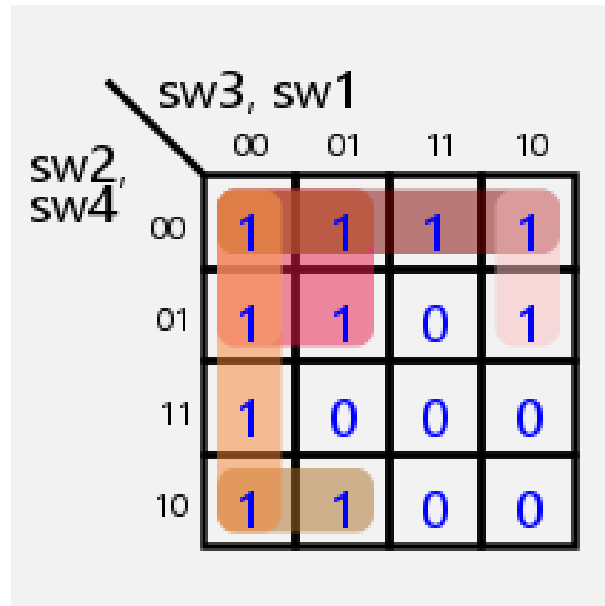


Figura 1: Mapa de Karnaugh que representa o circuito.

A partir do Mapa de Karnaugh apresentado acima, também é possível definir a equação booleana que melhor representa o circuito, sendo uma forma textual de representá-lo, pois é possível definir portas lógicas, relações e ligações do circuito em apenas uma linha. Para o projeto em questão, a equação booleana que melhor o representa é descrita abaixo:

$$z = !sw2 \&\& !sw4 \vee !sw2 \&\& !sw3 \vee !sw2 \&\& !sw1 \vee !sw4 \&\& !sw3 \vee !sw3 \&\& !sw1$$

4 Código VHDL comentado:

Código 1: Código VHDL

```

-----PACOTES-----
library ieee;
use ieee.std_logic_1164.all;
-----ENTIDADES-----

--A entidade projeto2 utiliza as portas sw1, sw2, sw3 e sw4
--como entrada do sinal e a saída z
entity projeto2 is
--portas sw1 a sw4 sao entradas switches, STD_LOGIC e o tipo
--de entrada (binario, assumindo valores 0 ou 1)
    port (sw1, sw2, sw3, sw4: in STD_LOGIC;
          z: out STD_LOGIC); --porta z e uma saída
--em LED tambem assumindo valores 0 ou 1 (apagado ou aceso)
end entity;
-----PROJETO-----
architecture projeto2 of projeto2 is --utiliza a arquitetura do
--proprio projeto para definir a funcao e operacao
--entre as portas usadas

```

```

begin --inicia o bloco de comando do projeto
--porta z recebe as combinacoes em que o led acende a partir dessas
--para o caso de o sw1 e sw4 ativos ao mesmo tempo, o LED nao acende
--mas se sw2 e/ou sw3 estiverem ativos tambem, o LED acendera
    z <= (NOT sw1 AND NOT sw2) OR
        (NOT sw1 AND NOT sw3) OR
        (NOT sw2 AND NOT sw3) OR
        (NOT sw2 AND NOT sw4) OR
        (NOT sw3 AND NOT sw4); --a porta z recebe o sinal
--a partir das combinacoes entre sw1, sw2, sw3 e sw4
end architecture;

```

5 Imagem da Simulação:

O projeto em questão, define que a saída somente estará em alta quando dois ou mais switches estiverem fechados simultaneamente, com exceção dos switches 1 e 4, pois o enunciado explicita que, durante o funcionamento da máquina de cópias, em hipótese alguma, os dois estarão fechados ao mesmo tempo. Desta forma, as combinações levarão uma saída em alta quando duas ou mais entradas estiverem como bit 0. Para isso, o a simulação foi gerada com valores aleatórios para 'sw1', 'sw2', 'sw3' e 'sw4', resultado em uma saída para z, a depender dos valores gerados. Como é possível ver na imagem a seguir:

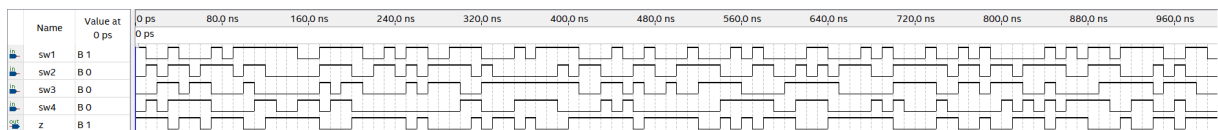


Figura 2: Simulação do circuito, com entradas aleatórias geradas pelo Quartus.

6 Imagem do Diagrama RTL:

A partir do código descrito no tópico 4, foi gerado um circuito capaz de representar o funcionamento da máquina de cópias. O circuito em questão, é gerado com a opção RTL Viewer, que permite uma visualização melhor da construção. Abaixo está uma figura gerada a partir do RTL Viewer, que demonstra a entrada a partir dos switches, passando pelas portas lógicas e levando até a saída z.

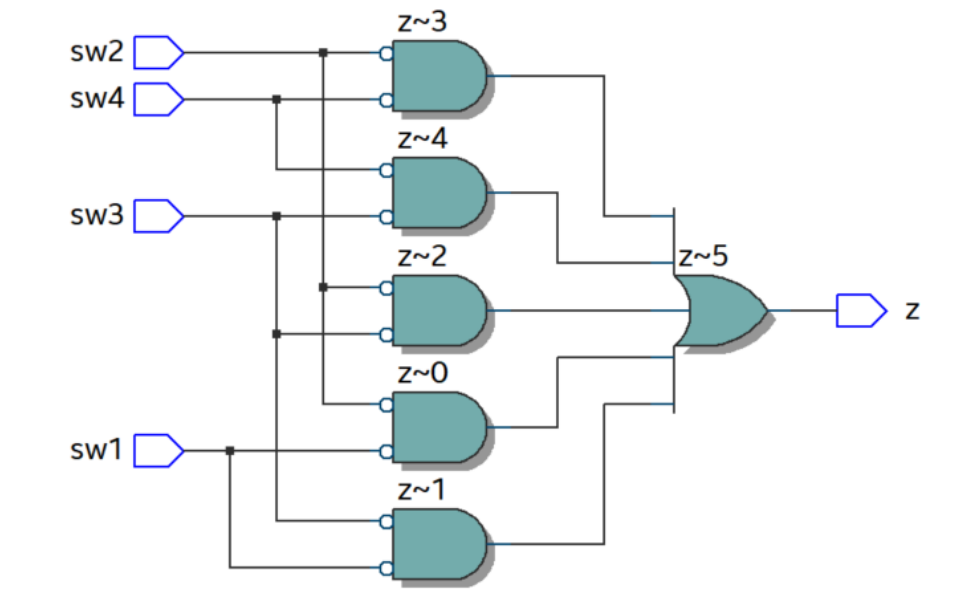


Figura 3: Circuito STL gerado pelo Quartus.