

CIRCUITOS LÓGICOS COMBINACIONAIS

Júlia Pessoa Souza

Prática de Eletrônica Digital - Turma C
Faculdade Gama - Universidade de Brasília
15/0133294

João Pedro Vergara Coneglian

Prática de Eletrônica Digital - Turma C
Faculdade Gama - Universidade de Brasília
14/015974

RESUMO

O experimento consistiu em implementar em uma placa Basys 3 o alarme de um carro. Foram utilizadas quatro variáveis: a porta do motorista, os faróis, a ignição e o alarme. Com a seguinte legenda:
Porta - aberta: 1 fechada: 0
Ignição - ligada: 1 desligada: 0
Faróis - acesos: 1 apagados: 0
Alarme - ativado: 1 desativado: 0
O alarme deveria ser ativado se os faróis estivessem acesos e a ignição desligada ou se a porta estivesse aberta e a ignição ligada. Utilizamos as iniciais das palavras como variáveis.

1. INTRODUÇÃO

No circuitos lógicos combinacionais, a saída depende somente das combinações feitas com as variáveis de entrada, porque estes circuitos não possuem memória. Para construir estes circuitos é necessário montar uma expressão a partir da tabela verdade que representa uma situação já mencionada. Uma vez obtida a expressão do circuito, podemos simplificá-la para reduzir o número de portas utilizadas para montar o projeto. Para isso, pode-se usar álgebra booleana e mapa de Karnaugh. Depois de simplificar a expressão ao máximo, pode-se montar o circuito.

2. EXPERIMENTO

2.1 Fazendo o projeto no VIVADO

O experimento começou quando se abriu o VIVADO e criou-se um novo projeto. Na nova janela, selecionou-se a opção RTL Project, sem especificar as sources. Foi selecionado o menu Boards e nele a placa Basys3. Clicou-se em Finish.

Na janela nova, em Project Manager, foi selecionado o Add Sources e logo após a opção Add or create design sources. Após isso, criou-se um novo arquivo e

especificou-se seu tipo como sendo VHDL. Para definir o módulo, foram incluídas as entradas (P, I e F) e a saída (A) do circuito.

De volta a janela principal do projeto, clicou-se duas vezes no arquivo criado para que seu código fosse editado. Com o código já pré criado pelo programa, adicionou-se em architecture, logo após "begin" o comando "A <= (NOT(I) AND F) OR (P AND I);". Salvou-se as alterações do arquivo.

Para simular o circuito sem testbench, selecionou-se a opção Run Simulation na janela Flow Navigator e, depois, Run Behavioral Simulation. Selecionando as portas lógicas, clicou-se com o botão direito do mouse em cada uma e forçou-se o clock desejado. Geraram-se sinais periódicos com frequências diferentes para cada entrada, abrangendo todas as oito combinações. Foi escolhido o período de 10 ns para a entrada F, 20 ns para a entrada I e 40 ns para a entrada P, e executou-se a simulação por 40 ns (Run for 40 ns). Em Force clock, o campo Value radix estava em binário, o campo Leading edge value em 0, Trailing edge value em 1, e o campo Period com o valor de 10 ns.

2.2 Implementando o projeto na placa Basys3

Após os procedimentos realizados em 2.1 terem sido validados, mapearam-se as entradas e saídas do circuito nos periféricos da placa. Cada entrada foi ligada a uma chave (switch) e a saída foi ligada a um led. Esse mapeamento foi feito em um arquivo de extensão .xdc baixado previamente pelo moodle da disciplina, selecionado em Add Sources, na opção constraints. Antes de clicar em Finish, selecionou-se a opção Copy constraints files into project.

Na janela Sources, abriu-se o arquivo Basys3_Master.xdc e editou-se seu código retirando os comentários apenas para os periféricos que foram utilizados. Retirou-se os comentários das linhas correspondentes às 3 primeiras chaves e substituiu-se os nomes genéricos, que por padrão são sw[0], sw[1] e sw[2], pelos nomes das 3 entradas (P, I e F, respectivamente). Para mapear a saída em um led, editou-se e retirou-se o comentário de uma linha dos leds, alterando o nome padrão led[0] para A. Com isso, a saída

A foi mostrada no led U16.

Executou-se a síntese em Run Synthesis na janela Flow Navigator e clicou-se em OK. Selecionou-se a opção para abrir o projeto sintetizado e clicou-se em Ok. Visualizou-se o esquemático do circuito clicando-se em Schematic no menu Synthesis.

Clicou-se no menu Tools e selecionou-se Edit Device Properties. Nessa janela, efetuaram-se as seguintes alterações:

- General: Enable Bitstream Compression: TRUE;
- Configuration: Configuration Rate (MHz): 33 ;
- Configuration Modes: selecionar Master SPI x 4.

Em seguida, fechou-se o projeto sintetizado e selecionou-se em Run Implementation. Ao final da implementação, a janela Implementation Completed apareceu. Selecionou-se Generate Bitstream. Terminado o processo, a janela Bitstream Generation Completed apareceu. Passando para a última etapa, selecionou-se Open Hardware Manager e conectou-se a placa ao computador via cabo USB.

Clicou-se em Open Target e em seguida em Auto Connect. Selecionou-se Program Device e, em seguida, o modelo da placa. Na janela Program Device, escolheu-se a opção Program, e aguardou-se o led DONE da placa acender, indicando que o dispositivo foi corretamente programado. Após isso acontecer, o circuito se encontrou pronto para ser testado, alterando-se os valores das entradas nas chaves e verificando a saída no led.

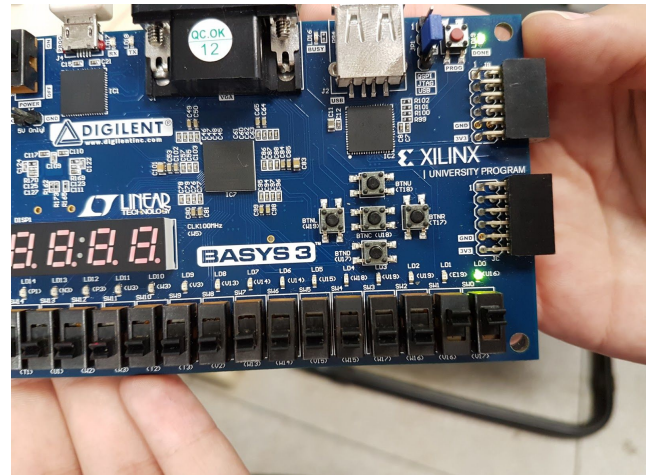


Figura 2: Placa Basys3 com led aceso

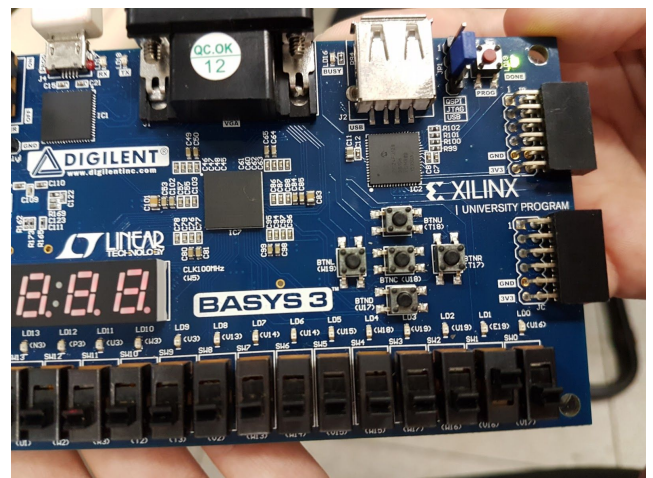


Figura 3: Placa Basys3 com led apagado

2.3 Figuras e Tabelas

Porta (P)	Ignição (I)	Faróis (F)	Alarme (A)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figura 1: Tabela verdade

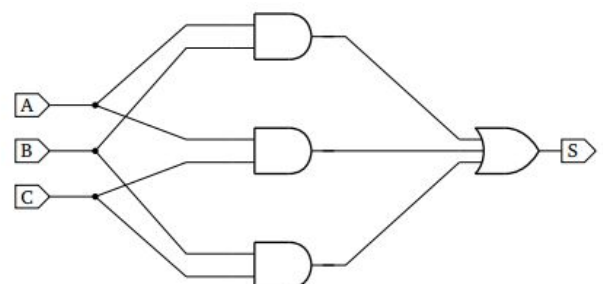


Figura 4: Circuito da apostila.

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figura 5: Tabela verdade do circuito da apostila

3. RESULTADOS

Em sala de aula, vimos que a expressão booleana simplificada para este problema é:

$$A = /I.F + P.I \quad (1)$$

E, de acordo com a expressão e a tabela verdade representada na figura 3, chegamos ao resultado esperado. Depois de implementar o circuito na placa, testamos todos os valores da tabela. Quando o alarme tem valor 1, o led de saída acende, e, quando o alarme tem valor 0, o led fica apagado. De acordo com as figuras percebe-se algumas situações, considerando que a porta v17 é a porta, a v16 é a ignição e a w16 são os faróis. E ainda, quando a chave está para cima, temos o valor 1, e, para baixo, o valor 0. As figuras mostram a situação em que a porta está aberta, a ignição ligada e os faróis apagados, e o led está aceso e a situação em que a ignição está ligada, os faróis estão apagados e a porta está fechada, e, o led está apagado. Ou seja, o alarme (led) foi acionado na situação correta.

Do circuito da apostila podemos formar a expressão

$$S = A.B + A.C + B.C$$

Este circuito possui 3 entradas e 1 saída. Sua tabela

verdade está representada na figura 5. O circuito faz uma saída do tipo A e B ou A e C ou B e C. Pode ser utilizado por exemplo para automação residencial. Considerando que a saída seja o ar condicionado da sala e as entradas, a luz da sala, a televisão e a janela. O ar será ligado quando a janela estiver fechada e a televisão ligada ou quando a janela estiver fechada e a luz acesa ou quando a televisão estiver ligada e a luz acesa.

4. DISCUSSÃO E CONCLUSÕES

Com base no experimento realizado pelo VIVADO utilizando a placa Basys 3 e nos cálculos teóricos, percebe-se que o resultado do circuito é o mesmo nas duas situações, tanto na prática quanto na teórica. Na implementação na placa, a posição das chaves condiz com os valores apresentados na tabela teórica, conforme o esperado. Todos as posições possíveis para as chaves A, B e C foram testadas e comparadas uma a uma com os valores obtidos na tabela.

Conclui-se então que o experimento foi bem sucedido.

5. REFERÊNCIAS

- [1] I. Idoeta e I. Valeije, “Elementos de Eletrônica Digital”, Érica Ltda, São Paulo, 2008.
- [2] R. Tocci, N.Widmer e G. Moss, “Sistemas Digitais: Princípios e aplicações” Pearson.
- [3] G. Bezerra, “Apostila Prática de Eletrônica Digital I”, 2017.
- [4] G. Bezerra, “Tutorial de Simulação e Implementação no Vivado”, 2017.