



**UNIVERSIDADE FEDERAL DO CEARÁ – CAMPUS SOBRAL**  
**CURSO DE ENGENHARIA DA COMPUTAÇÃO**  
**DISCIPLINA: ELETRÔNICA DIGITAL**  
**PROFESSORES: ROMULO NUNES DE CARVALHO ALMEIDA E DAVID**  
**NASCIMENTO COELHO**

**PRÁTICA Nº 02**  
**CIRCUITOS COMBINACIONAIS**

<b>ALUNO</b>	<b>MATRÍCULA</b>
<b>Klayver Ximenes Carmo</b>	<b>427651</b>

**Sobral – CE**

**2021**

## SUMÁRIO

1. INTRODUÇÃO .....	3
2. OBJETIVOS .....	4
3. MATERIAL UTILIZADO .....	4
4. PROCEDIMENTO EXPERIMENTAL .....	5
5. QUESTIONÁRIO .....	21
6. CONCLUSÃO .....	23
7. REFERÊNCIAS BIBLIOGRÁFICAS .....	24

## **1. INTRODUÇÃO**

Quando nos deparamos com problemas do mundo real, é comum encontrarmos alguns que tenham relação direta com lógicas booleanas, desde os mais simples, como acender uma lâmpada, até os mais complexos, como no gerenciamento de mais de um equipamento.

Este relatório visa exemplificar as maneiras de resolução, através da lógica booleana, de possíveis problemas ou desafios, utilizando portas lógicas para a construção de circuitos lógicos e técnicas para a otimização dos mesmos.

Para a construção dos circuitos lógicos, foi utilizado o software de simulação Proteus.

## **2. OBJETIVOS DA PRÁTICA**

- Descobrir formas diferentes de implementação de uma mesma função lógica;
- Utilizar a Álgebra Booleana na simplificação de circuitos lógicos;
- Desenvolver no aluno a capacidade de resolver problemas envolvendo circuitos combinacionais;
- Inserir o aluno no contexto de desenvolvimento de projetos de eletrônica digital, estimulando o mesmo a, primeiramente, testar a ideia do circuito em simulador.

## **3. MATERIAL UTILIZADO**

- Portas lógicas (AND, OR, XOR, NOT);

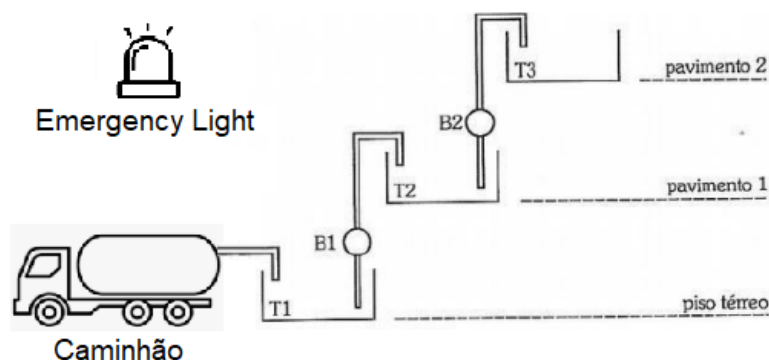
#### 4. PROCEDIMENTO EXPERIMENTAL

A simulação inicia-se na resolução do problema citado na questão 5.1, onde tem-se como requisito o controle de 4 sistemas baseado em dados fornecidos de 3 sensores embutidos no problema.

Contextualizando, tem-se um caminhão que serve para abastecimento de 3 tanques de glicose de uma determinada indústria, sendo localizados, cada um, em pavimentos diferentes.

Cada tanque possui um sensor, informando o seu nível atual, podendo ser cheio ou não cheio, além das bombas utilizadas para transferir o líquido para os tanques 2 e 3. Todo o processo consta na Figura 1.

Figura 1 – Representação do sistema de abastecimento.



Fonte: Adaptação roteiro da prática

Os 3 tanques do sistema tem seus sensores com representação T1, T2 e T3, assim como as bombas B1 e B2.

O caminhão, responsável pelo abastecimento do sistema, tem representação com a letra C, onde pode ser controlado seu funcionamento, entre ligado e desligado, acontecendo o mesmo funcionamento para as bombas.

A luz de emergência, como requisitos abordados na questão, é acionada quando ocorre algum problema externo que influencia o sistema de abastecimento, como o mau funcionamento de um sensor.

Também tem seu funcionamento lógico, podendo ser ligada em casos de erros.

Inicialmente montam-se os possíveis casos dos problemas, através da tabela verdade do mesmo. A Tabela 1 representa os possíveis eventos.

Tabela 1 – Tabela verdade do problema.

T3	T2	T1	C	E	B1	B2
0	0	0	1	0	0	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	1	1	0	1	1
1	0	0	0	1	0	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	0	0	0	0

Fonte: Autor

Na Figura 1, a saída C, representa o caminhão, onde tem nível lógico 1 (ligado) nos momentos em que os tanques não estão completamente cheios, e nível lógico 0 (desligado), quando os tanques estão completamente cheios ou quando tem algum problema no sistema, fazendo seu desligamento por segurança.

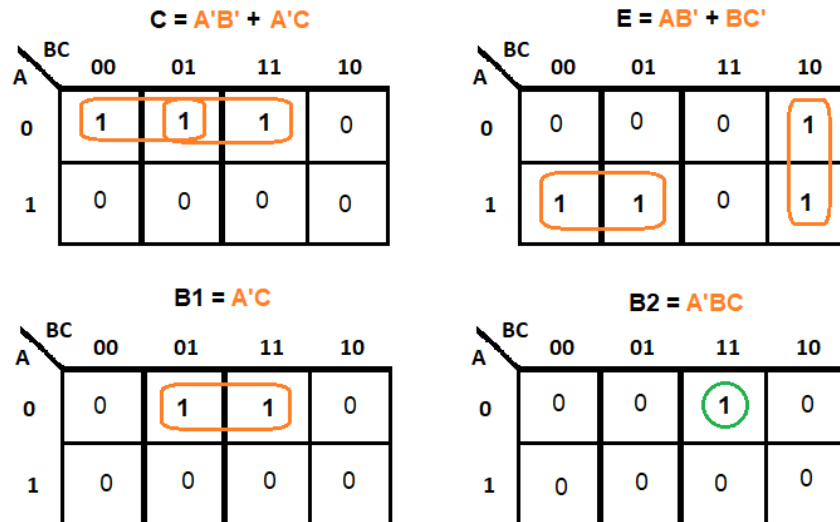
Em paralelo se usa a luz de emergência, representada pela letra E, onde é ligada quando ocorre alguma inconsistência nos eventos, como acontece no caso em que é informado pelo sensor T3 que o tanque correspondente está cheio enquanto os tanques T2 e T3 estão vazios. Este caso em específico é representado pelos valores 100 dos tanques T3, T2 e T1, respectivamente, sendo acionado, assim, a luz de emergência.

As bombas B1 e B2 são acionadas quando os tanques inferiores a ela estão cheios, e os dos pavimentos superiores estão vazios. Podendo ser abordado no caso 011, onde os tanques T1 e T2 estão cheios e o T3 está vazio, sendo assim necessário o acionamento das bombas B1 e B2.

As bombas possuem dois casos em que não são ligadas, é o momento em que nenhum tanque está cheio, caso 000, sendo necessário apenas o funcionamento do caminhão para encher o tanque T1, e nos casos em que acontece algum mau funcionamento no sistema, sendo interrompidas.

Na Figura 2 é possível verificar os mapas de Karnaugh montados para a simplificação do sistema para a montagem do circuito lógico.

Figura 2 – Mapas de Karnaugh do problema 5.1.

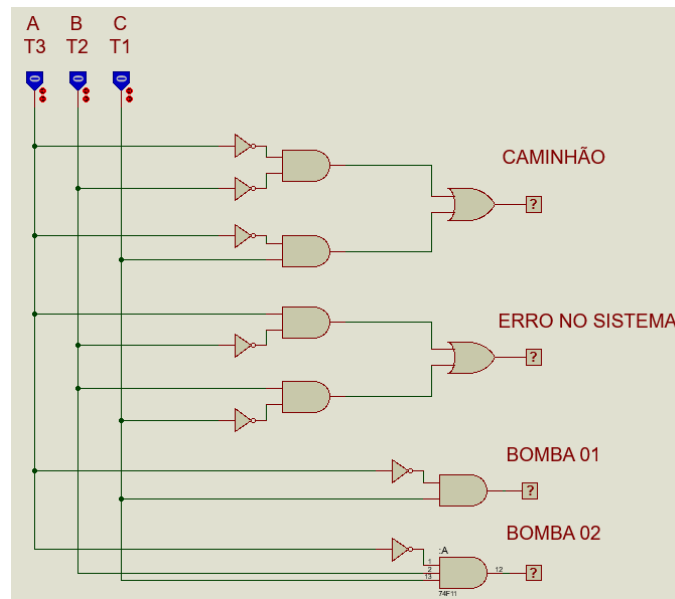


Fonte: Autor

As letras A, B e C, correspondem aos sensores T3, T2 e T1, respectivamente, onde é feita uma simplificação para cada dispositivo que será controlado pelo circuito lógico.

Na Figura 3 é apresentado o circuito resultante após a simplificação do sistema utilizando o mapa de Karnaugh, contendo 4 saídas booleanas, responsáveis por fazer o controle do fluxo do caminhão no abastecimento, controle do funcionamento da luz de emergência, caso haja inconsistência e das bombas B1 e B2, responsáveis por fazer a transferência do produto para os tanques de pavimentos superiores.

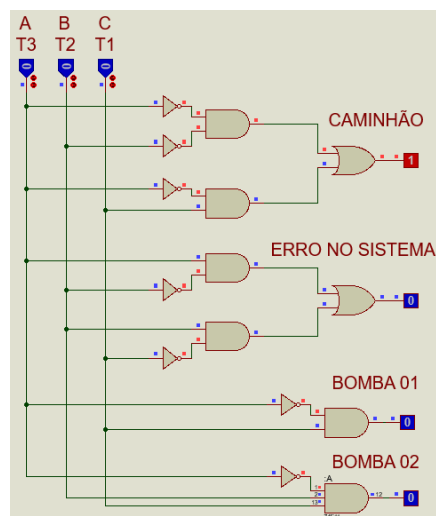
Figura 3 – Circuito lógico do problema 5.1.



Fonte: Autor

Com isso, é possível simular todos os possíveis casos presentes do problema em questão, desde a simulação de preenchimento do primeiro tanque, sem a utilização das bombas, o preenchimento dos tanques T2 e T3, onde necessita de pelo menos uma bomba para serem preenchidos, e até mesmo o evento de erro, que simula algum mau funcionamento do sistema, onde pode ser ligado um sinalizador.

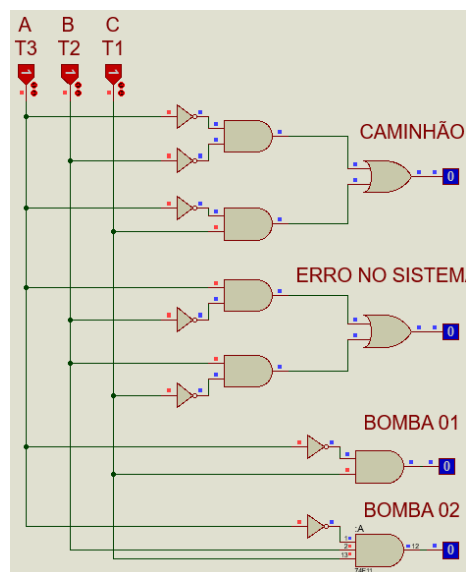
Figura 4 – Simulação enchimento do tanque T1.



Fonte: Autor

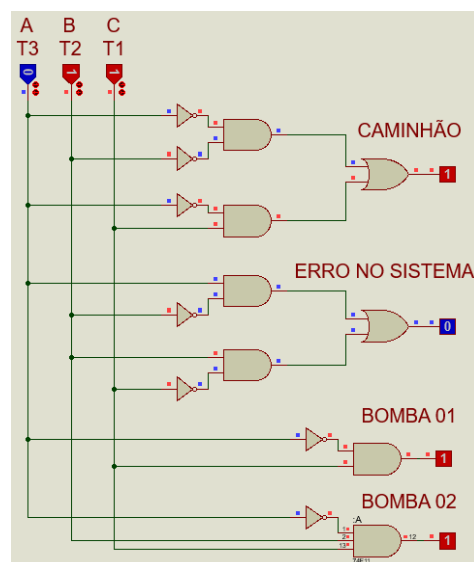


Figura 5 – Simulação de todos os tanques cheios.



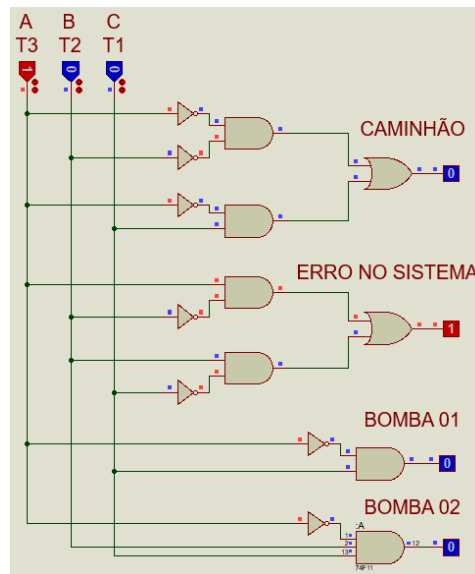
Fonte: Autor

Figura 6 – Simulação com o uso das bombas B1 e B2.



Fonte: Autor

Figura 7 – Simulação de erro no sistema.



Fonte: Autor

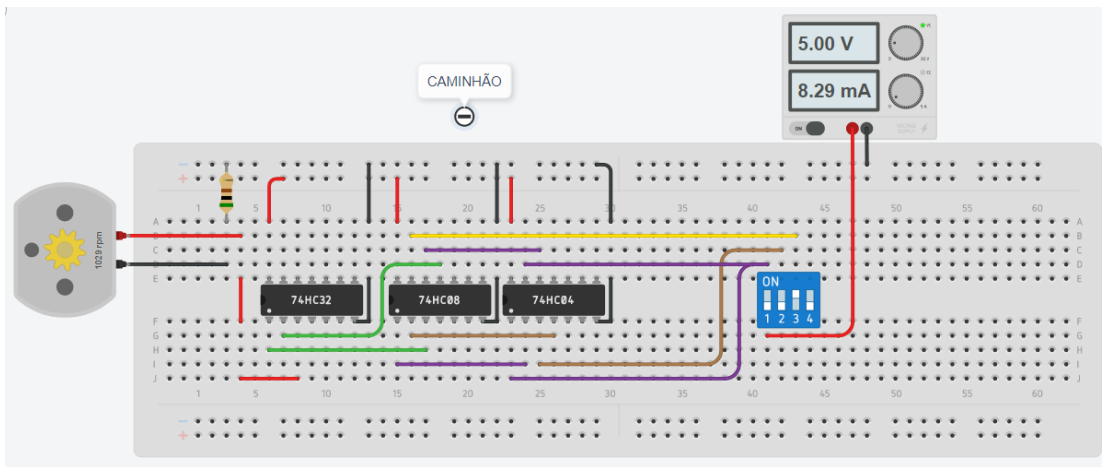
As figuras 4, 5, 6 e 7 apresentam alguns dos possíveis eventos que podem ser encontrados durante a execução do abastecimento.

Foi utilizado também o software de simulação TinkerCad, onde foi construído de forma individual o circuito para cada sistema envolvido.

Para a representação dos dados do sistema, foi utilizado um switch, onde o número 1 corresponde ao sensor T3, o 2 ao T2 e o 3 ao T1.

A figura 5.1.1 apresenta o circuito utilizado para o funcionamento do caminhão, representado por um motor.

Figura 5.1.1 – Simulação do circuito para o caminhão.

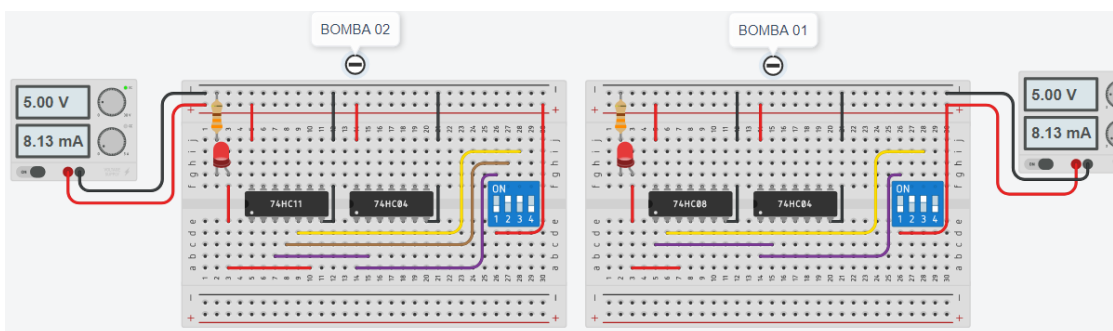


Fonte: Autor

Na figura 5.1.1, é simulado o caso em que o T1 está cheio e o caminhão continua ligado para o preenchimento do T2.

A figura 5.1.2 apresenta os circuito para o funcionamento das bombas, onde o mesmo simula o caso em que T1 e T2 estão cheios, sendo necessário então o acionamento das duas bombas para o preenchimento do tanque T3

Figura 5.1.2 – Simulação do circuito para as bombas.

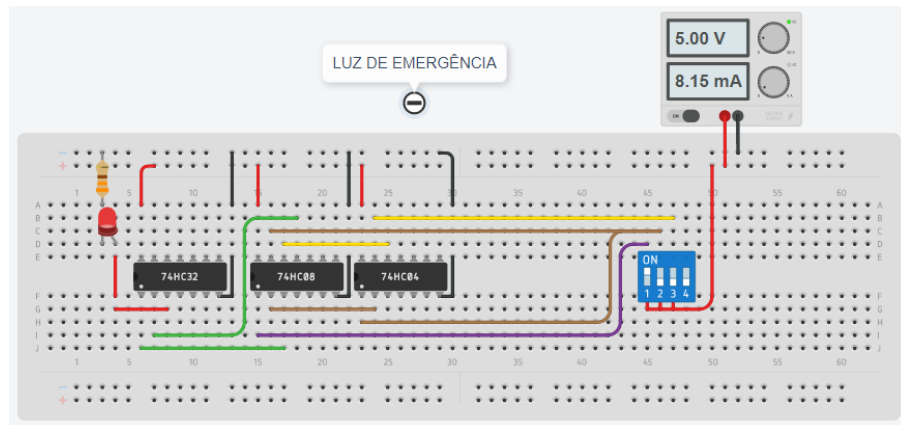


Fonte: Autor

Também se faz necessário um circuito para o funcionamento da luz de emergência, para caso haja algum mau funcionamento do sistema. O mesmo é

apresentado na figura 5.1.3, onde simula o caso em que o T3 acusa estar cheio sem o T2 e T1 estarem, caso esse que é considerado como erro para apresentação do sistema.

Figura 5.1.3 – Simulação de erro.



Fonte: Autor

Os circuitos apresentados nas figuras 5.1.1, 5.1.2 e 5.1.3 utilizam alguns componentes em comuns, como o gerador de energia, necessário para alimentar o sistema e os componentes dos mesmos, um switch para simular o funcionamento dos sensores, leds e motor para simular quando os dispositivos envolvidos devem ser acionados e circuitos integrados, onde são as representações das portas lógicas utilizadas.

O CI 74HC04 representa a porta NOT, 74HC08 a porta AND, 74HC11 a porta AND com 3 inputs e 74HC32 para a porta OR.

Na questão 5.2, é abordado o conceito de paridade, utilizado muitas vezes em detecções de erros, incluindo um dado para validação da integridade dos dados, garantindo assim o controle sobre o processo de transferência e corrompimento dos dados [1].

No seu contexto, um bit é adicionado na informação principal a ser repassada, e informado, é uma palavra que contém 4 bits de informação, onde é adicionado um bit de paridade.

Para a paridade par, é verificado a quantidade de 1's ou 0's dos bits de informação, sendo adicionado um bit de paridade para deixar quantidade par dos mesmos.

Temos como exemplo a palavra com os bits 0111, onde tem uma quantidade ímpar de 1's, com isso, o bit de paridade par é 1, ficando então o valor a ser transmitido 10111, ficando com a quantidade par de 1's.

O inverso acontece com o bit de paridade ímpar, usando o exemplo anterior da palavra 0111, onde a quantidade de bits 1's já é ímpar, o bit de paridade é 0, resultando no valor 00111, tendo uma quantidade ímpar na transmissão.

A tabela 2 apresenta todos os possíveis valores que podem ser formados com uma palavra de 4 bits, sendo representados pelas letras A, B, C e D.

Tabela 2 – Bits de paridade par e ímpar de uma palavra de 4 bits.

ABCD	P I
0000	0 1
0001	1 0
0010	1 0
0011	0 1
0100	1 0
0101	0 1
0110	0 1
0111	1 0
1000	1 0
1001	0 1
1010	0 1
1011	1 0
1100	0 1
1101	1 0
1110	1 0
1111	0 1

Fonte: Autor

As saídas resultantes representadas pelas letras P e I correspondem aos bits de paridade par e ímpar, respectivamente.

Com isso, é possível gerar as expressões para o circuito lógico, como é apresentado na figura 8 a expressão encontrada para o circuito utilizado para o bit de paridade par e na figura 9 para o bit de paridade ímpar.

Figura 8 – Expressão para o bit de paridade par para uma palavra de 4 bits.

$$P = (A \oplus B) \oplus (C \oplus D)$$

CD \ AB	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Fonte: Autor

Figura 9 – Expressão para o bit de paridade ímpar para uma palavra de 4 bits.

$$I = ((A \oplus B) \oplus (C \oplus D))'$$

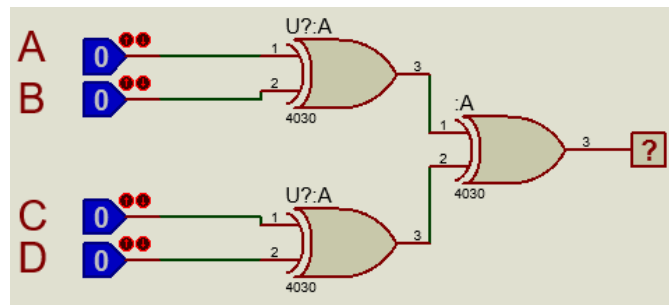
CD \ AB	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
10	0	1	0	1

Fonte: Autor

Como é possível perceber, é encontrado um padrão nas diagonais dos mapas de Karnaugh, padrões esses que são descritos pela porta lógica XOR, além de ter uma saída invertida nos bits de paridade par e ímpar, que é representado no circuito por uma porta lógica do tipo NOT, onde inverte toda e qualquer saída [2].

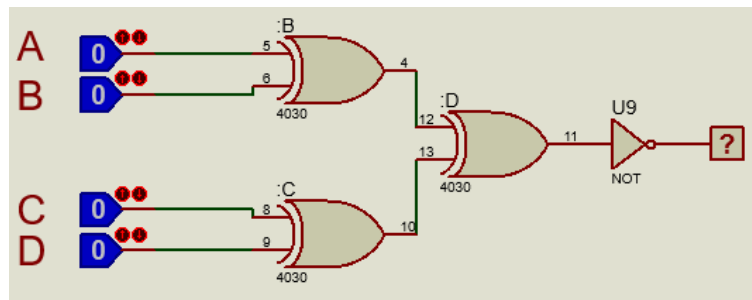
O circuito para tal problema com o bit de paridade par é apresentado na figura 10 e na figura 11 para o bit de paridade ímpar.

Figura 10 – Circuito do bit de paridade par para uma palavra de 4 bits.



Fonte: Autor

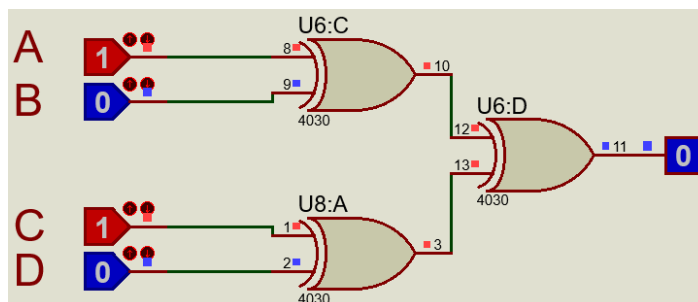
Figura 11 – Circuito do bit de paridade ímpar para uma palavra de 4 bits.



Fonte: Autor

As figuras 12 e 13 apresentam dois exemplos de bits de paridades par e ímpar, respectivamente. É utilizado a contagem dos bits 1's para fazer a contagem e definição do bit de paridade.

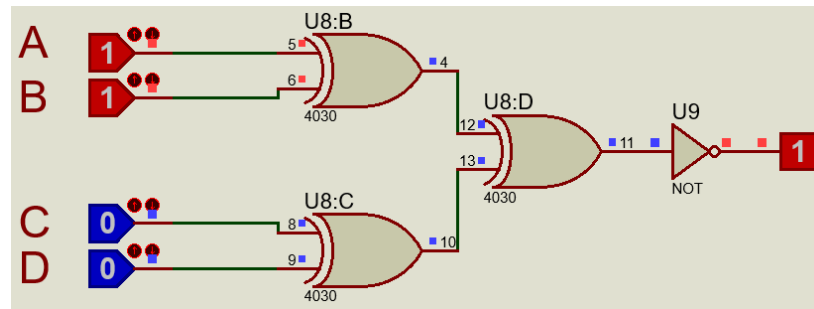
Figura 12 – Exemplo do bit de paridade par para a palavra com bits 1010.



Fonte: Autor

O exemplo do bit de paridade par utiliza a palavra 1010, tendo o bit de paridade 0, devido a quantidade de bits 1 já ser par na palavra. A informação resultante é 11010.

Figura 13 – Exemplo do bit de paridade ímpar para a palavra com bits 1100.



Fonte: Autor

O exemplo do bit de paridade ímpar, na figura 13, utiliza a palavra 1100, tendo o bit de paridade 1, devido a quantidade de bits 1 ser par na palavra. A informação resultante é 11100.

Na questão 5.3 é requisitado um circuito para a detecção da quantidade ímpar de chaves fechadas num circuito de 3 bits, sendo a chave fechada equivalente ao nível 0.

A tabela 3 apresenta todos os possíveis casos de informação formados por 3 bits.

Tabela 3 – Casos de verificação das chaves fechadas.

ABC	S
000	1
001	0
010	0
011	1
100	0
101	1
110	1
111	0

Fonte: Autor



A saída representada pela letra S indica os eventos que contém uma quantidade ímpar de chaves fechadas, sendo os casos 000, 011, 101 e 110.

A figura 14 apresenta a obtenção da expressão gerada pela tabela verdade do problema.

Figura 14 – Expressão para o circuito do problema 5.3.

$$S = A'B'C' + A'BC + AB'C + ABC'$$

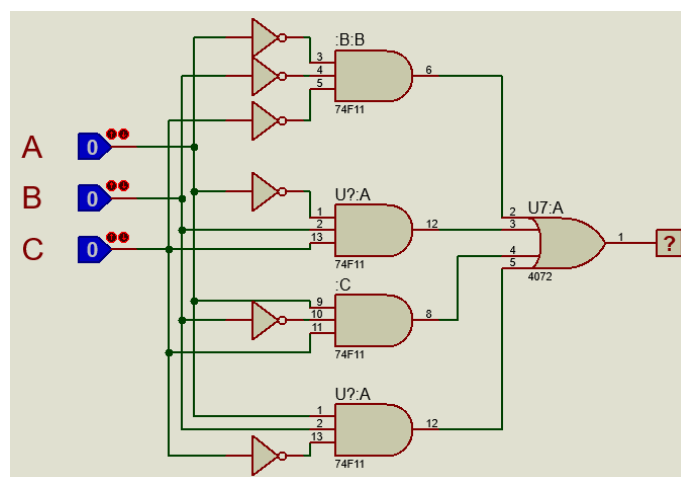
$$S = (A \oplus B \oplus C)'$$

		BC			
		00	01	11	10
A	0	1		1	
	1		1		1

Fonte: Autor

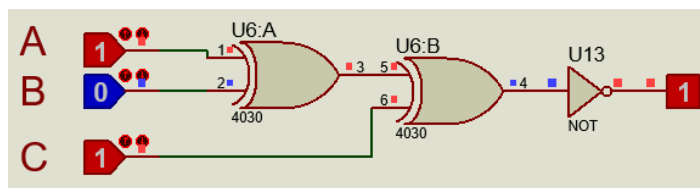
Como visto, é resultante uma expressão contendo uma porta lógica não universal, a XOR. O circuito correspondente à expressão, em azul, é apresentado na figura 15, e o que contém as portas XOR é apresentado na figura 16.

Figura 15 – Circuito do problema 5.3.



Fonte: Autor

Figura 16 – Circuito com portas XOR do problema 5.3.

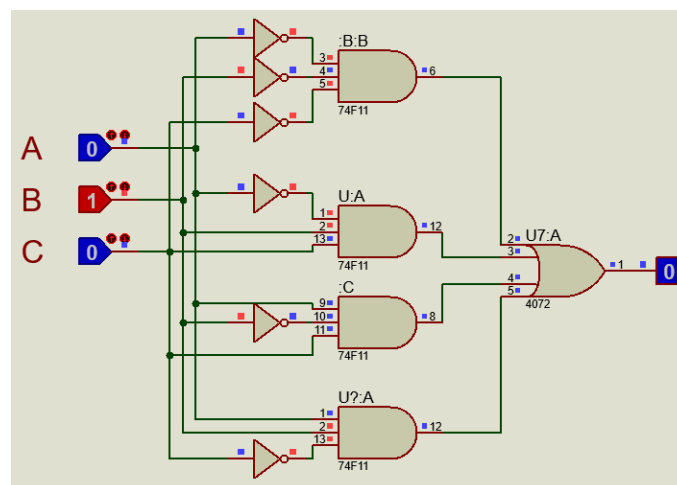


Fonte: Autor

A figura 16 também apresenta um exemplo utilizando os bits 101, resultando na saída 1, já que a quantidade de bits 0 é ímpar.

A figura 17 apresenta um exemplo com os bits 010 utilizando a expressão contendo apenas as portas lógicas universais.

Figura 17 – Exemplo do problema 5.3 utilizando os bits 010.



Fonte: Autor

A saída do evento apresentado na figura 17 é 0, já que a quantidade de chaves fechadas é par.

Para a questão 5.4 é pedido um circuito para fazer a identificação de números primos em binário de até 7 bits, ou seja, fazer o reconhecimento dos números 2, 3, 5 e 7.

Com isso, é possível montar a tabela verdade com os possíveis casos, que é apresentado na tabela 4.

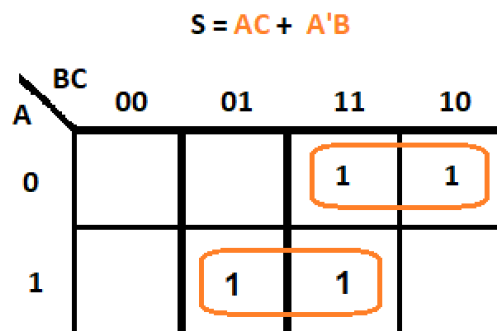
Tabela 3 – Casos de verificação das chaves fechadas.

ABC	S
000	0
001	0
010	1
011	1
100	0
101	1
110	0
111	1

Fonte: Autor

A figura 18 apresenta o desenvolvimento da expressão lógica obtido após a tabela verdade do problema.

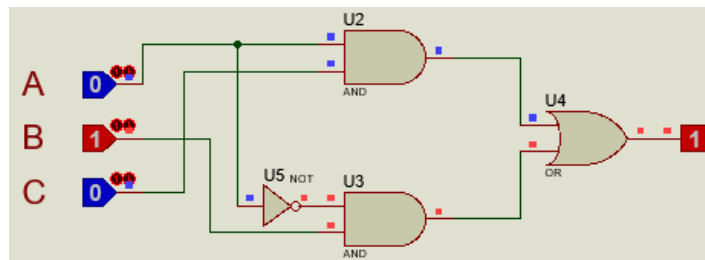
Figura 18 – Expressão para o circuito do problema 5.4.



Fonte: Autor

Observa-se que é possível montar o circuito apenas com as portas lógicas universais, que é apresentado na figura 19.

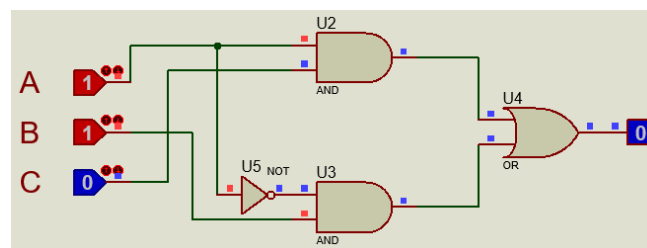
Figura 19 – Expressão para o circuito do problema 5.4.



Fonte: Autor

A figura 19 também apresenta um exemplo utilizando os bits 010, correspondente ao 2, que é um número primo, com isso, tendo saída 1. A figura 20 apresenta um exemplo utilizando o número 6, que não é um número primo, representado por 110 em binário.

Figura 20 – Expressão para o circuito do problema 5.4 utilizando o número 6.

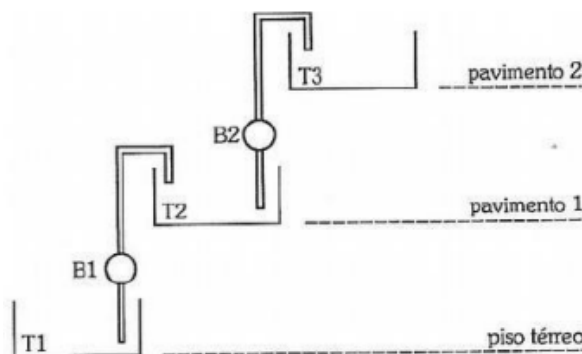


Fonte: Autor

É possível perceber que a saída resultou no valor 0, visto que o número 6, com representação binária 110, não é um número primo.

## 5. QUESTIONÁRIO

**5.1)** Projete um circuito lógico para abastecer três tanques (T1, T2 e T3) de glicose em pavimentos distintos em uma indústria de balas e biscoitos, através do controle de duas bombas conforme esquematizado na Figura 2. O abastecimento principal é feito por caminhão-tanque que fornece o produto diretamente ao T1 disposto no piso térreo localizado à entrada da empresa. Desenvolva o projeto supondo que o nível máximo de T1 seja controlado pelo caminhão, coloque os sensores de controle de nível nas caixas, convencie as variáveis e desenhe o circuito final.



### OBS:

- O sensor pode estar em mau funcionamento.
- Uma queda de energia pode aleatoriamente deixar as entradas em uma situação “Absurda”. Portanto, faça considerações na tabela verdade para manter a segurança do projeto.

**5.2)** Projete um circuito que faça a geração e detecção de paridade par e ímpar para uma palavra de 4 bits. O bit de paridade é usado para detectar erros em transmissões geralmente de caracteres. Para se evitar erros na transmissão é adicionado um bit de paridade, ou seja um bit a mais que segue duas regras que podemos se dizer simples. O código gerado é do tipo paridade par, quando a somatória de dos 1's presentes na informação total, incluindo o bit de teste, resulta em um número par e acontece de forma análoga para o caso do código do tipo paridade ímpar. Podemos considerar que os bits A, B, C e D como relativos à informação, e P e I como os de paridade par e ímpar, a serem gerados.

**5.3)** Elabore um circuito para, em um conjunto de 3 chaves, detectar o número ímpar destas chaves fechadas. Convencionar que a chave fechada equivale ao nível 0.

**5.4)** Nesta prática será montado um circuito simples que identifica se um número binário entre 0 e 7 faz parte do conjunto dos números primos. A função lógica deste circuito pode ser representada como:  $f(A, B, C) = (2, 3, 5, 7)$ . Esta forma de descrever a função lógica indica que a saída do circuito é uma função das entradas A, B e C, o circuito é descrito por uma soma de produtos, sendo que os termos produtos são os números binários que representam os valores 2, 3, 5 e 7.

Tabela 1 – Tabela verdade do problema.

<b>Resistência [<math>\Omega</math>]</b>	<b>Estado do transformador</b>
50 a 1000	Bom
>1000 a $\infty$	Aberto
próximo de 0 ou 0	Curto

Fonte: Exemplo de fonte de tabela

## **6. CONCLUSÃO**

Nesta prática foi possível perceber a importância das simplificações das funções lógicas obtidas através das tabelas verdade, fazendo essa simplificação pelo método de mapa de Karnaugh.

Fazendo entendimento também das relações entre as implementações de uma mesma função lógica, utilizando portas lógicas não universais.

Além de aperfeiçoar os conhecimentos nas práticas simulacionais com a criação dos circuitos através das expressões lógicas obtidas nos problemas, fazendo sua relação com o mundo real.

## 7. REFERÊNCIA BIBLIOGRÁFICA

- [1] [Bit de paridade.](#)
- [2] [Portas XOR e XNOR.](#)