



SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE VIÇOSA · UFV  
CAMPUS FLORESTAL

**INTRODUÇÃO AOS SISTEMAS LÓGICOS**  
**TRABALHO PRÁTICO 1**

**Grupo 6**

MARIA EDUARDA DUARTE LACERDA[5920]

ANA CLARA SOARES OLIVEIRA [5896]

Florestal - MG

2024

## Sumário

<b>1. INTRODUÇÃO</b>	<b>2</b>
<b>2. TABELA DA VERDADE</b>	<b>3</b>
<b>3. MAPAS DE KARNAUGH E EXPRESSÕES BOOLEANAS</b>	<b>4</b>
3.1 BIT DE PARIDADE	4
3.2 SAÍDA A	4
3.3 SAÍDA B	5
3.4 SAÍDA C	5
3.5 SAÍDA D	6
3.6 SAÍDA E	6
3.7 SAÍDA F	7
3.8 SAÍDA G	7
<b>4. DIAGRAMAS DE PORTAS LÓGICAS</b>	<b>8</b>
4.1 GERADOR DE BIT DE PARIDADE	8
4.2 SAÍDA A	9
4.3 SAÍDA B	9
4.4 SAÍDA C	10
4.5 SAÍDA D	10
4.6 SAÍDA E	11
4.8 SAÍDA G	12
<b>5. COMPILAÇÃO E EXECUÇÃO</b>	<b>12</b>
<b>6. CONCLUSÃO</b>	<b>13</b>
<b>7.REFERÊNCIAS</b>	<b>13</b>

## 1. INTRODUÇÃO

No trabalho prático da disciplina Introdução aos Sistemas Lógicos, o objetivo era desenvolver um circuito combinacional que, ao receber 6 bits aciona um display de sete segmentos que representa um caractere, também usando um bit de paridade para conferir a integridade da passagem da mensagem, acendendo o display com um caractere de erro escolhido pelo grupo.

Para a implementação do circuito, cada grupo foi instruído a seguir uma linha da tabela apresentada na Especificação, de forma que as combinações de bits indo do 00000 ao 10011 representavam um caractere diferente e, caso a combinação esteja fora do intervalo, o display deve ficar apagado. Sendo assim, foi necessária a montagem de uma tabela da verdade para achar o bit de paridade e os casos onde cada segmento deve ou não acender, também usando a representação instruída pela Especificação. Após isso, foram feitos mapas de Karnaugh e diagramas com portas lógicas para cada bit de saída e para o de paridade, sua expressão booleana minimizada.

## 2. ORGANIZAÇÃO E IMPLEMENTAÇÃO

A organização do nosso projeto foi feita a partir da criação de um repositório criado no Github[3], denominado “CircCombinacional” onde as integrantes puderam contribuir na composição da implementação de Verilog. As outras partes do projeto foram devidamente organizadas na documentação, seguindo a ordem pedida na especificação.

A figura a seguir mostra a organização da implementação em Verilog:



Figura 1 - Repositório com a implementação em Verilog

O módulo de verificação de paridade foi implementado utilizando a função lógica XOR(ou exclusivo), pois ao montarmos a tabela da verdade do bit de paridade gerado, foi perceptível que o XOR seria suficiente para calcular a paridade e depois compará-la ao que era esperado.

```
module VerParidade(
    input [5:0] X,
    output erropar
);

assign erropar = ( X[5] == ~(X[0] ^ X[1] ^ X[2] ^ X[3] ^ X[4]));

endmodule
```

Figura 2 - Módulo de verificação de paridade

Em relação ao módulo de mapeamento para o Display de 7 segmentos, ele foi implementado após a verificação de paridade e trata das entradas corretas, entradas com paridade errada e fora do intervalo, e as correspondentes saídas no Display por meio dos comandos always e if(condicional).

```
module MapeamentoDisplay (
    input [4:0] entrada,
    input erropar,
    output reg [6:0] segmentos
);

always @(*) begin
    if (!erropar) begin
        segmentos = 7'b1011110; // Erro de Paridade
    end else begin
        case (entrada)
            5'd0: segmentos = 7'b1010100; // M
            5'd1: segmentos = 7'b0111011; // Y
            5'd2: segmentos = 7'b0011100; // V
            5'd3: segmentos = 7'b0110111; // H
            5'd4: segmentos = 7'b0001111; // T
            5'd5: segmentos = 7'b1111110; // 0
            5'd6: segmentos = 7'b1110111; // A
            5'd7: segmentos = 7'b1011011; // S
            5'd8: segmentos = 7'b1110011; // Q
            5'd9: segmentos = 7'b0101010; // W
            5'd10: segmentos = 7'b0001110; // L
            5'd11: segmentos = 7'b1111111; // 8
            5'd12: segmentos = 7'b0111101; // D
            5'd13: segmentos = 7'b1111001; // 3
            5'd14: segmentos = 7'b1101101; // 2
            5'd15: segmentos = 7'b1001111; // E
            5'd16: segmentos = 7'b0110111; // K
            5'd17: segmentos = 7'b0110011; // 4
            5'd18: segmentos = 7'b0001011; // R
            5'd19: segmentos = 7'b1111011; // 9
        endcase
    end
end
```

Figura 3 - Módulo Mapeamento do Display

Em relação ao erro de paridade, o caractere escolhido pelo grupo foi a da figura a seguir:

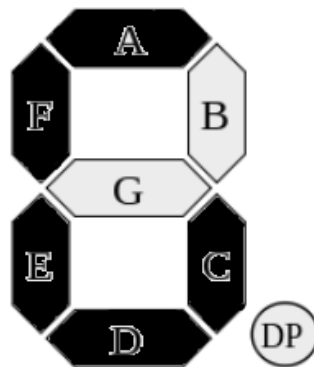


Figura 4 - Saída erro de paridade

### 3. TABELA DA VERDADE

REP	X16	X8	X4	X2	X1	BP	A	B	C	D	E	F	G
M	0	0	0	0	0	1	1	0	1	0	1	0	0
Y	0	0	0	0	1	0	0	1	1	1	0	1	1
V	0	0	0	1	0	0	0	0	1	1	1	0	0
H	0	0	0	1	1	1	0	1	1	0	1	1	1
T	0	0	1	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	1	1	0
A	0	0	1	1	0	1	1	1	1	0	1	1	1
S	0	0	1	1	1	0	1	0	1	1	0	1	1
Q	0	1	0	0	0	0	1	1	1	0	0	1	1
W	0	1	0	0	1	1	0	1	0	1	0	1	0
L	0	1	0	1	0	1	0	0	0	1	1	1	0
8	0	1	0	1	1	0	1	1	1	1	1	1	1
D	0	1	1	0	0	1	0	1	1	1	1	0	1
3	0	1	1	0	1	0	1	1	1	1	0	0	1
2	0	1	1	1	0	0	1	1	0	1	1	0	1
E	0	1	1	1	1	1	1	0	0	1	1	1	1
K	1	0	0	0	0	0	0	1	1	0	1	1	1
4	1	0	0	0	1	1	0	1	1	0	0	1	1

R	1	0	0	1	0	1	0	0	0	0	1	0	1
9	1	0	0	1	1	0	1	1	1	1	0	1	1
-	1	1	0	1	0	0	0	0	0	0	0	0	0
-	0	1	0	1	0	1	0	0	0	0	0	0	0
-	0	1	0	1	1	0	0	0	0	0	0	0	0
-	1	1	0	1	1	1	0	0	0	0	0	0	0
-	1	1	1	0	0	0	0	0	0	0	0	0	0
-	0	1	1	0	0	1	0	0	0	0	0	0	0
-	0	1	1	0	1	0	0	0	0	0	0	0	0
-	1	1	1	0	1	1	0	0	0	0	0	0	0
-	0	1	1	1	0	0	0	0	0	0	0	0	0
-	1	1	1	1	0	1	0	0	0	0	0	0	0
-	1	1	1	1	1	0	0	0	0	0	0	0	0
-	0	1	1	1	1	1	0	0	0	0	0	0	0

Caractere sendo representado

Bits de entrada

Bit de paridade

Segmentos do display

## 4. MAPAS DE KARNAUGH E EXPRESSÕES BOOLEANAS

Os Mapas de Karnaugh são essenciais para o sucesso no processo de minimização das expressões booleanas, eles foram feitos utilizando o site Canva[1] e preenchidos de acordo com a Tabela da Verdade mostrada anteriormente.

### 4.1 BIT DE PARIDADE

**Mapa de Karnaugh bit de paridade**

		X4 X2 X1							
		000	001	011	010	110	111	101	100
X8 X16	00	0 <b>1</b>	1 <b>0</b>	3 <b>1</b>	2 <b>0</b>	6 <b>1</b>	7 <b>0</b>	5 <b>1</b>	4 <b>0</b>
	01	8 <b>0</b>	9 <b>1</b>	11 <b>0</b>	10 <b>1</b>	14 <b>0</b>	15 <b>1</b>	13 <b>0</b>	12 <b>1</b>
	11	24 <b>1</b>	25 <b>0</b>	27 <b>1</b>	26 <b>0</b>	28 <b>0</b>	29 <b>1</b>	31 <b>0</b>	30 <b>1</b>
	10	16 <b>0</b>	17 <b>1</b>	19 <b>0</b>	18 <b>1</b>	20 <b>1</b>	21 <b>0</b>	23 <b>1</b>	22 <b>0</b>

- Expressão Booleana Minimizada:

$$\neg(X16 \wedge X8 \wedge X4 \wedge X2 \wedge X1)$$

### 4.2 SAÍDA A

**Mapa de Karnaugh saída A**

		X4 X2 X1							
		000	001	011	010	110	111	101	100
X8 X16	00	0 <b>1</b>	1 <b>0</b>	3 <b>0</b>	2 <b>0</b>	6 <b>1</b>	7 <b>1</b>	5 <b>1</b>	4 <b>0</b>
	01	8 <b>1</b>	9 <b>0</b>	11 <b>1</b>	10 <b>0</b>	14 <b>1</b>	15 <b>1</b>	13 <b>1</b>	12 <b>0</b>
	11	24 <b>0</b>	25 <b>0</b>	27 <b>0</b>	26 <b>0</b>	28 <b>0</b>	29 <b>0</b>	31 <b>0</b>	30 <b>0</b>
	10	16 <b>0</b>	17 <b>0</b>	19 <b>1</b>	18 <b>0</b>	20 <b>0</b>	21 <b>0</b>	23 <b>0</b>	22 <b>0</b>

- Expressão Booleana Minimizada:

$$\neg X_{16} \cdot \neg X_4 \cdot \neg X_2 \cdot \neg X_1 + \neg X_{16} \cdot X_4 \cdot X_1 + \neg X_{16} \cdot X_4 \cdot X_2 + \neg X_{16} \cdot X_8 \cdot X_2 \cdot X_1 + \\ X_{16} \cdot \neg X_8 \cdot \neg X_4 \cdot X_2 \cdot X_1$$

#### 4.3 SAÍDA B

**Mapa de Karnaugh saída B**

		X4 X2 X1							
X8 X16		000	001	011	010	110	111	101	100
		0	1	3	2	6	7	5	4
00		0	1	1	0	1	0	1	0
01		8	9	11	10	14	15	13	12
		1	1	1	0	1	0	1	1
11		24	25	27	26	28	29	31	30
		0	0	0	0	0	0	0	0
10		16	17	19	18	20	21	23	22
		1	1	1	0	0	0	0	0

- Expressão Booleana Minimizada:

$$\neg X_{16} \cdot \neg X_4 \cdot X_1 + \neg X_{16} \cdot \neg X_2 \cdot X_1 + \neg X_8 \cdot \neg X_4 \cdot X_1 + \neg X_{16} \cdot X_4 \cdot X_2 \cdot \neg X_1 + \\ \neg X_{16} \cdot X_8 \cdot \neg X_2 + X_{16} \cdot \neg X_8 \cdot \neg X_4 \cdot \neg X_2$$

#### 4.4 SAÍDA C

**Mapa de Karnaugh saída C**

		X4 X2 X1							
X8 X16		000	001	011	010	110	111	101	100
		0	1	3	2	6	7	5	4
00		1	1	1	1	1	1	1	0
01		8	9	11	10	14	15	13	12
		1	0	1	0	0	0	1	1
11		24	25	27	26	28	29	31	30
		0	0	0	0	0	0	0	0
10		16	17	19	18	20	21	23	22
		1	1	1	0	0	0	0	0

- Expressão Booleana Minimizada:



$$\neg X_8 \cdot \neg X_4 \cdot \neg X_2 + \neg X_8 \cdot \neg X_4 \cdot X_1 + \neg X_{16} \cdot \neg X_8 \cdot X_2 + \neg X_{16} \cdot \neg X_4 \cdot X_2 \cdot X_1 + \\ \neg X_{16} \cdot X_4 \cdot \neg X_2 \cdot X_1 + \neg X_{16} \cdot X_8 \cdot \neg X_2 \cdot \neg X_1$$

#### 4.5 SAÍDA D

# Mapa de Karnaugh saída D

		X4 X2							
		X1							
X8	X16	000	001	011	010	110	111	101	100
	00	<sup>0</sup> 0	<sup>1</sup> 1	<sup>3</sup> 0	<sup>2</sup> 1	<sup>6</sup> 0	<sup>7</sup> 1	<sup>5</sup> 1	<sup>4</sup> 1
	01	<sup>8</sup> 0	<sup>9</sup> 1	<sup>11</sup> 1	<sup>10</sup> 1	<sup>14</sup> 1	<sup>15</sup> 1	<sup>13</sup> 1	<sup>12</sup> 1
	11	<sup>24</sup> 0	<sup>25</sup> 0	<sup>27</sup> 0	<sup>26</sup> 0	<sup>28</sup> 0	<sup>29</sup> 0	<sup>31</sup> 0	<sup>30</sup> 0
	10	<sup>16</sup> 0	<sup>17</sup> 0	<sup>19</sup> 1	<sup>18</sup> 0	<sup>20</sup> 0	<sup>21</sup> 0	<sup>23</sup> 0	<sup>22</sup> 0

- Expressão Booleana Minimizada:

$$\neg X_{16} \cdot \neg X_2 \cdot X_1 + \neg X_{16} \cdot \neg X_4 \cdot X_2 \cdot \neg X_1 + \neg X_{16} \cdot X_4 \cdot \neg X_2 + \neg X_{16} \cdot X_4 \cdot X_1 + \\ \neg X_{16} \cdot X_8 \cdot X_2 + X_{16} \cdot \neg X_8 \cdot \neg X_4 \cdot X_2 \cdot X_1$$

#### 4.6 SAÍDA E

# Mapa de Karnaugh saída E

X4 X2

X1

X8	X16	000	001	011	010	110	111	101	100
	00	1 <sup>0</sup>	0 <sup>1</sup>	1 <sup>3</sup>	1 <sup>2</sup>	1 <sup>6</sup>	0 <sup>7</sup>	1 <sup>5</sup>	1 <sup>4</sup>
	01	0 <sup>8</sup>	0 <sup>9</sup>	1 <sup>11</sup>	1 <sup>10</sup>	1 <sup>14</sup>	1 <sup>15</sup>	0 <sup>13</sup>	1 <sup>12</sup>
	11	0 <sup>24</sup>	0 <sup>25</sup>	0 <sup>27</sup>	0 <sup>26</sup>	0 <sup>28</sup>	0 <sup>29</sup>	0 <sup>31</sup>	0 <sup>30</sup>
	10	1 <sup>16</sup>	0 <sup>17</sup>	0 <sup>19</sup>	1 <sup>18</sup>	0 <sup>20</sup>	0 <sup>21</sup>	0 <sup>23</sup>	0 <sup>22</sup>

- Expressão Booleana Minimizada:

$$\neg X8 \cdot \neg X4 \cdot \neg X1 + \neg X16 \cdot \neg X4 \cdot X2 + \neg X16 \cdot \neg X8 \cdot X4 \cdot \neg X2 + \neg X16 \cdot X4 \cdot \neg X1 + \neg X16 \cdot X8 \cdot X2$$

#### 4.7 SAÍDA F

**Mapa de Karnaugh saída F**

		X4 X2							
		X1							
X8 X16		000	001	011	010	110	111	101	100
	00	0 <sup>0</sup>	1 <sup>1</sup>	1 <sup>3</sup>	0 <sup>2</sup>	1 <sup>6</sup>	1 <sup>7</sup>	1 <sup>5</sup>	1 <sup>4</sup>
	01	1 <sup>8</sup>	1 <sup>9</sup>	1 <sup>11</sup>	1 <sup>10</sup>	0 <sup>14</sup>	1 <sup>15</sup>	0 <sup>13</sup>	0 <sup>12</sup>
	11	0 <sup>24</sup>	0 <sup>25</sup>	0 <sup>27</sup>	0 <sup>26</sup>	0 <sup>28</sup>	0 <sup>29</sup>	0 <sup>31</sup>	0 <sup>30</sup>
	10	1 <sup>16</sup>	1 <sup>17</sup>	1 <sup>19</sup>	0 <sup>18</sup>	0 <sup>20</sup>	0 <sup>21</sup>	0 <sup>23</sup>	0 <sup>22</sup>

- Expressão Booleana Minimizada:

$$\neg X8 \cdot \neg X4 \cdot X1 + \neg X16 \cdot X2 \cdot X1 + \neg X16 \cdot \neg X8 \cdot X4 + \neg X16 \cdot X8 \cdot \neg X4 + X16 \cdot \neg X8 \cdot \neg X4 \cdot \neg X2$$

#### 4.8 SAÍDA G

**Mapa de Karnaugh saída G**

		X4 X2							
		X1							
X8 X16		000	001	011	010	110	111	101	100
	00	0 <sup>0</sup>	1 <sup>1</sup>	1 <sup>3</sup>	0 <sup>2</sup>	1 <sup>6</sup>	1 <sup>7</sup>	0 <sup>5</sup>	1 <sup>4</sup>
	01	1 <sup>8</sup>	0 <sup>9</sup>	1 <sup>11</sup>	0 <sup>10</sup>	1 <sup>14</sup>	1 <sup>15</sup>	1 <sup>13</sup>	1 <sup>12</sup>
	11	0 <sup>24</sup>	0 <sup>25</sup>	0 <sup>27</sup>	0 <sup>26</sup>	0 <sup>28</sup>	0 <sup>29</sup>	0 <sup>31</sup>	0 <sup>30</sup>
	10	1 <sup>16</sup>	1 <sup>17</sup>	1 <sup>19</sup>	1 <sup>18</sup>	0 <sup>20</sup>	0 <sup>21</sup>	0 <sup>23</sup>	0 <sup>22</sup>

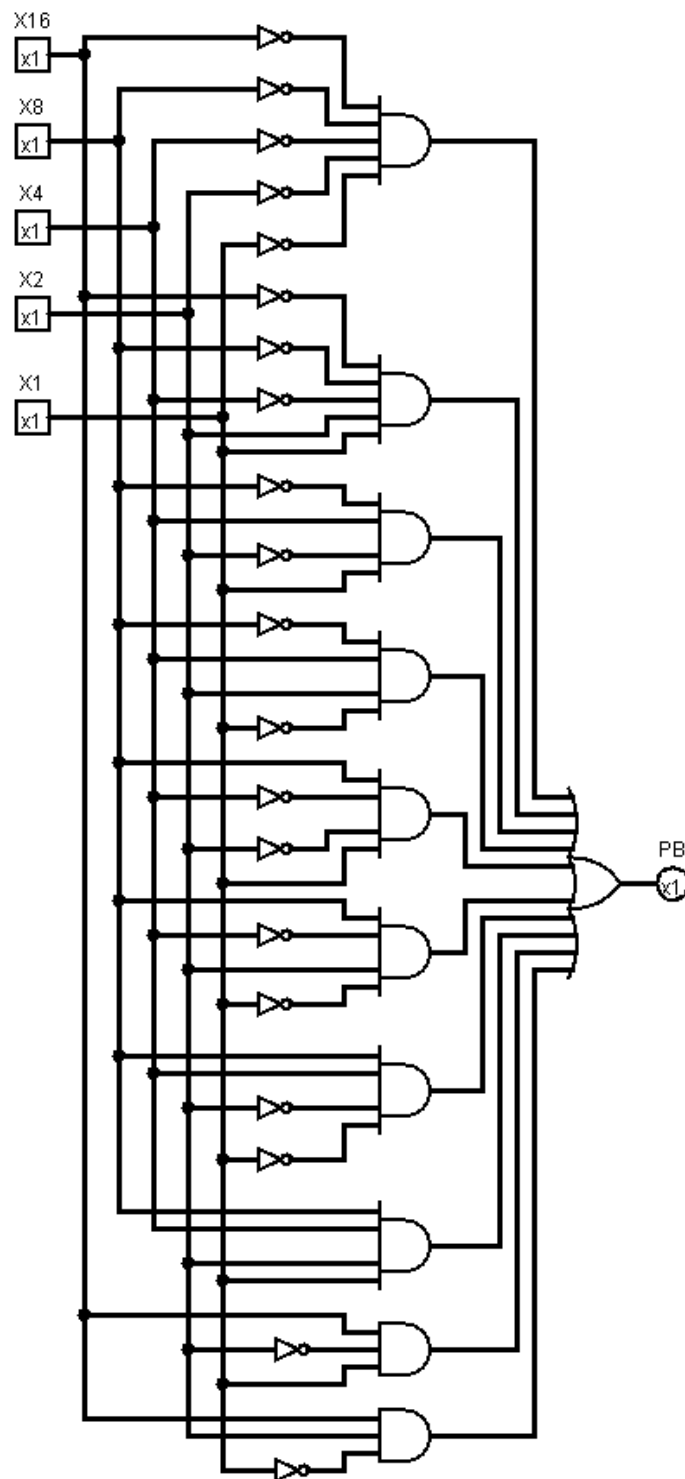
- Expressão Booleana Minimizada:

$$\neg X8 \cdot \neg X4 \cdot X1 + \neg X16 \cdot X2 \cdot X1 + \neg X16 \cdot X4 \cdot \neg X1 + \neg X16 \cdot X8 \cdot \neg X2 \cdot \neg X1 + \neg X16 \cdot X8 \cdot X4 + X16 \cdot \neg X8 \cdot \neg X4$$

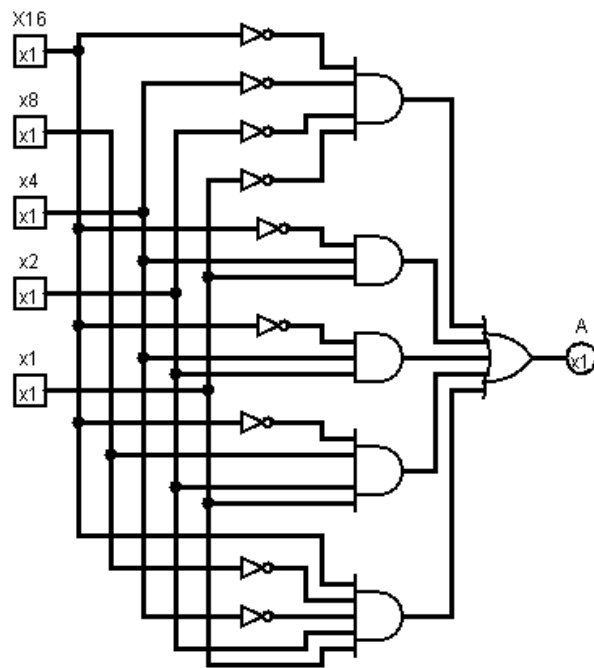
## 5. DIAGRAMAS DE PORTAS LÓGICAS

A construção dos diagramas de portas lógicas foi realizada utilizando o software Logisim[1] e as expressões booleanas minimizadas.

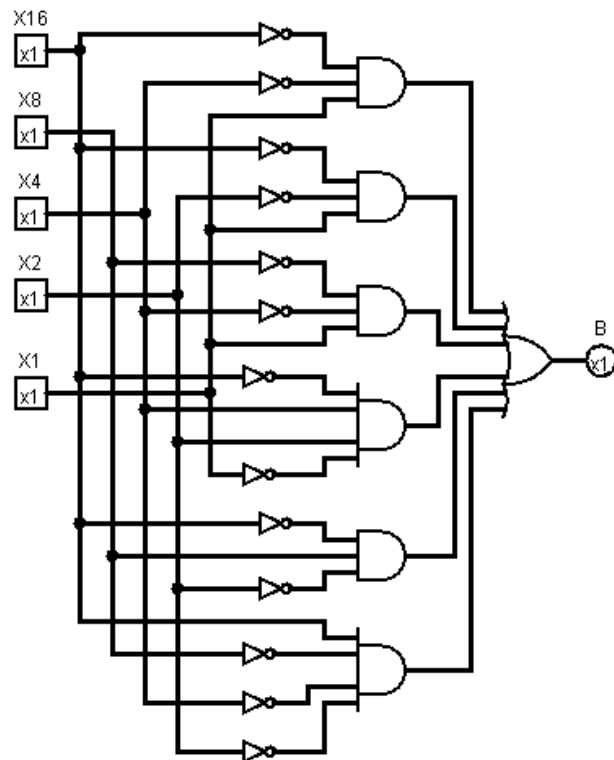
### 51 GERADOR DE BIT DE PARIDADE



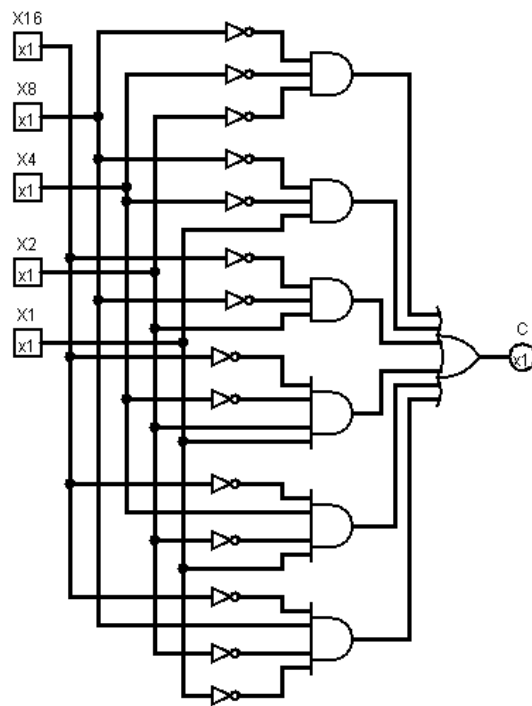
## 5.2 SAÍDA A



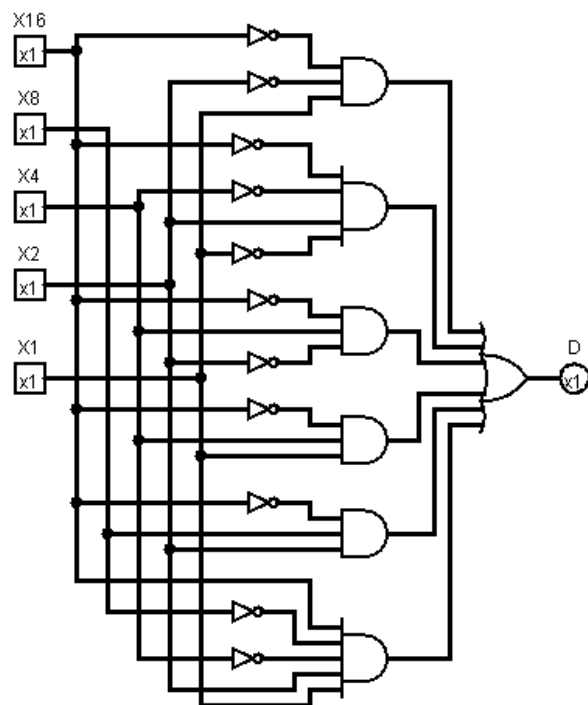
## 5.3 SAÍDA B



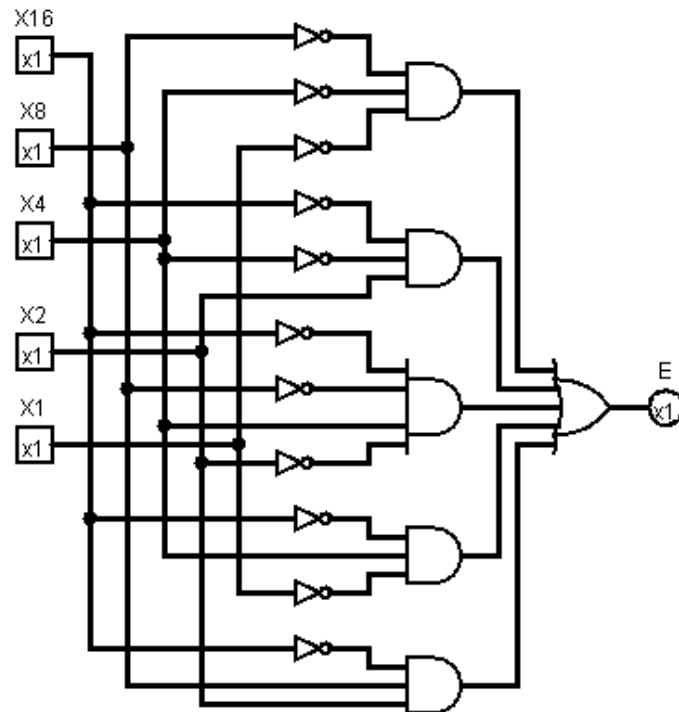
#### 5.4 SAÍDA C



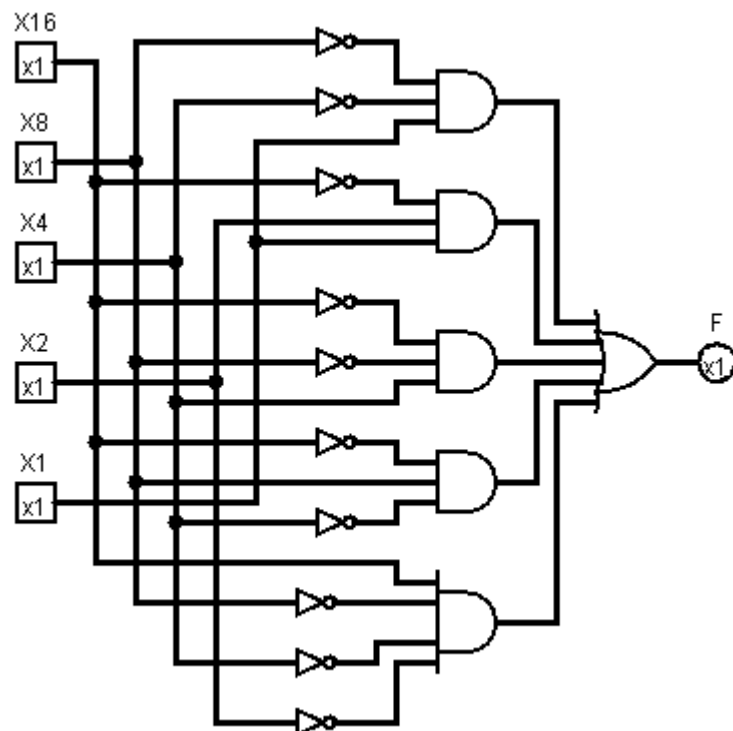
#### 5.5 SAÍDA D



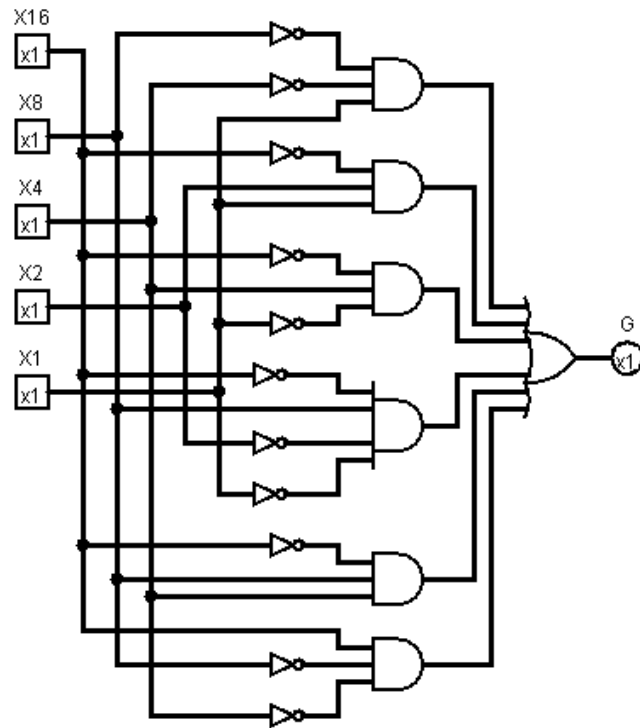
## 5.6 SAÍDA E



## 5.7 SAÍDA F



## 5.8 SAÍDA G



## **7. COMPILAÇÃO E TESTE**

Para executar a compilação e abrir o GtkWave, criamos um MakeFile com o caminho para os arquivos a serem compilados e executados, a fim de facilitar esse processo, portanto basta apenas digitar no terminal integrado o comando:

- `mingw32-make compile`



## 8.REFERÊNCIAS

[1] Canva. <<https://www.canva.com/>>. Último acesso em 24 de novembro de 2024.

[2] Logisim. <<https://sourceforge.net/projects/circuit>> . Último acesso em 25 de novembro de 2024.

[3] GitHub. <<https://github.com/lacerdamadu/CircCombinacional>>. Último acesso em 25 de novembro de 2024.