# UNIVERSIDADE FEDERAL DE LAVRAS
# GCC113 - CIRCUITOS DIGITAIS - 2020/2
# RELATÓRIO DE ATIVIDADE AVALIATIVA REO #02

Prof. Bruno Silva e Prof. Eric Araújo

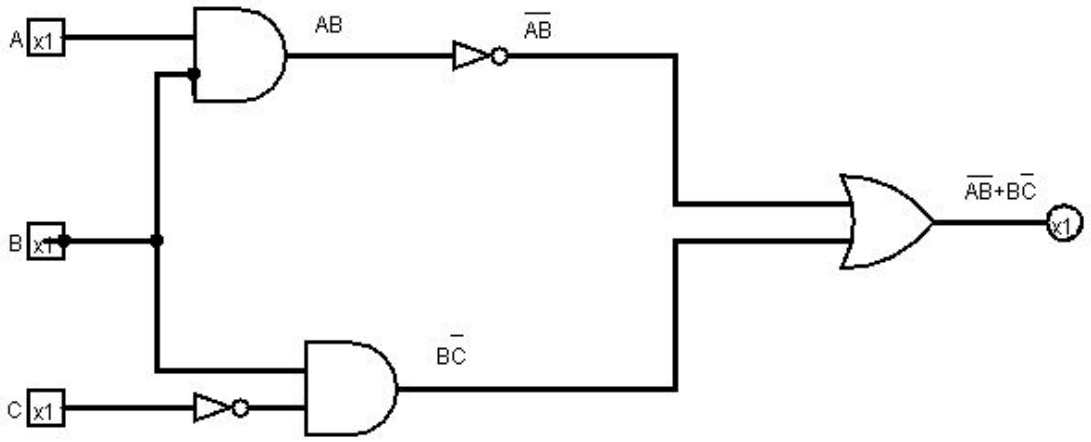

**Nome:** Layra Vilas Boas Ferreira

**Turma:** 10 A

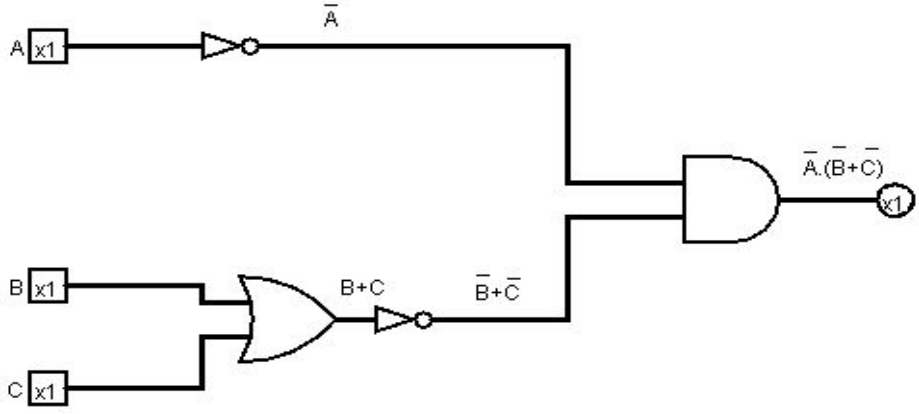
**Matrícula:** 202010142


1. **(AB)' + BC'**
   **(letra A)**

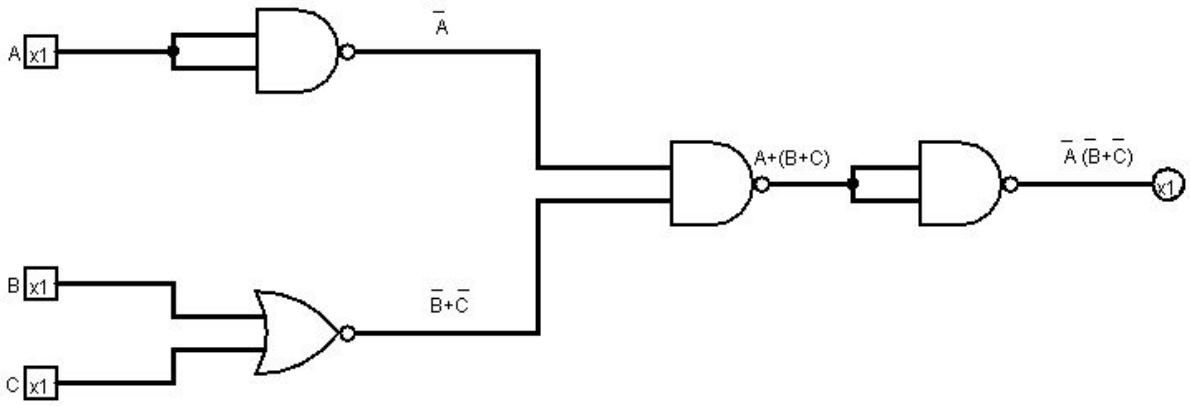
   **1.1 sem simplificação → custo = 12**




   **1.2 com simplificação:**
   **(AB)' + BC' → A' + B' + (BC') → A' + (B' + B).(B'C') → A' + (B'C') → (A'+B').(A'+C') → A'(B'+C') custo = 10 e → A'(BC)' também da custo = 10.**

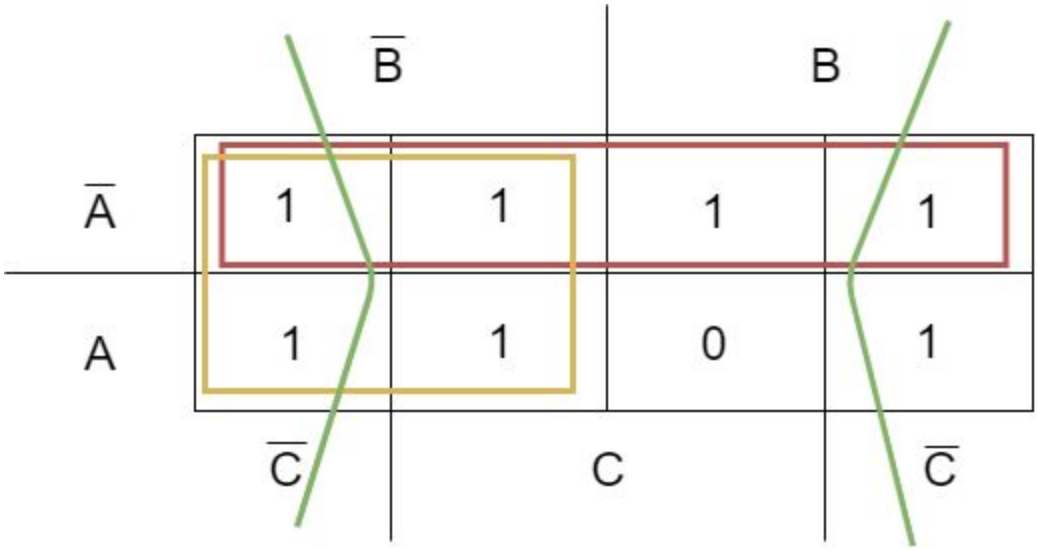


**1.3 simplificação (A'(BC)' e A'(B'+C') dao no mesmo custo) + universalidade das portas NAND e NOR → custo = 12**



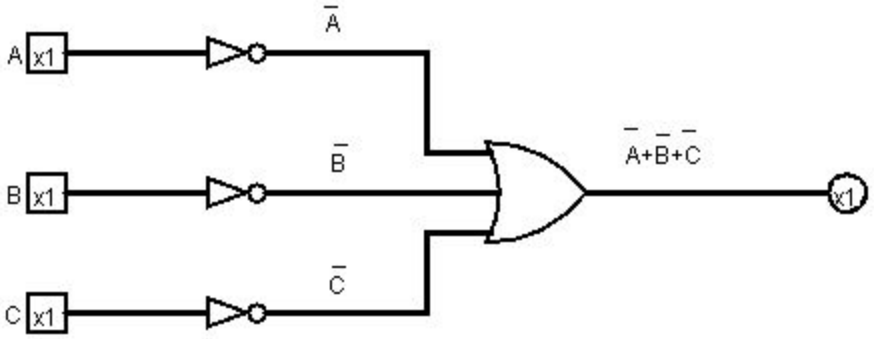

**1.4 mapa de karnaugh**

| A | B | C | (AB)' | BC' | X |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

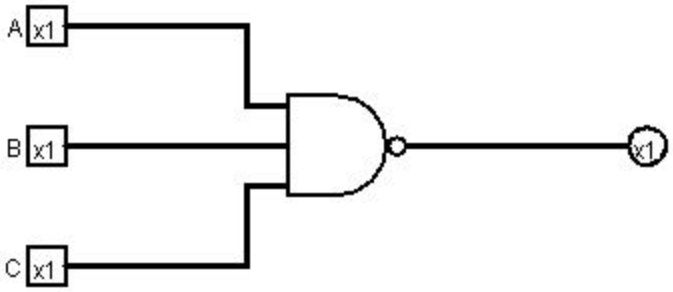


**X = A’ + B’ + C’**

**custo = 10**

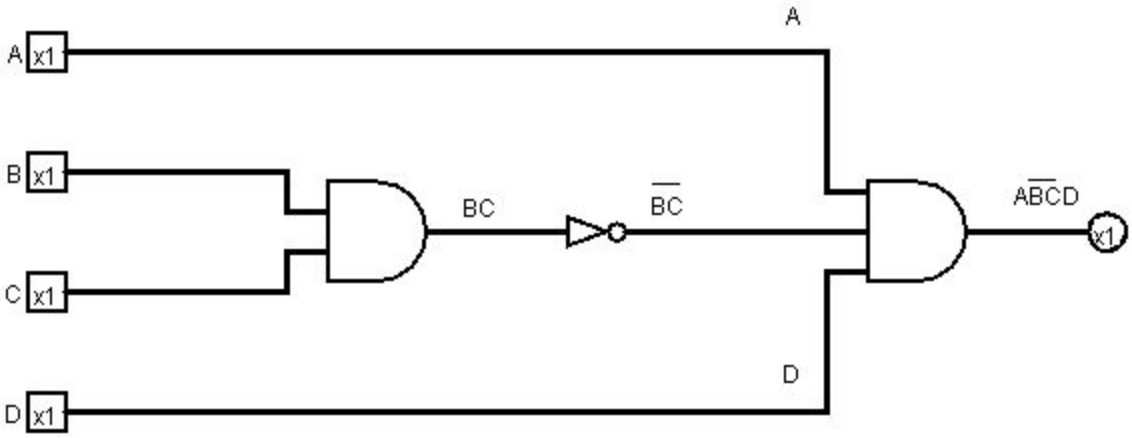



**1.5 simplificando expressão do mapa de karnaugh**
**A’ + B’ + C’ → (ABC)’**
**custo = 4**

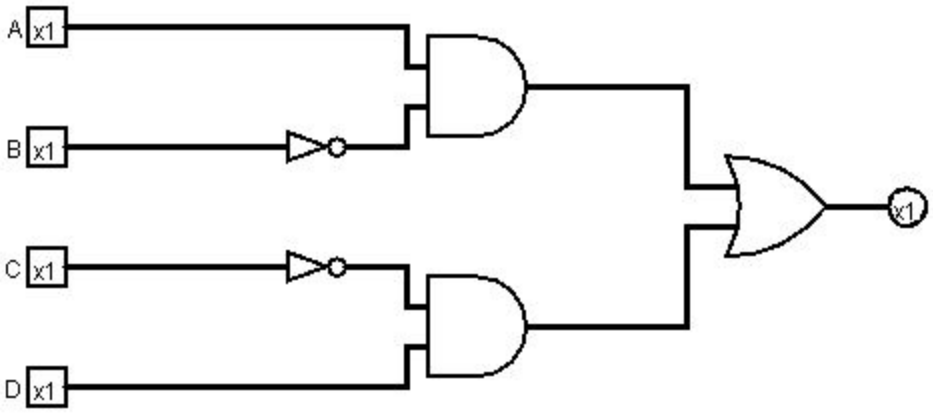



**2. A.(BC)’.D**
**(letra B)**

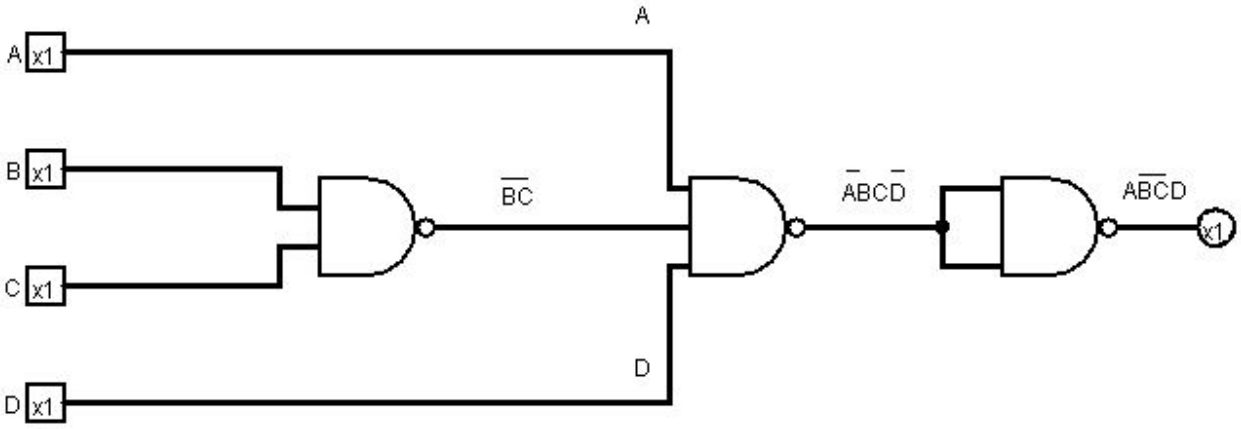
**2.1 sem simplificação → custo = 9**



**2.2 alterando a expressão**
**A.(BC)'.D → A.B' + C'.D**
**custo = 13**

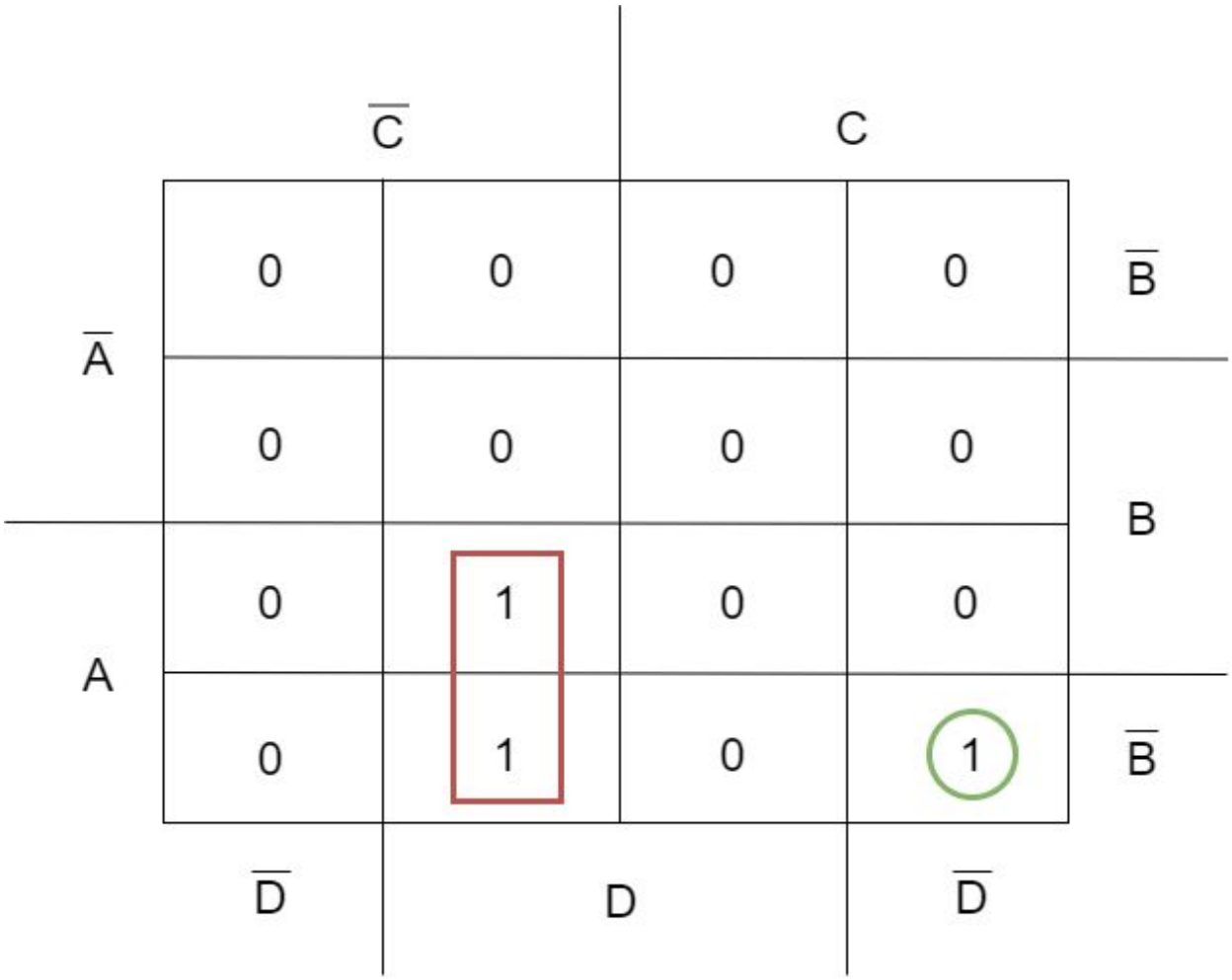



**2.3 A.(BC)'.D + universalidade das portas NAND e NOR → custo = 10**

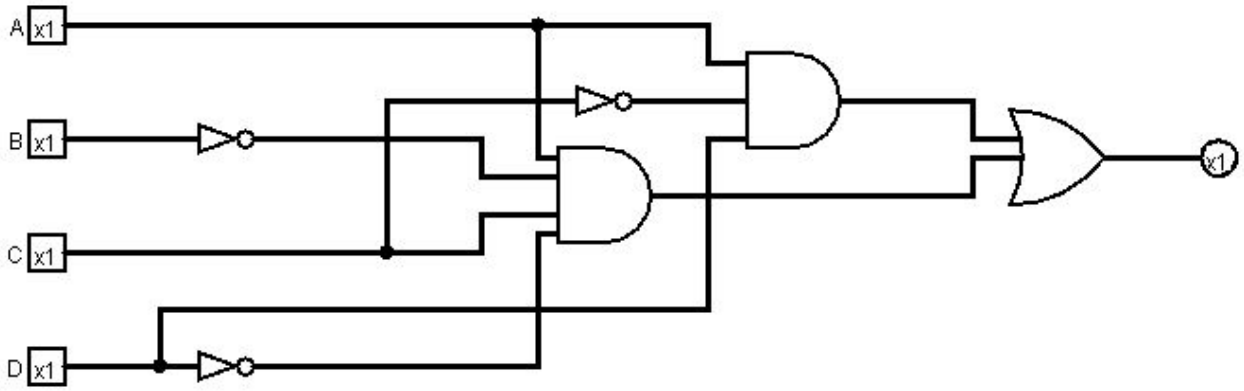


**2.4 mapa de karnaugh**

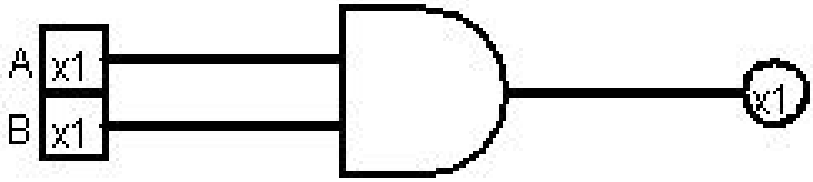
| A | B | C | D | (BC)' | X |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |



**X = AC'D + AB'CD' → custo = 18**




**2.5 simplificando expressão do mapa de karnaugh**
**AC'D + AB'CD' → A(C'D + B'CD') → A[(C'D)' + (B'CD')'] → A(C+D' + B + C' + D) → AB → custo = 3**

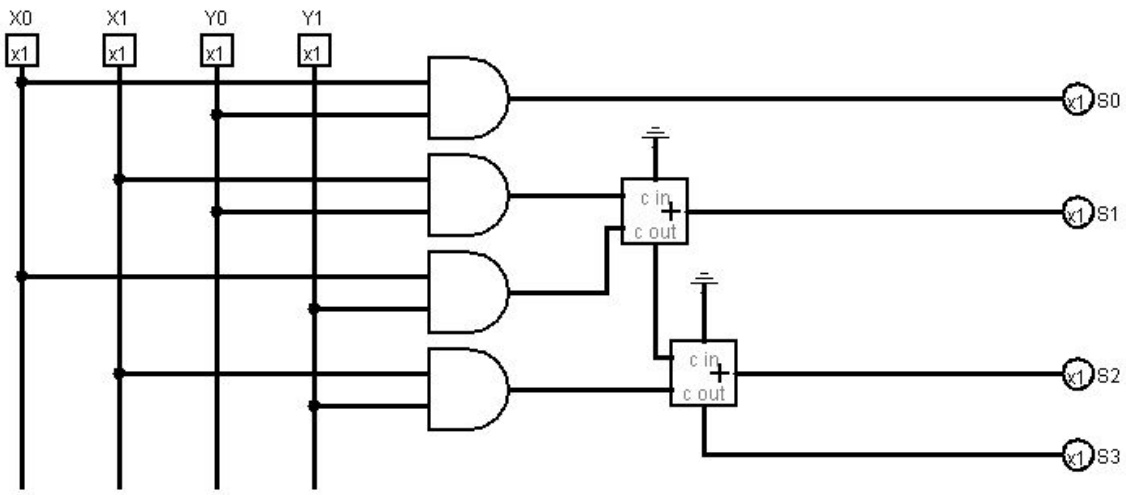



**3. (letra C) → utilizei o S para saída invés de Z (eu esqueci, desculpa)**

| | | | | |
|---|---|---|---|---|
| | | | Y1 Y0<br>X1 X0 | |
| | X1 Y1 | X0 Y1<br>X1 Y0 | X0 Y0 | |
| S3 | S2 | S1 | S0 | **(<--multiplicação)** |

**Carry de S1 + (soma com) S2 → carry → S3**

**Circuito :**


X0
X1
Y0
Y1
x1
x1
x1
x1
c in
c out
c in
c out
x1
S0
x1
S1
x1
S2
x1
S3