Nome: Eduardo Zirbell, Felipe Oliveira Mesquita de Sousa e Heloísa.

**Trabalho 1\_23\_1\_h**

Sistemas de Numeração e Lógica Booleana

1. Escreva o nome de um membro da equipe (com no mínimo 5 letras) em código ASCII. ( escrever em hexadecimal)

R: Nome 46-65-6C-69-70-65 (Felipe)

1. Converter 3 códigos de erro do Windows para decimal, explicando a conversão de hexadecimal para decimal;

1º Código

- 0x0000007B

0x168 + 0x167  + 0x166 + 0x165 + 0x164 + 0x163  + 0x162 + 7x161 + 11x160

Soma: 0 + 0 + 0 + 0 + 0 + 0 + 0 + 112 + 11

Valor em decimal = 123

2º Código

- 0x00000012

0x168 + 0x167  + 0x166 + 0x165 + 0x164 + 0x163  + 0x162 + 1x161 + 2x160

Soma: 0 + 0 + 0 + 0 + 0 + 0 + 0 + 16 + 2

Valor em decimal = 18

3º Código

- 0x00000029

0x168 + 0x167  + 0x166 + 0x165 + 0x164 + 0x163  + 0x162 + 2x161 + 9x160

Soma: 0 + 0 + 0 + 0 + 0 + 0 + 0 + 32 + 9

Valor em decimal = 41

1. Montar a tabela verdade da seguinte expressão booleana:

𝑆 = 𝐴 𝐵 𝐶 + 𝐴 𝐵 + 𝐴 𝐵 𝐶

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **AxBxC** | **xB** |  | **S** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

1. Dada a tabela verdade abaixo, montar a expressão booleana. Depois utilizar o Mapa de Karnaugh para simplificação e apresentar também a equação simplificada.

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **S** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

https://[www.profelectro.info/mapa-de-karnaugh-onlie-para-](http://www.profelectro.info/mapa-de-karnaugh-onlie-para-) simplificacao-de-funcoes-booleanas-a-partir-da-tabela-da-verdade/

R:.

NORMAL: S = () + () + () + ()

SIMPLIFICADA: F(ABC) = () + () + ()

1. Monte o circuito lógico utilizando portas lógicas da equação simplificada e simule-o utilizando logic.ly

