****

**UNIVERSIDADE FEDERAL DO CEARÁ – CAMPUS SOBRAL**

**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**DISCIPLINA: ELETRÔNICA DIGITAL**

**PROFESSORES: ROMULO NUNES DE CARVALHO ALMEIDA E DAVID NASCIMENTO COELHO**

**AVALIAÇÃO PARCIAL 01**

| **ALUNO** | **MATRÍCULA** |
| --- | --- |
| **Klayver Ximenes Carmo** | **427651** |

**Sobral – CE**

**2021**

**SUMÁRIO**

**Questões ............................................................................................................. Página**

**01-a) ......................................................................................................................... 03**

**01-b) ......................................................................................................................... 04**

**01-c) ......................................................................................................................... 05**

**01-d) ......................................................................................................................... 06**

**02-a) ......................................................................................................................... 07**

**02-b) ......................................................................................................................... 08**

**02-c) ......................................................................................................................... 09**

**03-a) ......................................................................................................................... 10**

**03-b) ......................................................................................................................... 11**

**03-c) ......................................................................................................................... 15**

**04 ….......................................................................................................................... 19**

**01 - a) A’BC + AB’C’ + A’B’C’+AB’C+ABC = BC + B’C’ + AB’**

A simplificação do lado esquerdo da expressão pode ser encontrada na figura 1.1.

| Figura 1.1 – Simplificação da expressão 1-a. |
| --- |
|  |
| Fonte: Autor |

Como visto, a simplificação foi feita utilizando algumas propriedades básicas da álgebra booleana, como distributiva, adição lógica e comutativa.

A simplificação encontrada no lado esquerdo foi igual ao lado direito da equação, confirmando assim a mesma.

**01 - b)** Utilizando as propriedades básicas da algebra booleana, é possível encontrar o resultado presente na figura 1.2.

| Figura 1.2 – Simplificação da expressão 01-b. |
| --- |
|  |
| Fonte: Autor |

A simplificação sem os barrados foi encontrada a partir das propriedades abaixo:

* (AB)’ = A’ + B’
* (A + B)’’ = A + B
* A + A’ = 1
* 1 + A + B = 1

**01 - c)** Com os dados da questão e algumas propriedades básicas de simplificação da álgebra booleana abordadas no item anterior, foi possível encontrar o resultado apresentado na figura 1.3.

| Figura 1.3 – Simplificação da equação 01-c. |
| --- |
|  |
| Fonte: Autor |

Sendo assim, uma afirmação verdadeira, onde são iguais ambos os lados da equação.

A simplificação utilizada foi encontrada a partir das propriedades abaixo:

* (AB)’ = A’ + B’
* (A + B)’’ = A + B
* A + A’ = 1
* 1 + A + B = 1
* AA’ = 0

**01 - d)** Com os dados da questão e algumas propriedades básicas de simplificação da álgebra booleana, foi possível encontrar o resultado apresentado na figura 1.4.

| Figura 1.4 – Simplificação da equação 01-d. |
| --- |
|  |
| Fonte: Autor |

A partir do resultado encontrado, foi possível perceber que a afirmação não é verdadeira, já que o lado esquerdo da equação resultou em uma expressão e o lado direito é uma constante.

**02 - a)** A associação da expressão ao seu mapa k foi apresentada na figura 2.1 de acordo com as cores representadas em cada termo da expressão.

| Figura 2.1 – Representação no mapa k da expressão da questão. |
| --- |
|  |
| Fonte: Autor |

**02 - b)** Não é mínima, podendo assim ser otimizada utilizando o mapa de karnaugh resultando em um termo a menos que a expressão repassada na questão. Como apresenta a figura 2.2.

| Figura 2.2 – Mínima da expressão da questão. |
| --- |
|  |
| Fonte: Autor |

Como no item anterior, os termos foram organizados de acordo com a cor.

**02 - c)** O padrão da porta lógica XNOR num mapa k com 6 variáveis é apresentado na figura 2.3.

| Figura 2.3 – Padrão XNOR no mapa k com 6 variáveis. |
| --- |
|  |
| Fonte: Autor |

Como é possível perceber, no padrão XNOR a saída lógica 1 é presente onde a quantidade de bits 0 é par.

**03 - a)** Para montar o circuito utilizando a decomposição de Shannon, primeiro temos que encontrar a expressão do mesmo. A figura 3.1 apresenta o mapa de karnaugh do Y0, juntamente com a expressão encontrada.

| Figura 3.1.1 – Mapa de karnaugh do Y0. |
| --- |
|  |
| Fonte: Autor |

Para o uso do MUX 2:1, adotamos A sendo a porta seletora, e seus casos em que é 0 e 1, resultando como entrada de dados B e B’ como consta na figura 3.1.2 o circuito feito utilizando o MUX 2:1.

| Figura 3.1.2 – Circuito da expressão de Y0. |
| --- |
|  |
| Fonte: Autor |

**03 - b)** O menor MUX implementável para o Y1 é o de 2:1. Na figura 3.2.1 é apresentado a tabela verdade do problema. É possível utilizar o don't care, visto que no último caso não é especificado, porém, não ajudaria na construção da expressão, sendo assim utilizado nível lógico 0 para o mesmo.

| Figura 3.2.1 – Mapa de karnaugh de Y1. |
| --- |
|  |
| Fonte: Autor |

Na figura 3.2.2 é apresentado o mapa de karnaugh com a obtenção da expressão do mesmo.

| Figura 3.2.2 – Mapa de karnaugh de Y1. |
| --- |
|  |
| Fonte: Autor |

Com a expressão encontrada (**AB’S’ + A’S**), é possível fazer manipulações para otimização da mesma, como o isolamento das entradas A e B, resultando em apenas uma seletora, que é a **S**. Obtendo assim uma expressão em uma das entradas de dado para um MUX 2:1, como apresentado na figura 3.2.3.

| Figura 3.2.3 – MUX 2:1 de Y1. |
| --- |
|  |
| Fonte: Autor |

Para fins de apresentação, também implementei o circuito num MUX 4:1, como apresenta a figura 3.2.4.

| Figura 3.2.4 – MUX 4:1 de Y1. |
| --- |
|  |
| Fonte: Autor |

Como visto, no MUX 4:1 as entradas são constantes, fazendo o isolamento apenas da seletora **A**, restando as seletoras **B** e **S**, onde a variável **A** deve aparecer nas entradas de dados.

Para fins de conhecimento e aprendizado, também implementei os MUX’s 8:1 e simulações do 4:1 e 2:1 no proteus. Utilizei MUX 8:1 pois não achei os menores (4:1 e 2:1) utilizando o método de aterramento das seletoras que não são utilizadas.

As figuras 3.2.5, 3.2.6 e 3.2.7 apresentam o MUX 8:1, simulação de um 4:1 com um 8:1 e a simulação de um 2:1 com um MUX 8:1, respectivamente. Todas utilizam a saída Y1.

| Figura 3.2.5 – MUX 8:1 de Y1. |
| --- |
|  |
| Fonte: Autor |

| Figura 3.2.6 – MUX 8:1 simulando 4:1. |
| --- |
|  |
| Fonte: Autor |

| Figura 3.2.7 – MUX 8:1 simulando 2:1. |
| --- |
|  |
| Fonte: Autor |

Os testes dos mesmos podem ser feitos pelo arquivo do projeto enviado.

**03 - c)** De acordo com as saídas dadas na questão de Y2, a figura 3.3.1 apresenta o MUX 64:1 de forma completa, onde as saídas do problema podem ser conectadas nas entradas do MUX 64:1. Sendo conectados no +Vcc as correspondentes ao nível lógico 1 e no terra os níveis lógicos 0.

| Figura 3.3.1 – MUX 64:1 de Y2. |
| --- |
|  |
| Fonte: Autor |

As figuras 3.3.2 e 3.3.3 apresentam de forma aproximada as entradas mais importantes do MUX 64:1 e das seletoras, respectivamente.

| Figura 3.3.2 – Inputs do MUX 64:1. |
| --- |
|  |
| Fonte: Autor |

| Figura 3.3.3 – Seletoras do MUX 64:1. |
| --- |
|  |
| Fonte: Autor |

Como as entradas de dados correspondentes às seletoras não são utilizadas, o circuito é otimizado colocando as mesmas conectadas no terra.

Para a construção do MUX a partir de MUX’s 4:1, como o multiplexador de 64:1 possui 64 entradas, a construção do mesmo é dividida em 16 multiplexadores 4:1 na primeira camada, onde o total de portas de entradas seria 64. Suas entradas seletoras seriam todas conectadas, verificando assim a combinação escolhida para o bit mais significativo.

Na segunda camada seriam 4 multiplexadores 4:1, tendo como entradas as saídas da primeira camada, e assim, a última camada, com um multiplexador recebendo as entradas da segunda camada.

Um exemplo da estrutura das camadas é apresentado na figura 3.3.4.

| Figura 3.3.4 – Implementação MUX 64:1 com MUX’s 4:1. |
| --- |
|  |
| Fonte: Autor |

**04 -** Matrícula 427651. A tabela verdade é montada seguindo os números da matrícula na figura 4.1. Para os valores de 10 a 15, casos extras no problema, foi adotado a representação de não pertencer à matrícula.

| Figura 4.1 – Tabela verdade da matrícula. |
| --- |
|  |
| Fonte: Autor |

Como é possível perceber pela tabela, as saídas E, F e G estão sempre ligadas, estado lógico 1. As saídas A e D são iguais, acionando nos mesmos instantes, igualmente com as saídas B e C, sendo inversas de A e D, ou seja, achando uma expressão, a outra é apenas invertida.

A partir da tabela verdade, foi feito o mapa de karnaugh para a obtenção da expressão do circuito, como consta na figura 4.2.

| Figura 4.2 – Mapa k da matrícula. |
| --- |
|  |
| Fonte: Autor |

A representação a ser formada no display de 7 segmentos quando um número pertence à matrícula está na figura 4.3.

| Figura 4.3 – Representação de um número contido na matrícula. |
| --- |
|  |
| Fonte: Autor |

A representação a ser formada no display de 7 segmentos quando um número não pertence à matrícula está na figura 4.4.

| Figura 4.4 – Representação de um número não contido na matrícula. |
| --- |
|  |
| Fonte: Autor |

Como o circuito deve ser montado apenas com portas NAND, é possível encontrar uma universalização da mesma, como consta nas figuras 4.5 (porta NOT), 4.6 (porta AND) e 4.7 (porta OR).

| Figura 4.5 – Representação da porta NOT com NAND. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.6 – Representação da porta AND com NAND. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.7 – Representação da porta OR com NAND. |
| --- |
|  |
| Fonte: Autor |

Com a expressão encontrada e já sabendo da universalização, é possível montar o circuito no proteus. O circuito resultante é apresentado na figura 4.8.

| Figura 4.8 – Circuito final da matrícula. |
| --- |
|  |
| Fonte: Autor |

Nas figuras 4.9 a 4.14 são apresentados os exemplos dos números de 0 a 5.

| Figura 4.9 – Exemplo de circuito com número 0. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.10 – Exemplo de circuito com número 1. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.11 – Exemplo de circuito com número 2. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.12 – Exemplo de circuito com número 3. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.13 – Exemplo de circuito com número 4. |
| --- |
|  |
| Fonte: Autor |

| Figura 4.14 – Exemplo de circuito com número 5. |
| --- |
|  |
| Fonte: Autor |