**Identificação:**

**Projeto Nº. 1: Elevador­**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pt** | **Ps** | **T** | **S** | **MS** | **MD** |
| 0 | 0 | 0 | 0 | X | X |
| 0 | 0 | 0 | 1 | 1 | 0  **MS** |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0  **MD** |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | X | X |
| 1 | 1 | 0 | 1 | X | X |
| 1 | 1 | 1 | 0 | X | X |
| 1 | 1 | 1 | 1 | X | X |

**MS**= **Pt.Ps.T.S + Pt.Ps.T.S + Pt.Ps.T.S + Pt.Ps.T.S**

**MD**= **Pt.Ps.T.S + Pt.Ps.T.S + Pt.Ps.T.S + Pt.Ps.T.S**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **00** | | **01** | **11** | | **10** |
| **00** | **X** |  | | **X** |  | |
| **01** | **1** |  | | **X** | **1** | |
| **11** | **1** |  | | **X** | **1** | |
| **10** |  |  | | **X** |  | |

**Pt.Ps**

**T.S**

**MS**= **Ps.S**

O estudo da tabela verdadepode ser efetuado analisando-se as condições apresentadas pelas variáveis de entrada. No primeiro caso, onde todas as variáveis (**Pt**, **Ps**, **T, S**) tem o valor lógico 0, o que nos mostra que o elevador não está em nenhum andar e não foi chamado, portanto, também não está em movimento. Entretanto, para que isso seja possível a porta do elevador deveria estar aberta, porém, como vimos, o elevador não está em nenhum andar, o que o torna a abertura da porta impossível, e nos leva a atribuir saídas irrelevantes. Nos outros casos onde tivemos saídas irrelevantes, deu-se, pois, que de acordo com a tabela verdade, o elevador estaria em mais de um andar simultaneamente, o que é também é impossível.

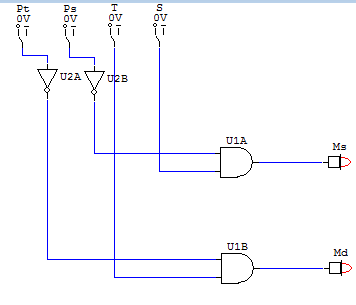
**T.S**

**Pt.Ps**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **00** | | **01** | **11** | | **10** |
| **00** |  |  | | **X** |  | |
| **01** |  |  | | **X** |  | |
| **11** | **1** | **1** | | **X** |  | |
| **10** | **1** | **1** | | **X** |  | |

**MD =** **Pt.T**

A partir da tabela obtida, simplificamos as saídas utilizando mapas de Karnaugh para 4 variáveis, que nos resultou as seguintes expressões: **MS** = **Ps.S** e **MD =** **Pt.T.** Finalmente, a partir das expressões reduzidas pode-se criar o circuito lógico do elevador em questão, utilizando 4 Logic Switchs ( 1 para cada variável **Pt, Ps, T, S**, sendo que **Pt, Pt,** foram conectados a inversores) a partir daí, **Pt** e **T** foram conectados a uma porta AND de duas entradas com a saída (Logic Display) **MD. Ps** e **S** foram conectados a uma porta AND de duas entradas com a Saída (Logic Display) **MS.**



**Projeto Nº 2: Semáforo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **L-O** | **N-S** |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0  **L-O** |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0  **N-S** |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1  **C.D** |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |

**L-O = A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D**

**N-S = A.B.C.D + A.B.C.D + A.B.C.D**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **00** | | **01** | **11** | | **10** |
| **00** | **1** |  | |  |  | |
| **01** | **1** | **1** | | **1** | **1** | |
| **11** | **1** | **1** | | **1** | **1** | |
| **10** | **1** | **1** | | **1** | **1** | |

**C.D**

**A.B**

**L-O = A.B + C + D**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **00** | | **01** | **11** | | **10** |
| **00** |  | **1** | | **1** | **1** | |
| **01** |  |  | |  |  | |
| **11** |  |  | |  |  | |
| **10** |  |  | |  |  | |

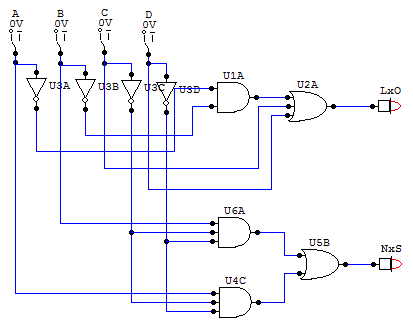
**A.B**

**N-S = B.C.D + A.C.D**

Começamos a montar a tabela verdade a partir que consideramos as entrada A, B (que juntas representam os sensores posicionados na via de acesso) e C, D (que juntas representam os sensores posicionados na rodovia); e como portas de saída, consideramos o as direções Leste-Oeste (L-O) e Norte-Sul (N-S).

A saída L-O recebeu como nível lógico 1 em 13 das 16 situações, pois, sempre quando passasse algum carro, podendo ser em C (que representa o carro vindo da direção Leste), OU em D (que representa o carro vindo da direção Oeste), os sensores ficariam com nível lógico ALTO (1), uma vez que as pistas C e D representam a rodovia, e os carros que passam pela rodovia têm a preferência. Por sua vez, a saída N-S recebeu como nível lógico 1 em 3 das 16 situações, pois, como as pistas C e D representam a rodovia, e assim a lógica OU deveria ser aplicada, só ficaria nível lógico ALTO (1) para a saída N-S apenas quando não houvesse carro na pista C e D, tendo apenas 3 situações assim constatadas pela tabela verdade.

A partir da tabela obtida, simplificamos as saídas utilizando mapas de Karnaugh para 4 variáveis, que nos resultou as seguintes expressões: **L-O =** **A.B + C + D**e **N-S** = **B.C.D + A.C.D** . A partir daí, pudemos desenvolver o circuito abaixo:

****

**Projeto Nº. 3: ULA**

**Definição:** A *ULA* **(**Unidade Lógica da Aritmética) é a seção do computador que executa todas as operações lógicas e aritméticas. As funções aritméticas requeridas, como adição e subtração e todas as operações lógicas, tais como: AND, NAND, OR, NOR, XOR e XNOR.

O conceito da ULA foi proposto pelo matemático John Von Neumann em 1945 quando escreveu um relatório sobre os fundamentos para um novo computador chamado EDVAC, que usava o sistema binário para suas operações. O circuito apresentado a seguir é uma ULA de 1 bit, que contém as seguintes operações:

Lógicas: AND, NAND, OR, XOR, XNOR

Aritméticas: Adição binária completa, Subtração binária completa

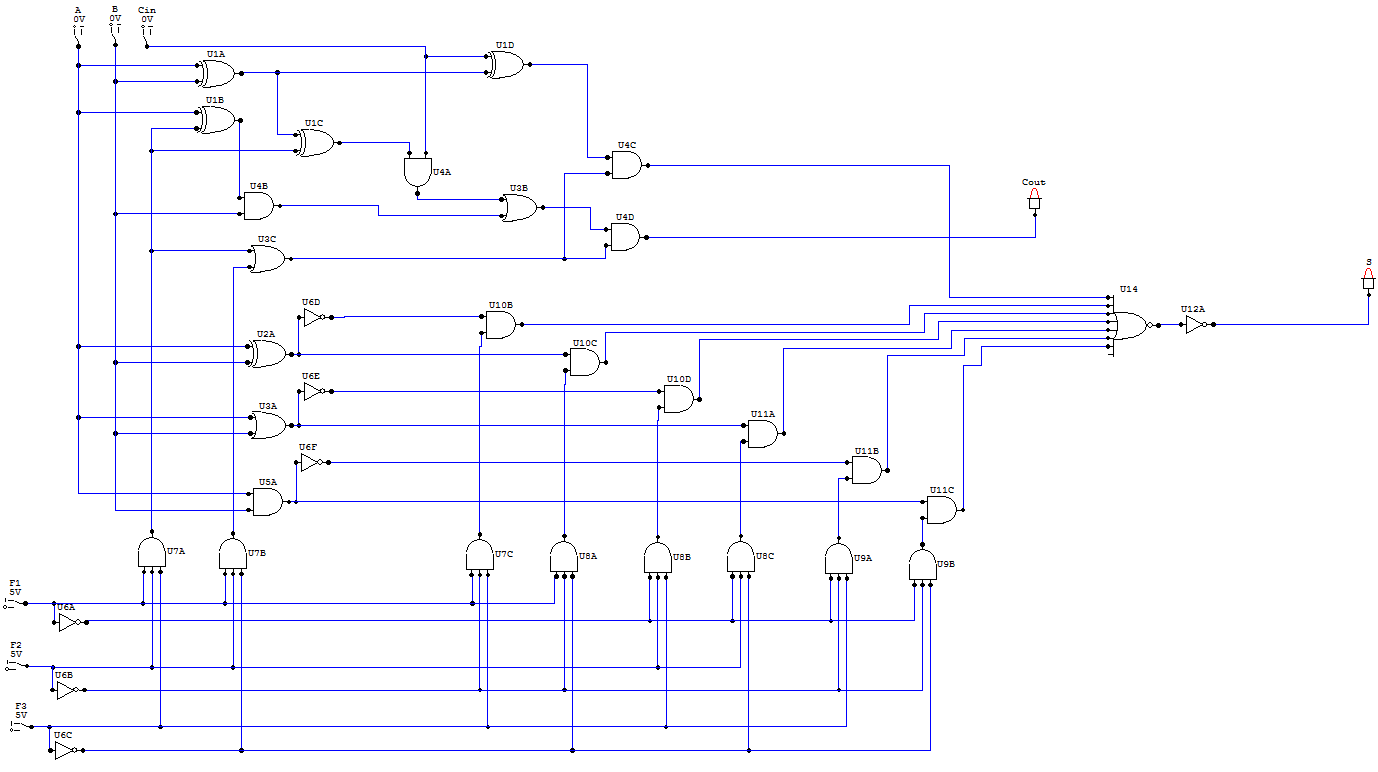
As entradas dessa ULA são o bit identificado por A, B e Cin e a saída escolhida pelo decodificador com suas entradas F1, F2 e F3 que geram, para cada combinação, a saída para a operação desejada. A tabela verdade do decodificador pode ser visto abaixo:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **F3** | **F2** | **F1** | **AND** | **XOR** | **OR** | **SOMA** | **NAND** | **XNOR** | **NOR** | **SUB** |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Dessa forma, as expressões booleanas serão:

* **AND**: F3.F2.F1
* **XOR:** F3.F2.F1
* **OR:** F3.F2.F1
* **SOMADOR:** F3.F2.F1
* **NAND:** F3.F2.F1
* **XNOR:** F3.F2.F1
* **NOR:** F3.F2.F1
* **SUBTRATOR:** F3.F2.F1

Para desenvolvermos a simulação da ULA conforme o circuito abaixo foi necessário o agrupamento de três principais circuitos: Circuito Decodificador, Circuito das Portas Lógicas e Circuito das Operações aritméticas, todos gerenciados por um circuito de portas de controle.



Portas de Controle

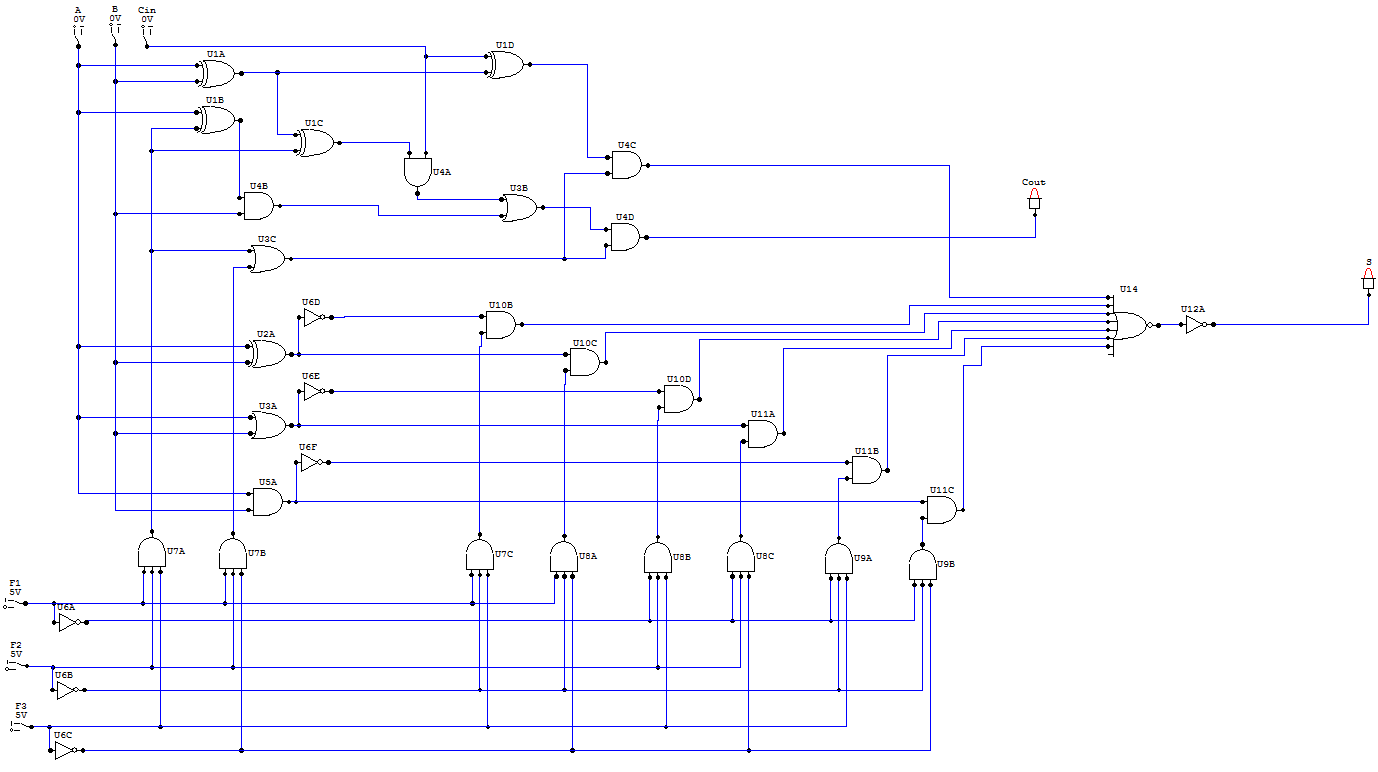
Operações Aritméticas

Portas Lógicas

Decodificador

O circuito decodificador foi montado com portas AND dispostas, em relação às entradas, de maneira que para cada combinação de entradas, somente uma única porta seja ativada, liberando apenas uma saída para os demais circuitos. Dessa maneira, conseguimos escolher qual função ativaremos na saída final, passando antes pelas portas de controle.

As portas de controle (portas AND e OR) foram organizadas de uma forma que cada porta está ligada com uma saída de uma única porta pertencente ao decodificador. Assim sendo, quando uma saída do decodificador é igual a 1, ao se juntar com a porta AND de controle, uma porta lógica ou uma operação aritmética é selecionada e a saída da porta de controle será a mesma que a porta ativada, já que 1.X = X. Finalmente, como todas as outras saídas são 0 pois a saída 1 do decodificador é única, a porta NOR de 8 entradas emite a saída contrária à porta selecionada e o inversor inverte a saída corrigindo-a e, por fim o resultado é mostrado no Logic Display.

OBS: Foi usada uma porta NOR de 8 entradas porque não foi possível localizar uma porta OR de 7 entradas, que seria mais adequada.

**Referências**

* HARRIS, DAVID MONEY; HARRIS, SARAH L. **DIGITAL DESIGN AND COMPUTER ARCHITECTURE FROM GATES TO PROCESSORS**. EDITORA ELSEVIER ISE, 1ª EDIÇÃO, 2007.
* CAPUANO, FRANCISCO GABRIEL E IDOETA, IVAN V. **ELEMENTOS DE ELETRÔNICA DIGITAL**. EDITORA ÉRICA, 40ª EDIÇÃO, 2006.
* UYEMURA, JOHN PAUL. **SISTEMAS DIGITAIS: UMA ABORDAGEM INTEGRADA.** EDITORA THOMSON PIONEIRA, 1ª EDIÇÃO, 2002.
* CAPRON, H. L; JOHNSON, J.A. **INTRODUÇÃO À INFORMÁTICA.** 8. ED. SÃO PAULO: PEARSON, 2004.