**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Identificación y análisis de circuitos integrados y compuertas lógicas |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | Determinar el funcionamiento y las aplicaciones de los circuitos electrónicos. | RESULTADOS DE APRENDIZAJE | Sintetizar circuitos lógicos a partir de una descripción inicial que utiliza un lenguaje convencional. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 04 |
| NOMBRE DEL COMPONENTE FORMATIVO | Lógica combinatoria |
| BREVE DESCRIPCIÓN | La lógica combinatoria utiliza compuertas lógicas como *OR, AND y NOT* para realizar funciones específicas. Representa operaciones mediante esquemas lógicos y tablas de verdad. Su simplificación optimiza diseños usando propiedades de Boole, teoremas de De Morgan y mapas de Karnaugh, reduciendo costos y espacio. Se aplica en circuitos digitales para procesos como control y supervisión. |
| PALABRAS CLAVE | Lógica combinatoria, operaciones, compuertas, simplificación, circuitos. |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:** 
   * + 1. **Lógica combinatoria**
2. **INTRODUCCIÓN**

La lógica combinatoria es un componente fundamental en la electrónica digital, permitiendo integrar diferentes compuertas lógicas para resolver problemas específicos y realizar funciones complejas. Este enfoque se basa en operaciones como *OR, AND y NOT*, que constituyen la base para procesar y manipular señales binarias en circuitos digitales.

|  |  |
| --- | --- |
| Technician testing electronic circuit board with probe  1064031580 | Una de las herramientas principales en la lógica combinatoria son los esquemas lógicos y las tablas de verdad, los cuales permiten visualizar y analizar el comportamiento de las funciones lógicas en función de todas las posibles combinaciones de entrada. Estas representaciones facilitan el diseño de circuitos eficientes y precisos en aplicaciones prácticas. |

Además, el proceso de simplificación de funciones lógicas, mediante el uso de propiedades de Boole, teoremas de De Morgan y mapas de Karnaugh, optimiza los diseños reduciendo la cantidad de compuertas necesarias. Esto no solo ahorra espacio y costos, sino que también mejora el rendimiento general de los circuitos, destacando la importancia de este concepto en el ámbito tecnológico.

1. **DESARROLLO DE CONTENIDOS:**
   * + 1. **Lógica combinatoria**

La lógica combinatoria implica la integración de diferentes compuertas en un único circuito o esquema lógico.

**Ejercicio práctico 1**

Realizar el esquema lógico y la tabla de verdad a partir de la siguiente función lógica: (A + B) · (B + C).

Para construir el esquema lógico basado en la función dada, primero se identifican las operaciones involucradas para que se cumpla la función, de la siguiente forma:



**3 operaciones: (+, \*, +).**

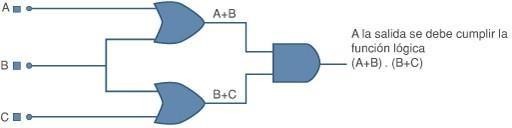
A continuación, se identifican las compuertas que corresponden a cada operación:

**Tabla 1**. Relación entre funciones lógicas, operaciones y compuertas

|  |  |  |
| --- | --- | --- |
| **Función o salida** | **Operación** | **Compuerta** |
| A + B | + | OR |
| B + C | + | OR |
| (A + B) · (B + C) | · | AND |

De esta forma, el esquema lógico quedaría diseñado considerando las compuertas correspondientes a cada operación.

**Figura 1.** Diagrama de compuertas lógicas



**Esquema lógico o plano**

Antes de construir la tabla de verdad, es fundamental tener en cuenta:

|  |  |
| --- | --- |
|  | **Tabla de verdad.** Es una representación gráfica de todos los valores que puede asumir la función lógica para cada una de las posibles combinaciones de las variables de entrada. |

El número de combinaciones posibles se calcula como 2n, siendo n el número de variables.

|  |
| --- |
| **Ejemplo**.  Para la función lógica F=(A+B) \* C  n=3 (ya que las variables son A, B y C).  23=8 (número de combinaciones posibles y filas que debe tener la tabla). |

La tabla de verdad y los niveles lógicos que cumplen dicha función en la salida son:

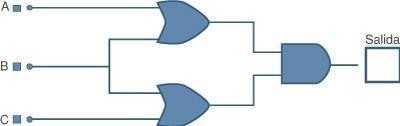
**Figura 2**. Tabla de verdad.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variables o entradas** | | | **Salidas parciales** | | **Función lógica** |
| **A** | **B** | **C** | **A + B** | **B + C** | **(A + B) \* (B + C)** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

**Combinaciones**

Para comprobar la tabla realice la simulación:

**Figura 3.** Diagrama lógico con compuertas OR y AND



**Simplificación de funciones lógicas**

|  |  |
| --- | --- |
| Al diseñar esquemas lógicos para cumplir funciones o procesos específicos, puede generarse inicialmente una función extensa que requiera un gran número de compuertas lógicas. Sin embargo, estas funciones pueden simplificarse mediante las propiedades de Boole, los teoremas de Morgan y los mapas de Karnaugh. | NOR Gate vector icon illustration of School iconset.  980565866 |

Como ejercicio práctico, se continuará trabajando con la función lógica del ejercicio 1.

**Ejercicio práctico 2**

Simplificar la función lógica: **(A+B) \* (B+C)**

Antes de simplificar la función, es importante recordar las propiedades de Boole y los teoremas de Morgan:

## **Tabla 1.** Propiedades o Reglas del Álgebra de Boole

|  |  |
| --- | --- |
| **Propiedad o Regla** | **Ejemplo** |
| Simplificación de negación | = A |
| Simplificación de igualdad | A + A = A A · A = A |
| Inversa | A + = 1 A · = 0 |
| Identidad | A + 0 = A A · 1 = A A + 1 = 1 A · 0 = 0 |
| Conmutativa | A + B = B + A A · B = B · A |
| Asociativa | A + (B + C) = (A + B) + C A · (B · C) = (A · B) · C |
| Distributiva | A · (B + C) = (A · B) + (A · C) (A + B) · C = (A · C) + (B · C) |
| Más reglas de simplificación o leyes de absorción | A + A · B = A A · (A + B) = A |
| Teorema de Morgan | = A′ + B′  = A′ · B′ |

El proceso de simplificación es:

**Validación de la simplificación**

Para validar que la función simplificada cumple con la misma salida que la función original, se debe:

|  |  |
| --- | --- |
| High-resolution image of a blue microchip central processing unit, illustrating technology and computing hardware concepts.  **866478098** |  |

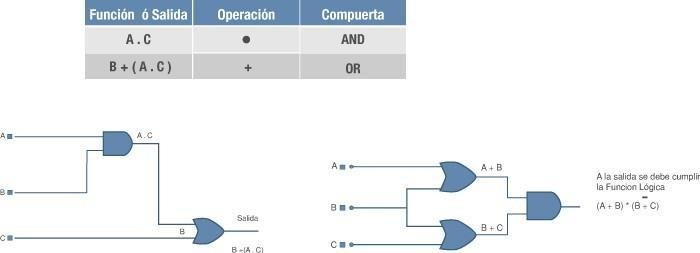
Esto evidencia que la simplificación no solo conserva la lógica original, sino que optimiza el diseño del esquema lógico.

|  |
| --- |
| **Función lógica sin simplificar:** (A + B) \* (B + C)  **Función lógica simplificada**: B + (A \* C) |

**B + (A \* C)**

**2 Operaciones (+. )**

**Figura 4.** Representación de funciones lógicas con compuertas



**Esquema lógico simplificado versus Esquema lógico sin simplificar**

Ambos esquemas realizan la misma función lógica, pero el esquema simplificado utiliza menos compuertas, lo que resulta en una reducción de costos y espacio. Esto resalta la importancia de simplificar las funciones lógicas al diseñar circuitos.

A continuación, se procederá a comprobar que las dos funciones, la original (A+B)\*(B+C)

y la simplificada B+(A\*C), generan los mismos resultados mediante sus respectivas tablas de verdad:

**Figura 5.** Tabla comparativa de funciones lógicas sin simplificar y simplificada

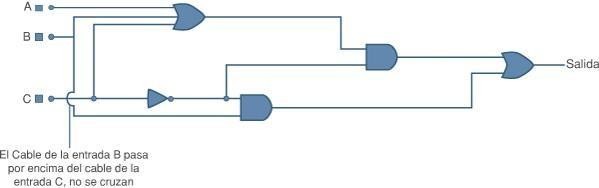
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variables o entradas** | | | **Función lógica sin simplificar** | **Variables o entradas** | | | **Salida parcial** | **Función lógica simplificada** |
| **A** | **B** | **C** | **(A+B)\*(B+C)** | **A** | **B** | **C** | **A\*C** | **B +(A\*C)** |
| o | o | o | o | o | o | o | o | o |
| o | o | 1 | o | o | o | 1 | o | o |
| o | 1 | o | 1 | o | 1 | o | o | 1 |
| o | 1 | 1 | 1 | o | 1 | 1 | o | 1 |
| 1 | o | o | o | 1 | o | o | o | o |
| 1 | o | 1 | 1 | 1 | o | 1 | 1 | 1 |
| 1 | 1 | o | 1 | 1 | 1 | o | o | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Se evidencia que las salidas para cada combinación son las mismas.

**Ejercicio práctico 3:**

A partir del esquema lógico, se debe elaborar la tabla de verdad y determinar la función lógica. Luego, simplificar la función lógica obtenida y comprobar que las salidas coinciden con las de la función original sin simplificar.

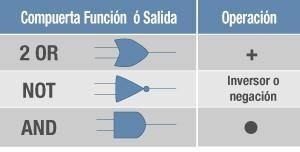
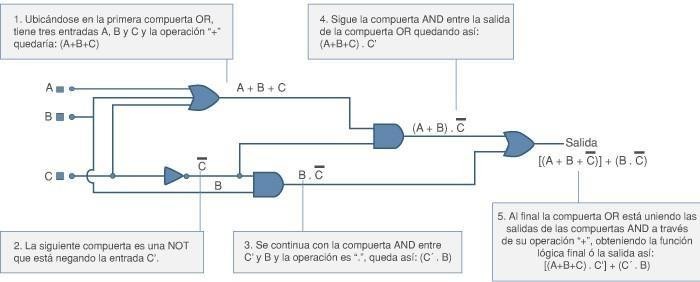
**Figura 6.** Circuito lógico con compuertas combinadas



**Solución:**

Para determinar la función lógica del esquema, es necesario identificar el tipo de compuertas lógicas y las operaciones que realiza cada una, de la siguiente manera:

**Figura 7. D**iagrama de circuito lógico y tabla de operaciones



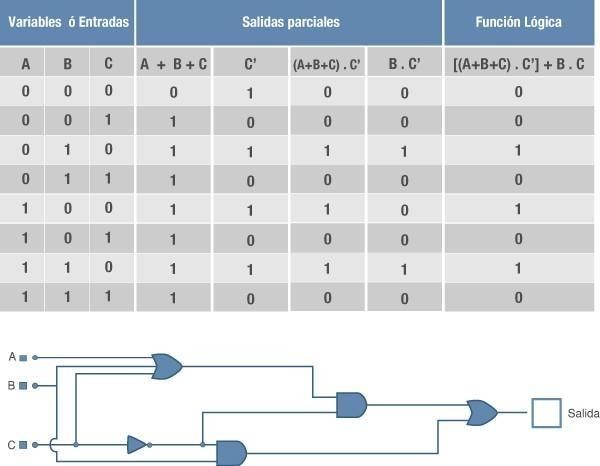
Con la función lógica, se procede a construir la tabla de verdad. La tabla de verdad y los niveles lógicos que corresponden a la salida de dicha función son los siguientes:

**Figura 8.** Tabla de verdad para función lógica compuesta

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variables o entradas** | | | **Salidas parciales** | | | | **Función lógica** |
| **A** | **B** | **C** | **A+B+C** | **C′** | **(A+B+C) \* C′** | **B\*C′** | **[(A+B+C) \* C′]+ B\*C′** |
| o | o | o | o | 1 | 0 | 0 | 0 |
| o | o | 1 | 1 | 0 | 0 | 0 | 0 |
| o | 1 | o | 1 | 1 | 1 | 1 | 1 |
| o | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | o | o | 1 | 1 | 1 | 0 | 1 |
| 1 | o | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | o | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Para comprobar la tabla se debe realizar la simulación:

**Figura 9.** Circuito lógico con compuertas combinadas



Simplificación de la función anterior:

[(A+B+C)⋅C′]+(B⋅C′)

Propiedades utilizadas:

Ahora, se comprobará que ambas funciones generan los mismos resultados mediante sus tablas de verdad.

**Figura 10.** Comparación de función lógica sin simplificar y simplificada

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variables o entradas** | | | **Función lógica sin simplificar** |  | **Variables o entradas** | | | **Salida parcial** |  | **Función lógica simplificada** |
| **A** | **B** | **C** | **[(A+B+C)] + B+ C′** | **A** | **B** | **C** | **A+B** | **C′** | **C′ \*(A+B)** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |  | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

Se evidencia que las salidas para cada combinación son las mismas

|  |  |
| --- | --- |
| **Figura 11. Esquema lógico inicial**  P239#y1 | **Figura 12.** Esquema lógico simplificado  P243#y1 |

|  |  |
| --- | --- |
| mujeres viajeras con personajes de maletas | **Caso**  El gerente de una empresa de transporte desea tener mayor control sobre el registro del equipaje de los pasajeros, por lo que solicita al ingeniero un circuito digital capaz de supervisar el paso de las maletas durante el recorrido de revisión.  **Objetivo** Diseñar un circuito digital capaz de detectar el equipaje desde que ingresa a la zona de carga, hasta que pasa la revisión y el registro, permitiendo finalmente su acceso a la zona de carga. |

Las variables de entrada son:

|  |  |
| --- | --- |
| Sensor Puerta 1 | Detecta el equipaje para permitir la apertura de la puerta 1 y el avance hacia el primer control. La salida es '0' cuando detecta equipaje y '1' cuando no. |
| Pulsador Puerta 2 | Permite la apertura de la puerta 2 si se oprime el botón y la puerta 1 previamente estuvo en '0'. La salida es '1' si el botón es oprimido y '0' si no. |
| Sensor Puerta 3 | Detecta el código de barras del tiquete para abrir la puerta 3 y permitir el paso a la zona de carga. La salida es '1' cuando detecta el código y '0' cuando no. |

**Solución**

Identificar entradas y salidas:

**Figura 13.** Tabla de condiciones de operación para puertas con diferentes entradas y salidas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Puerta 1 (una entrada)** | | **Puerta 2 (dos entradas)** | | **Puerta 3 (2 entradas)** | |
| Entrada | Salida | Entrada | Salida | Entrada | Salida |
| Sensor de objetos en "0". | Salida puerta 1 en "1". | Salida puerta 1 en "1". | Se abre la puerta 2 en "1". | Salida puerta 2 en "1". | Se abre la puerta 3 en "1". |
|  |  | Botón en "1". | Sensor que detecta tiquete en "1". | Se permite cargar el equipaje. |

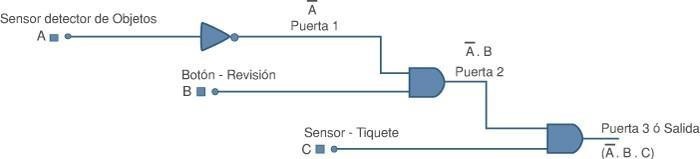
Identificación de compuertas lógicas

**Figura 14.** Tabla de operación y compuertas lógicas para tres puertas



El esquema lógico, según las entradas y salidas de las compuertas, es el siguiente:

**Figura 15.** Diagrama lógico de operación para control de puertas



La **función lógica** es:

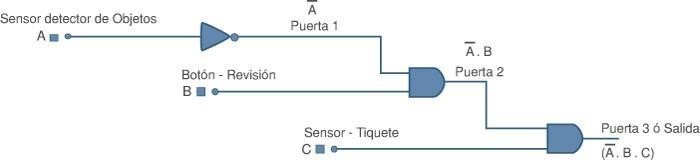
(A \* B \* C) = ( A’ \* B \* C)

Verifique, mediante la tabla de verdad, que el circuito cumple con el proceso requerido por el gerente. Esto implica evaluar todas las combinaciones posibles de las entradas y comprobar que las salidas corresponden a las condiciones establecidas para cada puerta del sistema.

**Figura 16.**  Tabla de verdad y condiciones de apertura de puertas

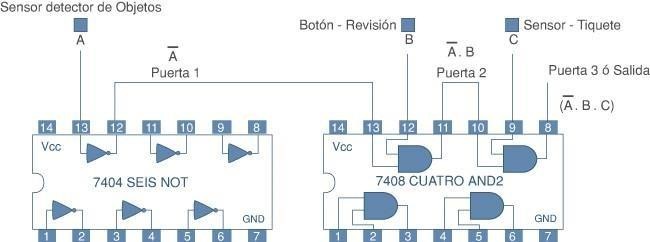
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable o entradas** | | | **Puerta 1** | | **Puerta 1** | | **Función lógica o salida puerta 3** |
| **A** | **B** | **C** | **A'** | **Si se cumple** | **A' · B** | **Si se cumple la condición** | **(A', B, C)** |
| 0 | 0 | 0 | 1 | Se abre | 0 | No abre | 0 |
| 0 | 0 | 1 | 1 | Se abre | 0 | No abre | 0 |
| 0 | 1 | 0 | 1 | Se abre | 1 | Se abre | 0 |
| 0 | 1 | 1 | 1 | Se abre | 1 | Se abre | 1 Se abre luego si se cumple |
| 1 | 0 | 0 | 0 | No abre | 0 | No abre | 0 |
| 1 | 0 | 1 | 0 | No abre | 0 | No abre | 0 |
| 1 | 1 | 0 | 0 | No abre | 0 | No abre | 0 |
| 1 | 1 | 1 | 0 | No abre | 0 | No abre | 0 |

Para finalizar se identificará que tipo de CI son necesarios para realizar el circuito:

**Figura 17.** Diagrama lógico de control de puertas

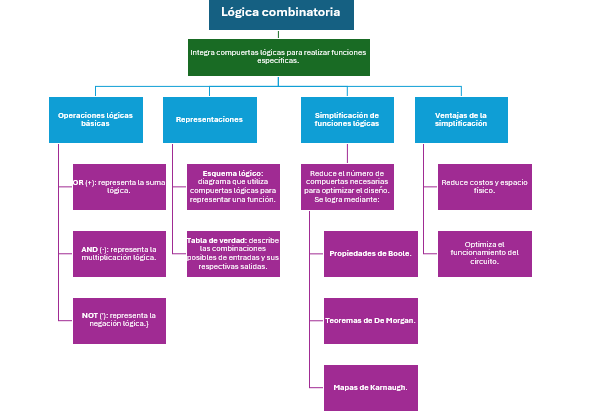
En el esquema se identifican tres compuertas necesarias para cumplir las funciones específicas del proceso, de las cuales dos son del tipo AND. Por lo tanto, los circuitos integrados adecuados para el diseño son:

**Figura 18.** Implementación del control de puertas con circuitos integrados



1. **SÍNTESIS**

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.



1. **ACTIVIDADES DIDÁCTICAS (Se debe incorporar mínimo 1, máximo 2)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Lógica combinatoria |
| Objetivo de la actividad | Identificar los conceptos fundamentales de la lógica combinatoria, incluyendo operaciones lógicas, herramientas de representación, y métodos de simplificación, para afianzar el conocimiento teórico y práctico en el diseño de circuitos digitales. |
| Tipo de actividad sugerida | Cuestionario |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | *CF0\_Actividad didactica* |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Lógica combinatoria | Fernando González. (2020). LÓGICA COMBINATORIA COMPUERTAS LÓGICAS CLASE 1. [Archivo de video] Youtube. | Video | <https://www.youtube.com/watch?v=b096QGPxj38&ab_channel=FernandoGonz%C3%A1lez> |
| Lógica combinatoria | Mundo Electrónica (2020). Compuertas lógicas y lógica combinacional | Curso de electrónica digital | #5 [Archivo de video] Youtube. | Video | <https://www.youtube.com/watch?v=OAA2B50e9nA&ab_channel=MundoElectr%C3%B3nica> |
| Lógica combinatoria | Electrónica FP. (2019). DeMorgan (Ejercicio) [Archivo de video] Youtube. | Video | <https://www.youtube.com/watch?v=N5YXG0KKLCc&ab_channel=Electr%C3%B3nicaFP> |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Compuerta lógica: | dispositivo digital que realiza operaciones básicas como a*nd, or o not.* |
| Esquema lógico: | representación gráfica de funciones lógicas mediante compuertas. |
| Lógica combinatoria: | rama de la electrónica digital que utiliza compuertas lógicas para realizar funciones específicas. |
| Mapas de Karnaugh: | técnica gráfica para simplificar funciones lógicas en circuitos digitales. |
| Operación *AND:* | representa la multiplicación lógica, donde la salida es 1 solo si todas las entradas son 1. |
| Operación *NOT:* | representa la negación lógica, invirtiendo el valor de la entrada. |
| Operación *OR:* | representa la suma lógica, donde la salida es 1 si alguna entrada es 1. |
| Simplificación lógica: | proceso de optimizar funciones lógicas reduciendo el número de compuertas necesarias. |
| Tabla de verdad: | herramienta para describir todas las combinaciones posibles de entradas y sus salidas. |
| Teoremas de De Morgan: | reglas matemáticas que permiten simplificar expresiones lógicas mediante negaciones. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Cidead, (s.f). Material interactivo sobre Lógica Binaria. <http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_index.htm>

Mc Graw Hill, (s.f), Introducción a los sistemas digitales. Unidad 1. En Mc Graw Hill. <http://www.mcgraw-hill.es/bcv/guide/capitulo/844817156X.pdf>

Neuroproductions, (s.f). Simulador On line.

ProfesorMolina, (s.f). Función interactiva de compuertas.

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor (es) | Magda Melissa Rodríguez Celis. | Experto temático | Regional Cundinamarca - Centro de Desarrollo Agroempresarial | 2012 |
| Paola Alexandra Moya | Evaluadora instruccional | Regional Antioquia - Centro de Servicios de Salud | 2024 |
|  | Olga Constanza Bermúdez Jaimes | Responsable Línea de Producción Antioquia | Regional Antioquia - Centro de Servicios de Salud | 2024 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |