



**UNIVERSIDAD  
DE ANTIOQUIA**

Título del Documento

Autor: Tu Nombre

Coautor: Otro Nombre

Universidad

April 7, 2024

# 1 Introducción

Se desea crear un circuito que permita hacer la suma de 16 números guardados en un vector, siguiendo el siguiente algoritmo:

$$\sum_{n=0}^{16} (A_n + A_{n+1})$$

Entre los componentes que se deben utilizar para desarrollar la solución están los Flip-Flops tipo **JK**

## 2 Almacenamiento

### 2.1 Creación del Vector

Se creó una unidad de almacenamiento el cual pudiera contener un número de 8 bits y a la vez permitirá almacenar un número predeterminado, hasta que llegara el nuevo número por el que se reemplazaría.

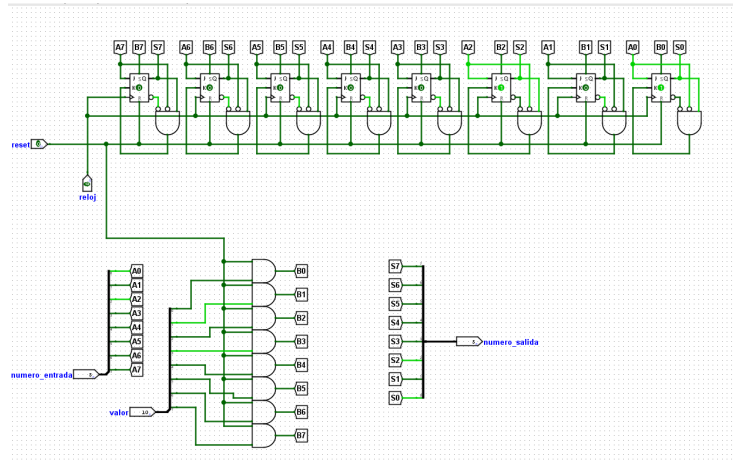


Figure 1: Unidad de almacenamiento

Así, de esta manera se podría crear de forma sencilla el vector de 16 posiciones y para el cual se permitiría almacenar números predeterminados. Además, se implementó una lógica para que solo escribiera en los arreglos correctos según la posición solicitada por el algoritmo y diario se retorne los numeros  $A_n$  y  $A_{n+1}$

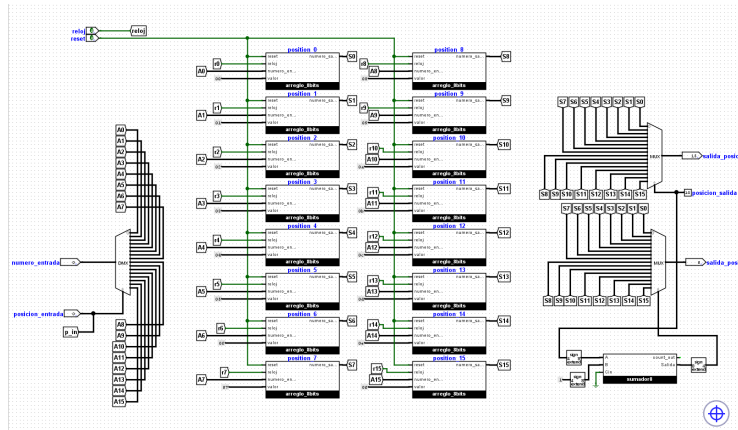


Figure 2: Vector de 16 posiciones

### 3 Sumador

Se construye primero un sumador de un bit full y con base a este se procede a construir uno para 8 bits que es la cantidad máxima para este ejercicio.

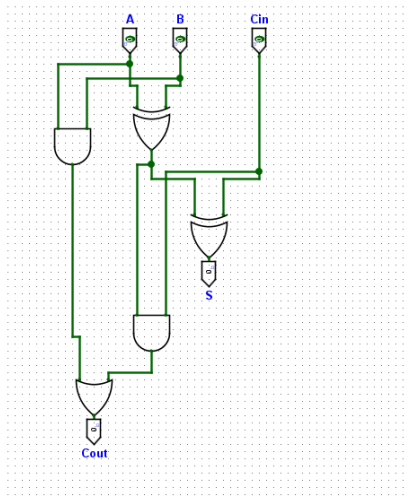


Figure 3: Sumador completo de 1 bit

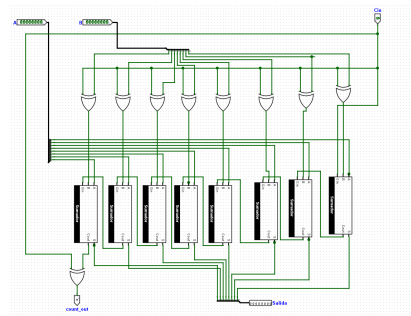


Figure 4: Sumador completo de 8 bits

## 4 Contadores

### 4.1 Contador unidad

Para este ejercicio se creó un contador de 4 bits (16 posiciones posibles) con el flip-flop tipo Jk conectados en cascada, el cual ayudaría para saber qué posición del vector guardar los datos, en que fase del algoritmo se encuentra y el estado de la máquina.

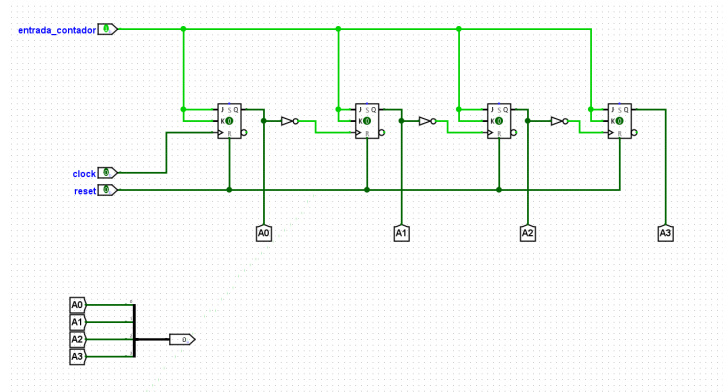


Figure 5: Contador de 4 bits

### 4.2 Contador par