# 2. Registro de almacenamiento de 6 bits

## Descripción del problema

Implementar un registro de trabajo universal de 6 bits de datos con dos modos de funcionamiento: Parallel-In/Serial-Out y Serial-In/Parallel-Out.

### Líneas de entrada

* Señal de control lectura/escritura (R/W), con 0 (cero) para lectura y 1 (uno) para escritura
* Modo de operación: con 0 (cero) para Parallel-In/Serial-Out y 1 (uno) para Serial-In/Parallel-Out
* Línea habilitadora
* Señal de reloj
* Reset: Setea la salida Q de los circuitos biestables (flip-flops) en 0.
* Datos: Entrada de 6 bits que indica el valor a agregar cuando la señal de control es 1. De lo contrario (modo Read), se ignora esta entrada. Cuando se escriben datos en paralelo se utilizan los 6 bits. Si se usa el modo en serie, solo el LSB será escrito en serie.

### Líneas de salida

* Datos: Salida de 6 bits que muestra el valor leído cuando R/W es 0 o se omite si el modo es escritura. Al leer en paralelo se cargan todos los bits, mientras que en serie solo se muestra el bit leído en el LSB.

## Solución del problema

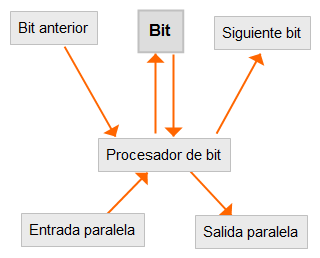
El circuito de almacenamiento, debe proporcionar métodos de lectura y escritura en modos distintos. Para implementar este registro, es necesario realizar dos operaciones básicas: almacenar los datos y leer/escribir según el modo.

#### Almacenamiento

Se utilizan 6 flip-flops D para almacenar los datos ingresados en el registro de almacenamiento universal.

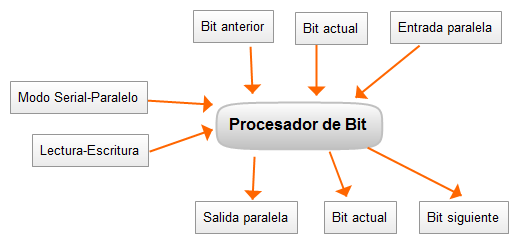
#### Procesamiento de bits

El circuito trabaja con un procesador de bits que permite enviar la información de cada bit a su salida correspondiente (para el modo lectura). El procesador también permite seleccionar el origen que será almacenado en el siguiente ciclo de reloj.



Procesador de bits

En el registro, se trabaja con un procesador de bit por cada bit de almacenamiento (Flip-Flop), por lo que actualmente el circuito de almacenamiento universal posee 6 procesadores de bit. Al bit asignado a un procesador de bit le llamamos “*bit actual*”, al bit ubicado antes del actual se le conoce como “*bit anterior*”, mientras que al bit posterior lo llamamos “*bit siguiente*”. Adicionalmente, cada bit posee su propia entrada y salida en paralelo, que se utiliza cuando se trabaja en alguno de estos modos.



Modo de control del procesador de bit

El procesador de bits, recibe las señales de **Modo** y de **R/W**. Utilizando esas señales se puede definir cuál es el dato que debe quedar guardado en el bit actual, en el siguiente y en la salida paralela asociada al bit. A continuación una tabla que muestra esos valores:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Salida | Lectura Serial | Lectura Paralela | Escritura Serial | Escritura Paralela |
| Bit actual | Bit anterior | Bit actual | Bit anterior | Entrada paralela |
| Bit siguiente | Bit actual | Se omite (cero) | Bit actual | Se omite (cero) |
| Salida paralela | Se omite (cero) | Bit actual | Se omite (cero) | Se omite (cero) |

### Modos de trabajo

El procesador de bits (nextBit) en combinación con los bits de almacenamiento (flip-flops D) realiza los siguientes métodos:

#### Lectura serial

En un ciclo de reloj se muestra el bit contenido en el LSB y los bits se corren a la derecha (ShR) conservándose el LSB en el MSB.

#### Lectura paralela

Cada bit se envía a su salida paralela definida. El bit se conserva en su posición original.

#### Escritura serial

El MSB se cambia por valor ingresado en la entrada serial (el LSB de la entrada paralela). El anterior MSB se coloca en el siguiente bit y así sucede con cada bit. Ocurre un corrimiento hacia la derecha (ShR) en el que se pierde el valor del LSB.

#### Escritura paralela

A todos los bits se les asigna su entrada paralela correspondiente.

### Otros casos

Adicionalmente, para el primer y el último bit se crean ciertas condiciones que permiten redirigir el valor del último bit hacia el primero como si fuese su anterior.

## Funciones y tiempos

**Inicializar:** El tiempo de propagación es de 11 pasos de reloj.

**Reset**: Al activar esta señal se resetean todos los flip-flops, dejando en todos el valor de Q = 0. El tiempo de propagación de esta función es de 20 pasos de reloj en la primera vez que se utiliza. Las siguientes veces tarda sólamente 7 pasos del reloj.

**Procesar bit:** Muestra como salidas el valor del bit actual, del siguiente y de la señal de salida paralela dependiendo del valor actual, del valor anterior y de la entrada paralela tomando en consideración las entradas de modo seleccionadas. El tiempo de propagación para esta función en el circuito es de 6 pasos.

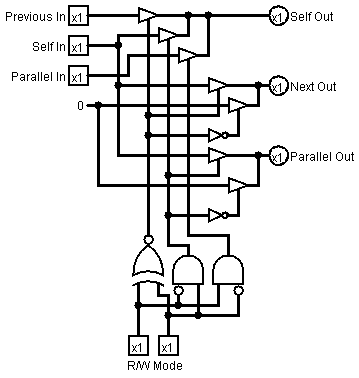
**Habilitar circuito:** Permite realizar lecturas y escrituras. El circuito tarda 4 pasos en ser habilitado.

**Tiempos de lectura y escritura:**

|  |  |  |
| --- | --- | --- |
| **Modo** | **Serial** | **Paralelo** |
| **Lectura** | **21** | **10** |
| **Escritura** | **21** | **12** |

*Valores dados en pasos del reloj. Los tiempos para funciones en serie son cambios o lecturas en un solo bit.*

#### Diagrama de circuito del procesador de bits

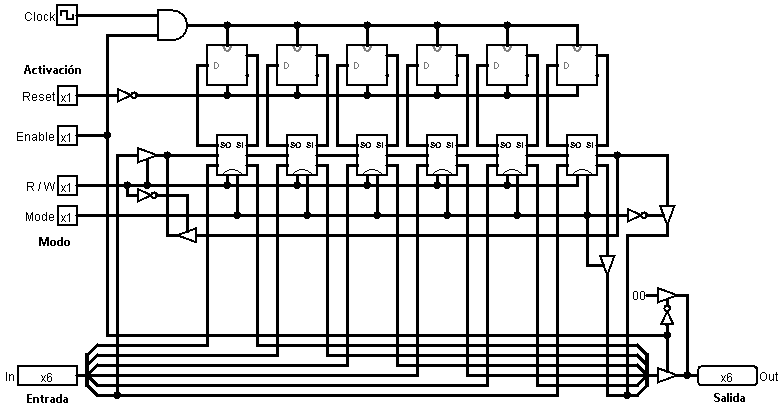
**

Entradas de control

Salidas de bits

Entradas de bits

#### Diagrama del Registro Universal de Almacenamiento



nextBit (Procesadores de bit)

Entrada/Salida paralela

Flip-Flops D

# 4. Subsistema de memoria

## Descripción del problema

Se debe crear una pastilla de memoria RAM para palabras de 6 bits capaz de almacenar un total de 16 palabras.

Además, y de acuerdo al modelo seguido para el desarrollo de la arquitectura (Von Neumman), es necesario implementar los registros utilizados para acceder la memoria (ya que los otros componentes no pueden accederla directamente). Estos registros son el MAR (Memory Address Register) y el MDR (Memory Data Register).

### RAM

#### Líneas de entrada

* Señal de control lectura/escritura (R/W): R/W=0 para lectura y R/W=1 para escritura.
* Línea habilitadora
* Clock
* Reset: R=1 “resetea” los Flip Flops asignando un valor de 0 a la salida Q.
* Datos: De 6 bits, indica el dato a almacenar en caso de que R/W=1. Si R/W=0 se debe ignorar.
* Dirección: De 4 bits, indica la dirección de memoria que se quiere leer (en caso de que R/W=0) o a la que se quiere escribir (en caso de que R/W=1).

#### Línea de salida

* Datos: De 6 bits, indica el dato que se leyó en caso de que R/W=0. Si R/W=1 deben salir ceros.

### MAR

#### Líneas de entrada

* Habilitadora
* R/W
* Palabra de 4 bits que indica la dirección de memoria que se desea acceder
* Clock
* Reset

#### Línea de salida

* Palabra de 4 bit con la dirección de memoria que se desea acceder

### MDR

#### Líneas de entrada

* Habilitadora
* R/W
* Palabra de 6 bits que indica el valor a leer/escribir
* Clock
* Reset

#### Línea de salida

* Palabra de 6 bit con el valor a leer/escribir

## Solución del problema

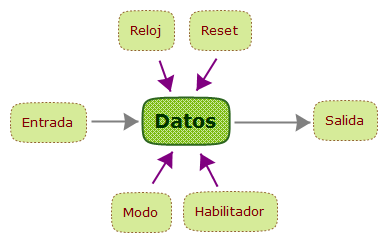
El sistema de memoria necesita de 3 componentes, todos ellos son secuenciales. En los 3 circuitos es necesario realizar almacenamiento, lectura y escritura. Describiremos el modo en que resolvimos cada componente en el orden en que se hizo.

### RAM

El circuito debe ser capaz de almacenar en distintas celdas señaladas por una dirección, un valor de igual forma dado. Una vez más necesitamos dividir el circuito en algunas tareas o funciones básicas.

#### Celdas de 6 bits

Tenemos que almacenar los datos en palabras de 6 bits, para lo cual realizamos un circuito que contiene 6 Flip-Flops D. Esta celda permite escribir los datos ingresados cuando el circuito está habilitado, se encuentra en modo escritura y el reloj se activa; en este caso el circuito no envía nada en la salida (ni ceros ni unos). Cuando el circuito está habilitado pero el modo es de lectura, los circuitos biestables envían su contenido directamente a la salida.



Celda de almacenamiento

Para esta celda, la señal de reset activa provoca que los datos almacenados se eliminen (todos en cero) asincrónicamente. El modo funciona como R/W, lo que quiere decir que al enviar 0 se lee, de lo contrario, se escribe.

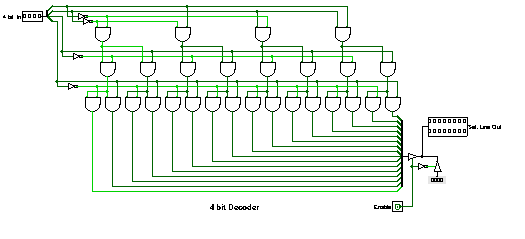
#### Decodificador

Para leer o escribir un dato, es necesario dar una dirección, la cual es un índice que se encuentra codificado como la posición de la celda solicitada. Para poder activar esa celda y desactivar todas las demás, es necesario crear un decodificador que active según el índice. En el caso de que la RAM esté deshabilitada en general, se envía una señal vacía (con ceros) hacia todas las direcciones.



Flujo del decodificador

El siguiente es el diagrama del decodificador

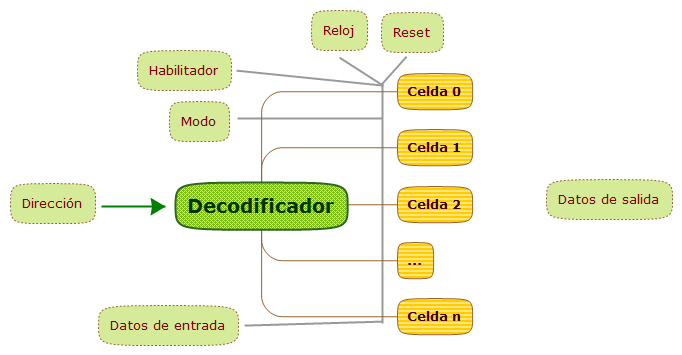


ANDs selectores

Cada señal de salida se puede activar con una combinación única de la entrada de dirección. De esta manera por cada dirección se activa una y sólo una señal de salida. Las combinaciones se obtienen mezclando las 4 líneas de entrada (y sus respectivos complementos) mediante compuertas AND.

#### RAM

Finalmente utilizamos los componentes creados anteriormente para formar un sistema de memoria. El decodificador habilita la señal de la celda de la que se desea leer o escribir utilizando el índice de dirección. La señal de R/W decide si los flip-flops en las celdas estarán recibiendo o enviando datos. La señal del reloj permite que se escriban los datos enviados en el flanco ascendente. La entrada de reset cambia el valor de todas las celdas a ceros. El habilitador decide si el circuito permite lecturas o escrituras, desactivando o activando las entradas de reloj de las celdas y la salida de datos.

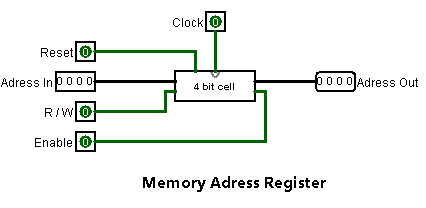


Componentes de la RAM

### MAR

El circuito de registro de dirección de memoria es más sencillo. Se necesita almacenar un dato de 4 bits que representa la dirección que será interpretada por el decodificador más adelante.

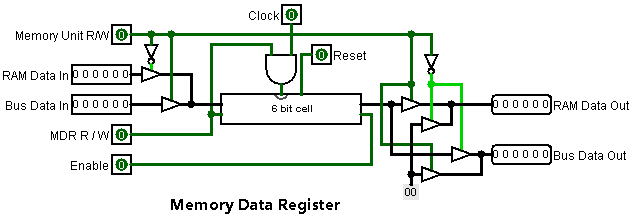
Para almacenar la dirección se requiere una celda de datos de 4 bits. Esta es muy similar a la utilizada en la RAM, por lo que no abordaremos mucho en cómo se implementó. El diseño de este componente es muy sencillo y tiene una estructura realmente simple, como se observa en la imagen inferior.



La celda de 4 bits recibe las señales regulares de Reloj, Habilitación, Dirección (Datos), Escritura-Lectura y Reset. La única salida es la dirección escrita en la celda, la cual se puede observar al habilitar el circuito en modo lectura.

### MDR

Este circuito es un tanto más complejo que el MAR. Difiere de este último en su capacidad de almacenamiento, que aumenta a 6 bits, además de que tiene dos modos de entrada y salida.



Modos de escritura/lectura del MDR

1. El modo en cero indica al MDR que el modo escritura debe realizarse desde la RAM, mientras que la lectura debe ser hacia el bus de datos. Este modo es el básico para lecturas de memoria.
2. Al activarse la señal (en uno), las escrituras se realizan desde el bus y los datos se leen en la RAM. Este modo permite escribir del bus a la RAM.

Utilizando estos modos, es posible seleccionar desde donde escribir y hacia dónde dirigir los datos leídos.

## Tiempos de propagación

|  |  |  |  |
| --- | --- | --- | --- |
| Componente | Lectura | Escritura | Reset |
| RAM | 7 | 10 | 8 |
| MAR |  |  |  |
| MDR |  |  |  |
| Decoder | - | 7 | - |

Valores dados en pasos de reloj.