

# Reporte Actividad 04: Diseño y simulación de memorias

Estrada Rivera Gustavo de Jesús 220746114 | 20/09/2025

## INTRODUCCIÓN

En los sistemas digitales, la memoria es un componente esencial: almacenar instrucciones, datos temporales, configuraciones, entre otros. Dependiendo de la aplicación, se necesita una memoria que sea muy rápida, resistente, económica, o una combinación de estas características. En este texto examinamos los tipos principales de memoria, sus propiedades y su vínculo con los flip-flops, que son las unidades básicas de almacenamiento.

## Tipos de memoria

Tipo	¿Volátil / No volátil?	¿Cómo funciona?	Ventajas principales	Desventajas principales	Usos típicos
<b>ROM (Read-Only Memory)</b>	No volátil	Memoria "solo lectura" cuya información se escribe en la fabricación (mask ROM) o mediante tipos especiales que permiten escritura, pero con restricciones.	Conserva datos sin necesidad de alimentación eléctrica. Ideal para firmware, arranque, constantes esenciales del sistema.	No puede modificarse. Menos flexibilidad. Es más lento para escritura si es de los tipos reprogramables.	Firmware, BIOS, microcontroladores, dispositivos embebidos, memorias de arranque.
<b>RAM (Random Access Memory)</b>	Volátil	Permite lectura y escritura rápidas, acceso aleatorio (cualquier dirección sin tener que recorrer secuencialmente). Se divide principalmente en SRAM y DRAM.	Velocidad alta, latencia baja, ideal para almacenamiento temporal de trabajo del procesador.	Los datos se pierden al cortar la energía. Algunos tipos (DRAM) requieren refresco. Costo más alto por bit en SRAM.	Memoria principal de ordenadores, buffers, cachés, procesamiento en tiempo real.
<b>SRAM (Static RAM)</b>	Volátil	Cada bit se almacena usando flip-flops, típicamente un circuito con varios transistores (por ejemplo 6	Muy alta velocidad, no requiere refresco, latencia baja. Ideal para cachés.	Costo alto por bit, consume más espacio físico en el chip (densidad menor), mayor consumo de energía	Cachés de CPU.

		transistores por célula) que se mantiene mientras haya corriente.		cuando es muy grande.	
<b>DRAM (Dynamic RAM)</b>	Volátil	Cada bit se almacena como carga eléctrica en un condensador, acompañada de un transistor; la carga decae con el tiempo, por lo que se necesita refresco periódico.	Mucha densidad de almacenamiento, costo menor por bit que SRAM, útil para grandes bloques de memoria principal.	Más lento que SRAM (latencia mayor), consumo de energía adicional para refresco, complejidad del controlador de memoria.	Memoria principal de PC o servidores.
<b>EEPROM (Electrically Erasable Programmable ROM)</b>	No volátil	Similar a los tipos ROM reescribibles, se puede borrar y reprogramar eléctricamente, generalmente byte a byte o página a página.	Permite modificaciones en campo, retiene datos tras apagado, útil para almacenamiento de parámetros o configuraciones.	Escritura relativamente lenta, número limitado de ciclos de escritura, costo por bit más alto que memorias exclusivamente de lectura o flash en ciertos contextos.	Configuración de microcontrolador, registros de usuario, parámetros que se guardan.
<b>Flash Memory</b>	No volátil	Deriva de EEPROM; permite borrar grandes bloques de memoria (por ejemplo, páginas o bloques), y luego reescribirlos. Hay variantes (NOR-flash, NAND-flash) con diferentes velocidades y usos.	Alta densidad, buen equilibrio costo/bit, bastante utilizado para almacenamiento masivo no volátil (tarjetas, memorias USB, SSDs).	Escritura más lenta que lectura; los bloques deben borrarse antes de escribir; número limitado de ciclos de borrado/escritura; desgaste si se usan mucho.	Almacenamiento masivo no volátil.

## **Características comparativas**

**Velocidad de acceso / latencia:** *SRAM > DRAM >* muchas memorias no volátiles como *Flash* o *EEPROM*. Las *ROM* suelen tener una lectura rápida, pero en los tipos reprogramables, la escritura puede ser muy lenta.

- **Persistencia (capacidad de retener datos sin energía):** solo los tipos no volátiles como *ROM*, *EEPROM* y *Flash* lo hacen; la *RAM* (incluyendo *SRAM* y *DRAM*) no.
- **Costo por bit / densidad:** las memorias volátiles *DRAM* son más económicas por bit que *SRAM*; las memorias no volátiles de alta densidad (como NAND flash) ofrecen un buen costo por bit para almacenamiento masivo.
- **Resistencia al desgaste:** *Flash* y *EEPROM* tienen un número limitado de ciclos de escritura/borrado; algunas memorias no volátiles emergentes mejoran este aspecto, aunque sigue siendo una limitación.
- **Consumo de energía:** *SRAM* consume más energía durante escritura/lectura; *DRAM* también, y requiere refresco continuo. Las memorias no volátiles consumen poca energía para mantener datos, pero la escritura puede necesitar tensiones especiales o un mayor consumo momentáneo.

## **Memorias síncronas vs asíncronas: principales diferencias**

En diseño digital, "síncrono" y "asíncrono" se refieren a cómo se controla el momento en que las operaciones de cambio en la memoria (o en circuitos secuenciales) se hacen efectivas, en relación con el reloj (clock) u otras señales.

- **Memoria síncrona:** todas o la mayoría de las operaciones (lectura, escritura, activación de filas, etc.) están coordinadas por una señal de reloj. El reloj determina momentos específicos en que los datos deben ser válidos y cuándo realizar cambios de estado. Ejemplo: *SDRAM* (Synchronous *DRAM*), donde los comandos externos se sincronizan con un reloj.
- **Memoria asíncrona:** no dependen de un reloj externo para coordinar los cambios; las señales de control pueden hacer que la memoria responda de inmediato, sin esperar un flanco de reloj. Un ejemplo son las *DRAM* antiguas o variantes de *RAM* sin interfaz sincronizada.

## **OBJETIVOS**

- ❖ **Objetivo General:** Comprender el funcionamiento y las diferencias de las memorias digitales mediante práctica en Verilog y su simulación, junto con investigación teórica.
- ❖ **Objetivo Particulares:**
  - **Investigar y documentar** sobre los tipos de memoria (ROM, RAM, etc.), sus características y la relación fundamental entre los flip-flops y la memoria RAM.
  - **Implementar** en Verilog los distintos tipos de memorias digitales: ROM, RAM asincrónica y RAM sincrónica, junto con sus testbenches.
  - **Integrar** los conocimientos teóricos y prácticos en el presente reporte que evidencie la comprensión de los conceptos y la correcta implementación en Verilog.

## **DESARROLLO**

A continuación, se pasar a explicar los códigos utilizados y sus simulaciones.

**ROM:** La ROM (Read Only Memory) se implementa como un **arreglo de registros** de 8 bits con 11 posiciones (ROM [0:10]).

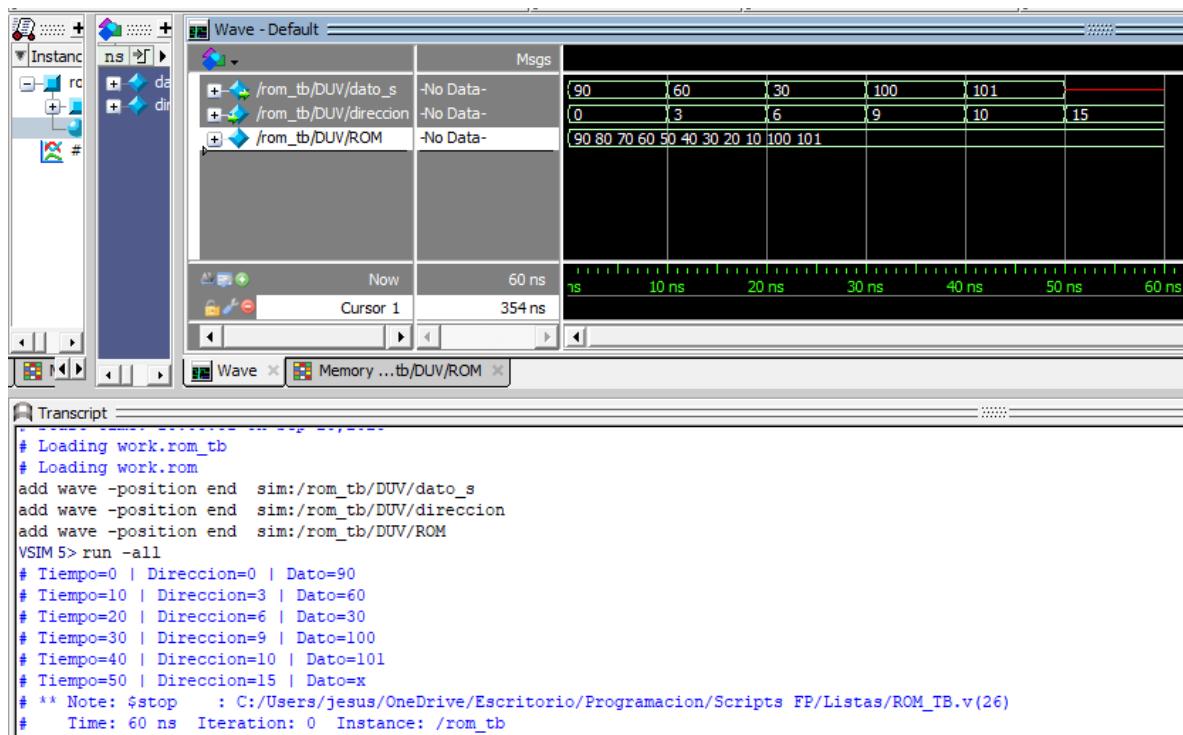
- Los valores de cada dirección se inicializan en un bloque initial.
- La lectura se realiza de forma **combinacional** en un always @\*, asignando a la salida dato\_s el contenido de la posición solicitada por la entrada direccion.
- Al ser memoria de solo lectura, no es posible modificar su contenido durante la simulación.

En esta memoria, cada dirección contiene un valor fijo que puede leerse en cualquier momento.

En su testbench (banco de pruebas) por su lado:

- Se define la señal direccion y se conecta al módulo rom.
- Se usan diferentes valores de dirección (0, 3, 6, 9, 10, 15) para comprobar que la salida dato\_s corresponde al valor almacenado.
- Se incluye \$monitor para mostrar en consola el tiempo de simulación, la dirección y el dato leído.

## Simulación (Waveform):



Se observa el comportamiento en el **waveform**, verificando que los valores son correctos y constantes, con este banco de pruebas se valida que la ROM responde de inmediato a cualquier dirección.

**RAM ASÍNCRONA:** La RAM (Random Access Memory) asíncrona también se implementó como un **arreglo de registros** de 8 bits (RAM [0:10]), pero a diferencia de la ROM, se permite lectura y escritura.

- Cuando la señal EN está en **1**, se realiza una **escritura inmediata** en la dirección indicada con el dato de entrada (Dato\_E).
- Cuando EN está en **0**, se habilita la **lectura inmediata**, y la salida dato\_s refleja el contenido de la dirección especificada.
- Todo esto se controla en un bloque always @\*, por lo que el acceso a la memoria ocurre **de forma asincrónica**, sin depender de un reloj.

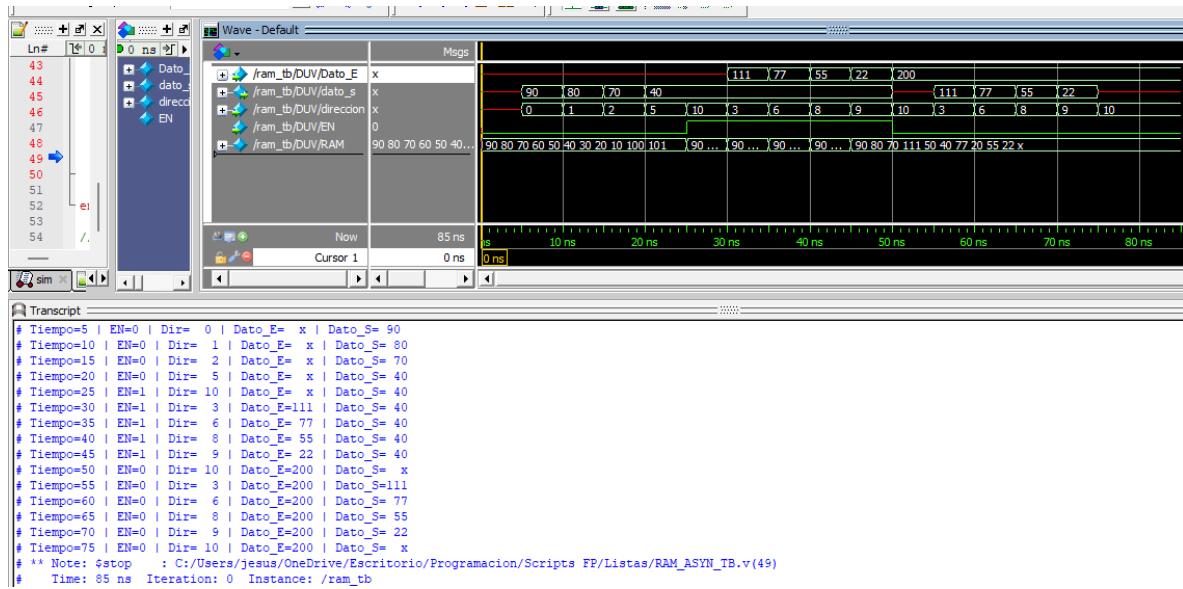
Esto significa que los cambios en entradas (*direccion*, *Dato\_E*, *EN*) se reflejan de manera instantánea en la salida.

En su testbench por su lado:

- Se definen las señales: *direccion*, *Dato\_E* y *EN*.
- Se realizan dos fases:

- Lectura (EN=0):** Es en donde se prueban varias direcciones inicializadas para verificar que los valores se lean correctamente.
- Escritura (EN=1):** Se escriben nuevos datos en direcciones específicas (por ejemplo: 3, 6, 8, 9, 10).
- Finalmente se vuelve a poner EN=0 y se hacen lecturas en esas mismas direcciones para comprobar que los valores realmente cambiaron.

### Simulación (Waveform):



Este testbench confirma que la escritura y lectura asincrónica funcionan de inmediato.

**RAM SÍNCRONA:** La RAM síncrona sigue la misma estructura del arreglo de 8 bits (RAM [0:10]), pero en este caso la lectura y escritura están controladas por un **reloj (clk)**.

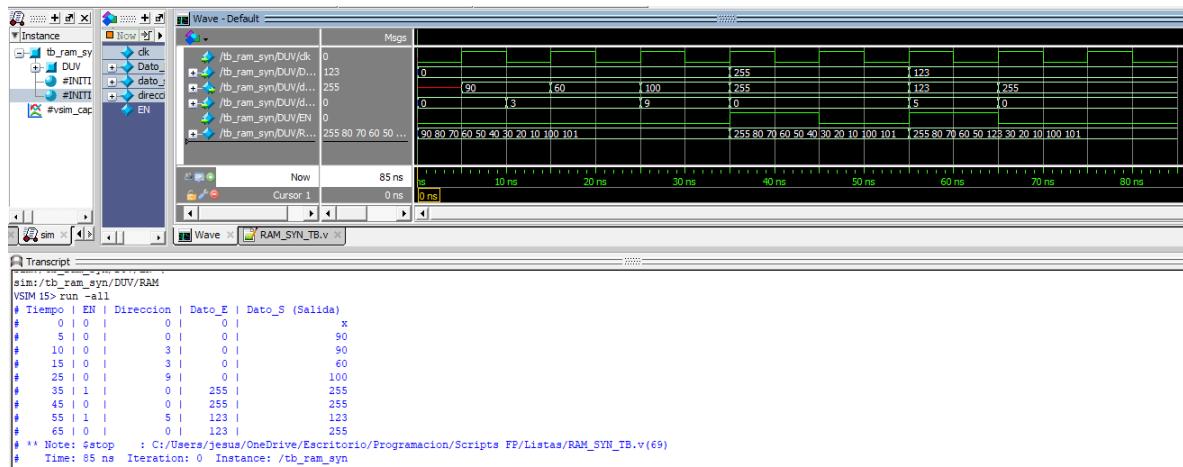
- Dentro de un bloque always @(posedge clk), se ejecutan las operaciones de memoria.
- Si EN = 1, se realiza una **escritura sincronizada al flanko positivo del reloj**.
- Independientemente de la escritura, la salida dato\_s refleja el valor de la dirección accedida después del flanko de reloj.

Una diferencia para destacar de la RAM asincrónica, esta versión **sincroniza todas las operaciones con el reloj**, lo que la hace más estable y adecuada para sistemas digitales complejos.

En su testbench por su lado:

- Se genera una señal de reloj (clk) con periodo de 10 ns.
- Se prueban distintos casos:
  1. **Lectura de valores inicializados:** Es en donde se definen las direcciones (Por ejemplo: dirección 3 y 9).
  2. **Escritura sincronizada:** Se activa cuando EN=1 y se carga un nuevo dato en una dirección (Por ejemplo: dirección 0 con valor 255).
  3. **Lectura post-escritura:** Se desactiva EN y se verifica que el valor escrito se mantiene en memoria.
  4. **Escritura y lectura en otra dirección:** Por ejemplo, dirección 5 con valor 123).

### Simulación (Waveform):



Aquí se observa que los cambios solo ocurren en los **flancos de reloj**, validando el comportamiento de una memoria sincrónica.

## CONCLUSIONES

Este proyecto de diseño y simulación de memorias en Verilog ha sido una experiencia completa que me permitió conectar teoría y práctica en arquitectura de computadoras.

La mayor dificultad fue entender la diferencia entre el comportamiento síncrono y asíncrono, especialmente la latencia de lectura en la RAM síncrona y la complejidad para depurar el testbench. También fue un reto recordar la sintaxis de Verilog para arreglos y cubrir casos de prueba importantes.

Lo más valioso que pude identificar fue convertir conceptos abstractos en algo tangible, comprendiendo el rol crucial del reloj, la importancia de un buen testbench y la diferencia entre ROM y RAM en código, y gracias a estos conocimientos que se relacionan directamente con tecnología cotidiana, como el firmware en ROM, la memoria caché del CPU y los bancos de memoria en sistemas embebidos.

Puedo concluir que esta práctica no solo ayudó a reforzar lo aprendido sobre los tipos de memoria digital, sino que también sirvió para mejorar las habilidades en programación con Verilog, creación de bancos de prueba y comprender mejor las simulaciones. Esto crea una base fuerte para proyectos más complejos en diseño digital y descripción de hardware.

## REFERENCIAS

- Alex MO. (2020, 1 junio). memorias; usos, tipos y funcionamiento [Vídeo]. YouTube. <https://www.youtube.com/watch?v=3kJKGltPbNk>
- Ron Mattino. (2023, 17 febrero). How do computers work? CPU, ROM, RAM, address bus, data bus, control bus, address decoding. [Vídeo]. YouTube. <https://www.youtube.com/watch?v=4knBXkN1GEU>
- Electrónica FP. (2020, 14 octubre). Introducción a las memorias. Parte 1 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=s-ARR8wGOhg>
- Les Ingénieurs. (2016, 2 marzo).  FLIP FLOP RS, D, JK y T [Todos los Flip-Flops en 6 minutos 😊 100] Electrónica Digital [Vídeo]. YouTube. <https://www.youtube.com/watch?v=3xy9WDgfqJg>
- Memory types. (n.d.). [https://mil.ufl.edu/3701/docs/umbc\\_8086\\_memory1.html](https://mil.ufl.edu/3701/docs/umbc_8086_memory1.html)
- Admin. (2025, April 3). The different between RAM, ROM, SRAM, DRAM, EEPROM, NAND, NOR. Wonderful PCB. <https://www.wonderfulpcb.com/blog/differences-between-ram-rom-sram-dram-eeprom-nand-nor/>
- Electronics, A. (2022, October 13). Computer memory types explained: Flash, SSD, RAM, EEPROM, HDD. Arrow.com. <https://www.arrow.com/en/research-and-events/articles/computer-memory-types-explained-flash-ssd-ram-eeprom-hdd>
- Ot, A. (2025, March 11). 9 Types of Computer Memory Defined. Enterprise Storage Forum. <https://www.enterprisestorageforum.com/hardware/types-of-computer-memory/>
- Wikipedia contributors. (2025, September 17). Static random-access memory. Wikipedia. [https://en.wikipedia.org/wiki/Static\\_random-access\\_memory](https://en.wikipedia.org/wiki/Static_random-access_memory)
- Efyian, E. (2025, July 31). Flip-Flops in digital electronics – types, truth table, circuit, and uses. Electronics for You – Official Site ElectronicsForU.com. <https://www.electronicsforu.com/technology-trends/learn-electronics/flip-flop-rs-jk-t-d>
- Types of memory in embedded systems. (n.d.). <https://barrgroup.com/blog/types-memory-embedded-systems>

GeeksforGeeks. (2025, July 11). Difference between synchronous and asynchronous sequential circuits. GeeksforGeeks. <https://www.geeksforgeeks.org/computer-organization-architecture/difference-between-synchronous-and-asynchronous-sequential-circuits/>

Team, L. (2024, November 17). 7 Main types of memory: Characteristics, uses, and applications | LinkSemicon. linqchip. <https://www.linksemicon.com/blog/types-of-memory/>

<https://web.eecs.umich.edu/~prabal/teaching/eecs373-f11/readings/sram-technology.pdf>

GeeksforGeeks. (2025, July 12). Difference between Volatile Memory and NonVolatile Memory. GeeksforGeeks. <https://www.geeksforgeeks.org/operating-systems/difference-between-volatile-memory-and-non-volatile-memory/>