# Project Overview

Project Title: Enhanced SRAM Controller with Advanced Features.

Objective: To design an SRAM controller with advanced features such as error detection and correction, burst mode, power management, security features, reliable clock domain crossing and byte addressable access.

## Theory:

An SRAM (Static RandomAccess Memory) controller manages the interface between a processor and SRAM memory. It handles read and write operations, ensuring data is accurately stored and retrieved

**Basic Working of SRAM Controller**

**Initialization**: Sets up control signals and initial states for operation.

**Addressing**: When the processor needs data, it sends the address to the SRAM controller. The controller decodes this address to access the specific memory location.

**Read Operation**:

1. **Processor Request**: The processor sends the read command with the memory address.
2. **Address Decoding**: The controller decodes the address.
3. **Data Retrieval**: The controller sends the address to SRAM, retrieves the data, and sends it back to the processor.

**Write Operation**:

1. **Processor Request**: The processor sends the write command with the memory address and data.
2. **Address Decoding**: The controller decodes the address.
3. **Data Storage**: The controller writes the data to the specified address in the SRAM.

**Control Signals**:

* **Chip Select (CS)**: Activates the SRAM chip.
* **Write Enable (WE)**: Indicates a write operation.
* **Output Enable (OE)**: Indicates a read operation.
* **Address Bus**: Carries the address for memory operations.
* **Data Bus**: Carries data to/from the memory.

**Additional Features**

Advanced SRAM controllers may include features like error detection and correction, power management, and burst mode for enhanced performance and reliability.

It’s essential in scenarios needing quick, reliable memory access and is widely used in cache memory, routers, and other high-speed applications.

## Features

1. **Basic Read/Write Operations**
   * **Functionality**: Supports basic read and write operations for SRAM.
   * **Implementation**: Registers for storing addresses and data, control signals for read (oe\_n) and write (we\_n) operations.
2. **Error Detection and Correction (EDAC)**
   * **Functionality**: Detects and corrects errors in stored data using parity bits.
   * **Implementation**: Parity bits calculated during write operations and verified during read operations.
3. **Security Features**
   * **Access Control**: Ensures that only authorized users can access the SRAM using a password mechanism.
   * **Encryption**: Data encryption and decryption using a simple XOR encryption technique to protect data integrity.
4. **Byte-Addressable Access**
   * **Functionality**: Allows access to individual bytes within a word.
   * **Implementation**: Byte-enable signals (byte\_en) to enable specific bytes during read/write operations.
5. **Burst Mode**
   * **Functionality**: Supports multiple consecutive memory accesses in a single burst for improved performance.
   * **Implementation**: Control signals for burst mode (burst\_mode) and burst length (burst\_length) to manage multiple accesses efficiently.
6. **Power Management**
   * **Power Gating**: Reduces power consumption by turning off inactive parts of the circuit.
   * **Dynamic Voltage Scaling**: Implied through the power\_save\_mode signal to control power-saving mechanisms.
7. **Clock Domain Crossing (CDC)**
   * **Functionality**: Ensures reliable operation when the SRAM controller operates in a different clock domain than the processor.
   * **Implementation**: Synchronizers for signals (data, we\_n, oe\_n) to prevent metastability issues.

## Detailed Implementation

### Module Description:

* **sram\_controller**: Main module integrating all the features mentioned above.

### Inputs:

* **clk:** Clock signal for the SRAM controller.
* **rst\_n:** Reset signal (active low).
* **addr:** Address bus for SRAM access.
* **data:** Data bus for read/write operations.
* **byte\_en:** Byte enable signals for byte-addressable access.
* **we\_n:** Write enable signal (active low).
* **oe\_n:** Output enable signal (active low).
* **password**: Password for access control.
* **clk\_processor:** Clock signal from the processor domain.
* **burst\_mode:** Signal to enable burst mode.
* **burst\_length:** Length of the burst for burst mode.

### Outputs:

* **cs\_n:** Chip select signal (active low).
* **sram\_addr:** Address bus for SRAM.
* **sram\_data:** Data bus for SRAM.
* **sram\_we\_n:** Write enable signal for SRAM (active low).
* **sram\_oe\_n:** Output enable signal for SRAM (active low).
* **error\_flag:** Signal indicating error detection.
* **access\_denied:** Signal indicating access denial.
* **power\_save\_mode:** Signal indicating power save mode.

### Internal Signals:

* **data\_reg:** Register to hold data.
* **addr\_reg:** Register to hold address.
* **we\_n\_reg, oe\_n\_reg:** Registers for write and output enable signals.
* **last\_address\_written:** Flag to prevent writing to the last address.
* **error\_detected:** Flag for error detection.
* **retry\_count:** Counter for retry mechanism.
* **access\_granted:** Flag for access control.
* **power\_save**: Flag for power saving mode.
* **burst\_count:** Counter for burst mode.
* **burst\_active:** Flag for burst mode activation.
* **parity:** Parity bit for error detection and correction.
* **calculated\_parity:** Calculated parity bit during read operations.
* **data\_sync, we\_n\_sync, oe\_n\_sync:** Synchronized signals for CDC.

## Conclusion

This enhanced SRAM controller integrates multiple advanced features to ensure robust, secure, and efficient memory operations. With error detection and correction, , burst mode, power management, security, byte addressable access and clock domain crossing, it is well-suited for complex and high-performance applications.