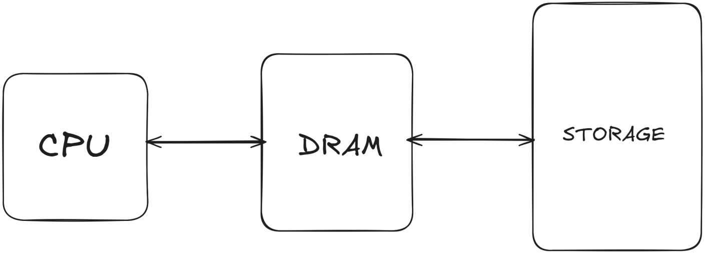
# SDR, DDR & LPDDR

## Evolution of Synchronous DRAM Interfaces

Pranav M – PES2UG23EC076
Lalith – PES2UG23EC074
Harshini – PES2UG23EC058

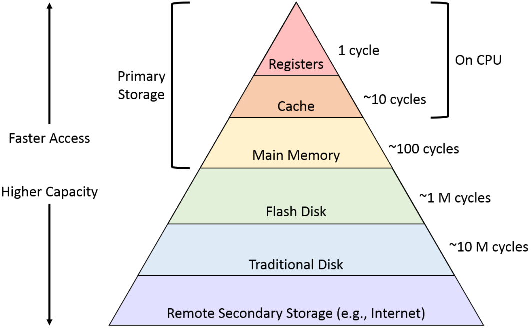
# Where DRAM sits in a system

- The CPU executes instructions and requests data continuously
- DRAM acts as the main working memory for active programs and data
- Storage is used for long-term data and is accessed much less frequently



# Why DRAM exists

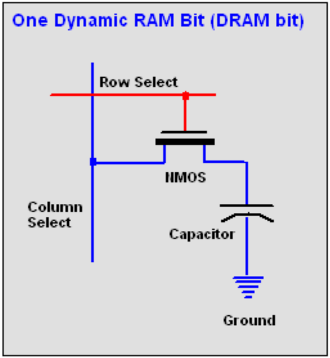
- Accessing storage directly is too slow for CPU execution
- Cache memory is fast but limited in size and expensive
- DRAM provides a balance between speed, capacity, and cost



The Memory Hierarchy

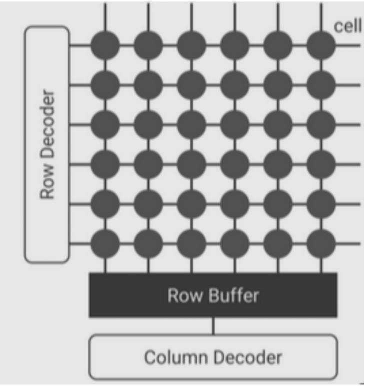
# What is SDRAM

- SDRAM stands for Dynamic Random Access Memory
- All operations are synchronized to an external clock
- Data is stored as charge in capacitor-based memory cells



# Working of SDRAM

- A row is activated, loading data into the sense amplifiers
- A column is selected to perform a read or write operation
- The row is precharged before accessing the next row



# Read and Write Operations in SDRAM

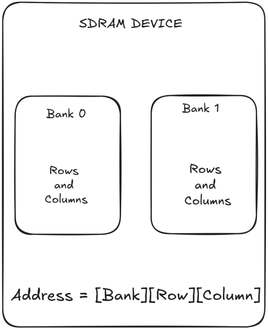
- Writing stores charge in a capacitor through an access transistor
- Reading transfers the stored charge to the bitline
- Sense amplifiers detect and amplify very small voltage changes
- Read operations are destructive and require immediate restoration

# Refresh Operation in SDRAM

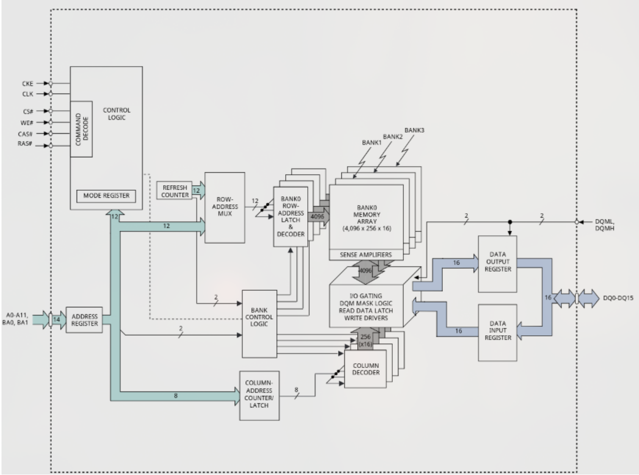
- Charge stored in capacitors leaks over time
- SDRAM periodically refreshes all rows to retain data
- Refresh internally reads and restores stored charge
- Typical refresh interval is around 64 ms

# Memory Organization in SDRAM

- Unlike SRAM, SDRAM is not a simple linear array
- Memory addresses are divided into bank, row, and column fields
- Address format: [Bank][Row][Column]
- This organization reflects the physical layout of DRAM cells

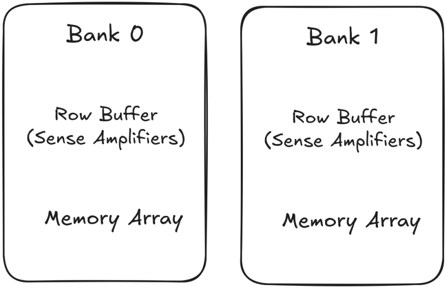


# SDRAM Internal Block Diagram (Architecture View)

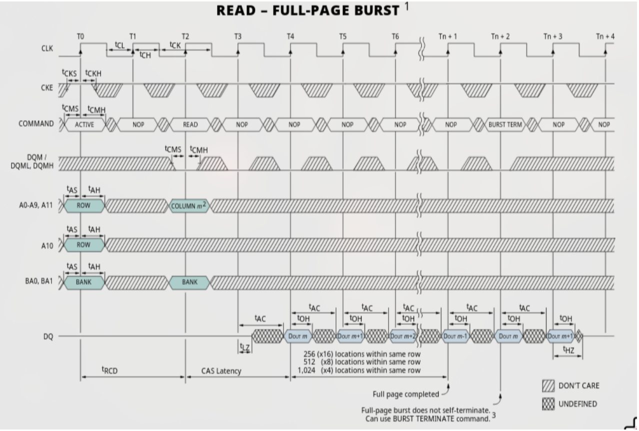


# Why Banks, Rows, and Columns?

- DRAM cells are arranged as a 2D array for efficient fabrication
- Entire rows are sensed together using shared sense amplifiers
- Keeping a row open enables fast repeated accesses (row buffer hits)
- Multiple banks allow parallel operations and hide memory latency

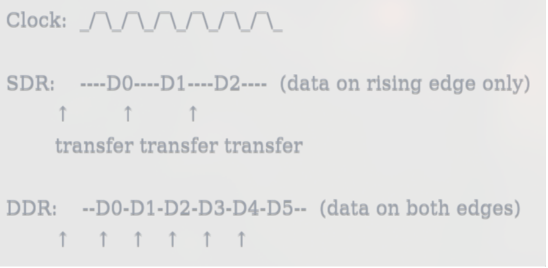


# SDRAM Timing: ACT → READ → PRE



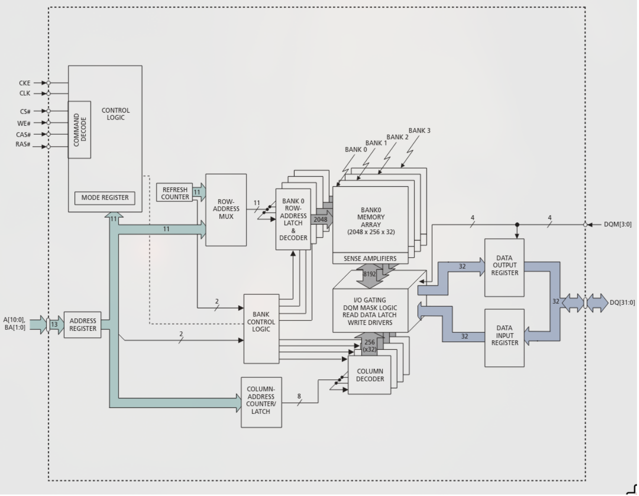
# SDR (Single Data Rate)

- Transfers one data word per clock cycle
- Data is sampled only on the rising edge of the clock
- Memory bandwidth scales linearly with clock frequency (one transfer per cycle)
- Row and column commands are synchronized to the clock
- Internal DRAM core operates asynchronously but is synchronized at the interface



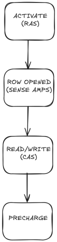
```
Clock: _/‾\_/‾\_/‾\_/‾\_/‾\_/‾\_

SDR:   ----D0----D1----D2----  (data on rising edge only)
       ↑         ↑         ↑
       transfer transfer transfer

DDR:   --D0-D1-D2-D3-D4-D5--  (data on both edges)
       ↑    ↑    ↑    ↑    ↑    ↑
```

# SDR SDRAM Internal Block Diagram (Architecture View)



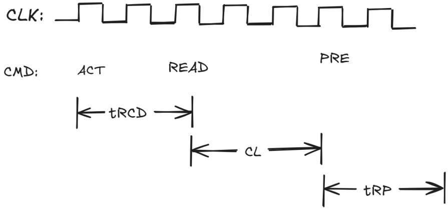
# SDR Key Operations

- RAS command activates a row in the selected bank
- Activated row data is latched into sense amplifiers (row buffer)
- Column READ or WRITE accesses data from the active row buffer
- Precharge closes the row before another row can be activated



# SDR Timing Parameters and Commands

- tRCD: Delay between row activation and column access
- tRP: Time required to precharge before next activation
- CL (CAS Latency): Cycles from READ command to data output
- All operations are constrained by strict timing parameters

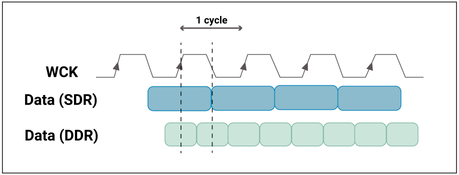


# Why Use SDR SDRAM?

- Cost-effective, high-density memory compared to SRAM
- Sufficient bandwidth for embedded and legacy systems
- Well-standardized with simple controller design
- Serves as a foundation for understanding DDR technologies

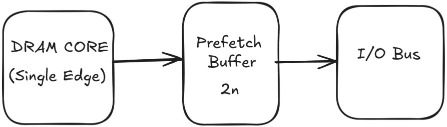
# DDR (Double Data Rate)

- Transfers two data words per clock cycle
- Data is sampled on both rising and falling clock edges
- Achieves higher bandwidth without increasing clock frequency



# DDR Architectural Upgrade: 2n Prefetch

- DRAM core operates at single-edge (SDR-like) speed
- Prefetch buffer reads multiple data words per access
- I/O interface transmits data on both clock edges
- Enables high bandwidth without faster memory cells

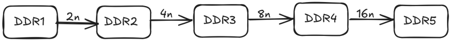


# Clocking and Data Capture in DDR

- Uses differential clock signals (CK and CK#)
- Introduces Data Strobe (DQS) for precise data capture
- Delay-Locked Loop (DLL) aligns internal and external timing
- Improves signal integrity at higher operating frequencies

# DDR Generations and Scaling

- Prefetch depth increases across generations
- External I/O bandwidth scales faster than core speed
- Bank groups introduced to sustain high throughput
- Operating voltage reduces with each generation



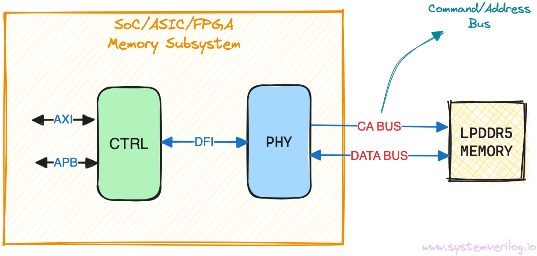
Bandwidth ↑, Voltage ↓ and Prefetch →

# Advantages and Trade-offs of DDR

- Doubles bandwidth without increasing clock frequency
- Prefetch and bank interleaving improve throughput
- Efficient for sequential access patterns
- Trade-off: higher latency for random accesses

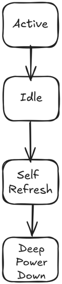
# LPDDR: Low Power DDR

- Designed for mobile and battery-powered devices
- Prioritizes power efficiency over peak performance
- Based on DDR architecture with aggressive power optimizations
- Widely used with SoCs in phones, tablets, and embedded systems



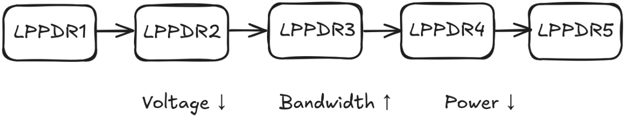
# Power-Saving Techniques in LPDDR

- Operates at significantly lower core and I/O voltages
- Supports multiple low-power and deep sleep states
- Self-refresh and deep power-down reduce idle power
- Optimized refresh and clock gating minimize energy usage



# Evolution of LPDDR

- LPDDR generations steadily reduce operating voltage
- Bandwidth increases while maintaining power efficiency
- Prefetch depth increases across generations
- New features added for reliability and energy savings



# SDR vs DDR vs LPDDR

- SDR: Simple design with limited bandwidth
- DDR: High performance through double-edge transfers
- LPDDR: Optimized for low power and mobile systems
- Choice depends on performance, power, and system needs

# Key Takeaways

- All DRAM types share the same core memory principles
- SDR, DDR, and LPDDR differ in interface and optimization goals
- Modern systems choose memory based on performance and power needs