DRAM Circuit and Architecture Basics

- Overview
- Terminology
- Access Protocol
- Architecture

![Diagram of DRAM circuit elements](image-url)

- Storage element (capacitor)
- Word Line
- Bit Line
- Switching element
DRAM Circuit Basics

DRAM Cell

Storage element (capacitor)

Word Line

Bit Line

Switching element

DRAM

Column Decoder

Sense Amps

... Bit Lines...

Memory Array

Data In/Out Buffers

Row Decoder

... Word Lines...
DRAM Circuit Basics

“Row” Defined

Row Size: 8 Kb @ 256 Mb SDRAM node
4 Kb @ 256 Mb RDRAM node
DRAM Circuit Basics

Sense Amplifier II: Precharged

- Precharged to $V_{cc}/2$
- $V_{cc}$ (logic 1)
- Gnd (logic 0)
- $V_{cc}/2$
DRAM Circuit Basics

Sense Amplifier III: Destructive Read

1. \(V_{cc}\) (logic 1)
2. \(V_{cc}/2\)
3. Gnd (logic 0)
4. Sense and Amplify
5. Wordline Driven
6. \(V_{cc}/2\)
DRAM Access Protocol

ROW ACCESS

AKA: OPEN a DRAM Page/Row
or
ACT (Activate a DRAM Page/Row)
or
RAS (Row Address Strobe)
DRAM Circuit Basics

“Column” Defined

Column: Smallest addressable quantity of DRAM on chip

SDRAM*: column size == chip data bus width (4, 8, 16, 32)
RDRAM: column size != chip data bus width (128 bit fixed)

SDRAM*: get “n” columns per access. n = (1, 2, 4, 8)
RDRAM: get 1 column per access.

* SDRAM means SDRAM and variants. i.e. DDR SDRAM
DRAM Access Protocol

COLUMN ACCESS I

READ Command
or
CAS: Column Address Strobe
then the data is valid on the data bus ... depending on what you are using for in/out buffers, you might be able to overlap a little or a lot of the data transfer with the next CAS to the same page (this is PAGE MODE).

note: page mode enables overlap with CAS
DRAM “Speed” Part I

How fast can I move data from DRAM cell to sense amp?

RCD (Row Command Delay)
DRAM “Speed” Part II

How fast can I get data out of sense amps back into memory controller?

\[ t_{\text{CAS}} \text{ aka } t_{\text{CASL}} \text{ aka } t_{\text{CL}} \]

**CAS**: Column Address Strobe

**CASL**: Column Address Strobe Latency

**CL**: Column Address Strobe Latency
How fast can I move data from DRAM cell into memory controller?

\[ t_{\text{RAC}} = t_{\text{RCD}} + t_{\text{CAS}} \]

RAC (Random Access Delay)
DRAM “Speed” Part IV

How fast can I precharge DRAM array so I can engage another RAS?

$\text{RP}$ (Row Precharge Delay)
DRAM “Speed” Part V

How fast can I read from different rows?

\[ t_{RC} = t_{RAS} + t_{RP} \]

**RC** (Row Cycle Time)
DRAM “Speed” Summary I

What do I care about?

- $t_{RCD}$
- $t_{CAS}$
- $t_{RP}$
- $t_{RC} = t_{RAS} + t_{RP}$
- $t_{RAC} = t_{RCD} + t_{CAS}$

- Seen in ads.
  - Easy to explain
  - Easy to sell
- Embedded systems designers
  - DRAM manufacturers
- Computer Architect:
  - Latency bound code
  - i.e. linked list traversal

**Definitions:**
- **RAS:** Row Address Strobe
- **CAS:** Column Address Strobe
- **RCD:** Row Command Delay
- **RAC:** Random Access Delay
- **RP:** Row Precharge Delay
- **RC:** Row Cycle Time
## DRAM “Speed” Summary II

<table>
<thead>
<tr>
<th>DRAM Type</th>
<th>Frequency</th>
<th>Data Bus Width (per chip)</th>
<th>Peak Data Bandwidth (per Chip)</th>
<th>Random Access Time (t&lt;sub&gt;RAC&lt;/sub&gt;)</th>
<th>Row Cycle Time (t&lt;sub&gt;RC&lt;/sub&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC133 SDRAM</td>
<td>133</td>
<td>16</td>
<td>200 MB/s</td>
<td>45 ns</td>
<td>60 ns</td>
</tr>
<tr>
<td>DDR 266</td>
<td>133 * 2</td>
<td>16</td>
<td>532 MB/s</td>
<td>45 ns</td>
<td>60 ns</td>
</tr>
<tr>
<td>PC800 RDRAM</td>
<td>400 * 2</td>
<td>16</td>
<td>1.6 GB/s</td>
<td>60 ns</td>
<td>70 ns</td>
</tr>
<tr>
<td>FCRAM</td>
<td>200 * 2</td>
<td>16</td>
<td>0.8 GB/s</td>
<td>25 ns</td>
<td>25 ns</td>
</tr>
<tr>
<td>RLDRAM</td>
<td>300 * 2</td>
<td>32</td>
<td>2.4 GB/s</td>
<td>25 ns</td>
<td>25 ns</td>
</tr>
</tbody>
</table>

DRAM is “slow”
But doesn’t have to be
\[ t_{RC} < 10\text{ns} \] achievable

Higher die cost → Not adopted in standard

Not commodity → Expensive
“DRAM latency”

A: Transaction request may be delayed in Queue
B: Transaction request sent to Memory Controller
C: Transaction converted to Command Sequences (may be queued)
D: Command/s Sent to DRAM
E₁: Requires only a CAS or
E₂: Requires RAS + CAS or
E₃: Requires PRE + RAS + CAS
F: Transaction sent back to CPU

“DRAM Latency” = A + B + C + D + E + F
DRAM Architecture Basics

PHYSICAL ORGANIZATION

This is per bank …
Typical DRAMs have 2+ banks
let's look at the interface another way .. the say the data sheets portray it.

main point: the RAS and CAS signals directly control the latches that hold the row and column addresses ...

DRAM Architecture Basics

Read Timing for Conventional DRAM
since DRAM’s inception, there have been a stream of changes to the design, from FPM to EDO to Burst EDO to SDRAM. The changes are largely structural modifications that target throughput.

Every change up to and including SDRAM has been relatively inexpensive, especially when considering the pay-off. FPM was essentially free, EDO cost a small premium, and Burst EDO and SDRAM each cost a moderate premium. However, all changes are considered expensive compared to the benefits they offer, and thus there is no consensus on new directions and myriad of choices has appeared.

### DRAM Evolutionary Tree

- **Conventional DRAM**
- **(Mostly) Structural Modifications Targeting Throughput**
  - FPM
  - EDO
  - P/BEDO
  - SDRAM
  - ESDRAM
- **Interface Modifications Targeting Throughput**
  - Rambus, DDR/2
  - Future Trends

### Structural Modifications Targeting Latency
- MOSYS
- FCRAM
- VCDRAM
DRAM Evolution

Read Timing for Conventional DRAM

- Row Access
- Column Access
- Transfer Overlap
- Data Transfer
DRAM Evolution

Read Timing for Fast Page Mode

Row Access
Column Access
Transfer Overlap
Data Transfer
solution to that problem -- instead of simple tri-state buffers, use a latch as well. By putting a latch after the column mux, the next column address command can begin sooner.
DRAM Evolution

Read Timing for Burst EDO

![Diagram showing DRAM read timing for Burst EDO with RAS (Row Access) and CAS (Column Access) signals, along with Address and Data lines, illustrating the overlap and transfer of valid data.]

- Row Access
- Column Access
- Transfer Overlap
- Data Transfer
**DRAM Evolution**

**Read Timing for Pipeline Burst EDO**

<table>
<thead>
<tr>
<th>RAS</th>
<th>CAS</th>
<th>Address</th>
<th>Column Address</th>
<th>DQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Row Access</td>
<td>Column Access</td>
<td>Valid Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transfer Overlap</td>
<td>Data Transfer</td>
<td>Valid Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Data</td>
<td>Valid Data</td>
<td>Valid Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Data</td>
<td>Valid Data</td>
<td>Valid Data</td>
</tr>
</tbody>
</table>
The main benefit of using DRAM memory is that it frees up the CPU or memory controller from having to control the DRAM's internal latches directly. Even though the time-to-first-word latency actually gets worse, the scheme increases system throughput.

**DRAM Evolution**

**Read Timing for Synchronous DRAM**

(RAS + CAS + OE ... == Command Bus)
DRAM Evolution

Inter-Row Read Timing for ESDRAM

“Regular” CAS-2 SDRAM, R/R to same bank

ESDRAM, R/R to same bank
DRAM Evolution

Write-Around in ESDRAM

“Regular” CAS-2 SDRAM, R/W/R to same bank, rows 0/1/0

ESDRAM, R/W/R to same bank, rows 0/1/0

(can second READ be this aggressive?)
Segment cache is software-managed, reduces energy
FCRAM opts to break up the data array. Only activate a portion of the word line.

8K rows requires 13 bits to select...

FCRAM uses 15 (assuming the array is 8k x 1k... the data sheet does not specify)

---

**Internal Structure of Fast Cycle RAM**

**SDRAM**

- 8M Array (8Kr x 1Kb)
- Sense Amps
- t\_RCD = 15ns (two clocks)

**FCRAM**

- 8M Array (?)
- Sense Amps
- t\_RCD = 5ns (one clock)

Reduces access time and energy/access
MoSys takes this one step further ... DRAM with an SRAM interface & speed but DRAM energy

[physical partitioning: 72 banks] auto refresh -- how to do this transparently? the logic moves through the arrays, refreshing them when not active. but what if one bank gets repeated access for a long duration? all other banks will be refreshed, but that one will not.

solution: they have a bank-sized CACHE of lines ... in theory, should never have a problem (magic)