SDRAM Memory Controller

- Static RAM Technology
  - 6T Memory Cell
  - Memory Access Timing

- Dynamic RAM Technology
  - 1T Memory Cell
  - Memory Access Timing

Tri-State Gates

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
<th>OE_L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Truth Table:

<table>
<thead>
<tr>
<th>X</th>
<th>OE_L</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Z</td>
</tr>
</tbody>
</table>

Truth Table:

<table>
<thead>
<tr>
<th>X</th>
<th>OE_L</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Z</td>
</tr>
</tbody>
</table>

Truth Table:

<table>
<thead>
<tr>
<th>X</th>
<th>OE_L</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Z</td>
</tr>
</tbody>
</table>

Truth Table:
Slick Multiplexer Implementation

2:1 Multiplexer

<table>
<thead>
<tr>
<th>S</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IN₀</td>
</tr>
<tr>
<td>1</td>
<td>IN₁</td>
</tr>
</tbody>
</table>

Basic Memory Subsystem Block Diagram

- **Address Decoder**
- **Word Line**
- **Memory cell**
- **2^n word lines**

what happens if n and/or m is very large?
**Static RAM Cell**

6-Transistor SRAM Cell

- **Write:**
  1. Drive bit lines (bit=1, bit=0)
  2. Select row
- **Read:**
  1. Precharge bit and bit to Vdd or Vdd/2 => make sure equal!
  2. Select row
  3. Cell pulls one line low
  4. Sense amp on column detects difference between bit and bit

**Typical SRAM Organization: 16-word x 4-bit**
Logic Diagram of a Typical SRAM

- Write Enable is usually active low (WE_L)
- Din and Dout are combined to save pins:
  - A new control signal, output enable (OE_L) is needed
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
  - Both WE_L and OE_L are asserted:
    - Result is unknown. Don’t do that!!!

Typical SRAM Timing

OE determines direction
Hi = Write, Lo = Read
Writes are dangerous! Be careful!
Double signaling: OE Hi, WE Lo

Write Timing:
- Data In
- Write Address
- OE_L
- WE_L
- Write Setup Time
- Write Hold Time

Read Timing:
- Data Out
- Read Address
- Read Access Time

A/N
WE_L
OE_L
D
2^N\text{words} \times \text{M bit}
SRAM
Problems with SRAM

- Six transistors use up lots of area
- Consider a “Zero” is stored in the cell:
  - Transistor N1 will try to pull “bit” to 0
  - Transistor P2 will try to pull “bit bar” to 1
- Bit lines are already pre-charged high: Are P1 and P2 really necessary?

1-Transistor Memory Cell (DRAM)

- Write:
  1. Drive bit line
  2. Select row
- Read:
  1. Precharge bit line to Vdd/2
  2. Select row
  3. Cell and bit line share charges
     - Minute voltage changes on the bit line
  4. Sense (fancy sense amp)
     - Can detect changes of ~1 million electrons
  5. Write: restore the value
- Refresh
  1. Just do a dummy read to every cell

Read is really a read followed by a restoring write
Classical DRAM Organization (Square)

- RAM Cell Array
- Column Selector & I/O Circuits
- Word (row) select
- Column Address
- Row and Column Address together:
  - Select 1 bit at a time

Square keeps the wires short:
- Power and speed advantages
- Less RC, faster precharge and discharge is faster access time!

DRAM Logical Organization (4 Mbit)

- 4 Mbit = 22 address bits
- 11 row address bits
- 11 col address bits
- 11
- Column Decoder
- Sense Amps & I/O
- Memory Array (2,048 x 2,048)
- Word Line
- Storage Cell

Square root of bits per RAS/CAS:
- Row selects 1 row of 2048 bits from 2048 rows
- Col selects 1 bit out of 2048 bits in such a row
Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low

Din and Dout are combined (D):
- WE_L is asserted (Low), OE_L is disasserted (High)
  - D serves as the data input pin
- WE_L is disasserted (High), OE_L is asserted (Low)
  - D is the data output pin

Row and column addresses share the same pins (A)
- RAS_L goes low: Pins A are latched in as row address
- CAS_L goes low: Pins A are latched in as column address
- RAS/CAS edge-sensitive

Logic Diagram of a Typical DRAM

DRAM READ Timing

Every DRAM access begins at:
- Assertion of the RAS_L
- 2 ways to read: early or late v. CAS

Early Read Cycle: OE_L asserted before CAS_L
Late Read Cycle: OE_L asserted after CAS_L
Early Read Sequencing

- Assert Row Address
- Assert RAS_L
  - Commence read cycle
  - Meet Row Addr setup time before RAS/hold time after RAS
- Assert OE_L
- Assert Col Address
- Assert CAS_L
  - Meet Col Addr setup time before CAS/hold time after CAS
- Valid Data Out after access time
- Disassert OE_L, CAS_L, RAS_L to end cycle

Sketch of Early Read FSM

```
FSM Clock?

Row Address to Memory
  ─► Assert RAS_L
  │    Setup time met?
  ─► Assert OE_L, RAS_L
  │    Hold time met?
  │    Col Address to Memory
  │    Setup time met?
  ─► Assert OE_L, RAS_L, CAS_L
  │    Hold time met?
  ─► Assert OE_L, RAS_L, CAS_L
  │    Data Available (better grab it!)
```

CS 150 - Fall 2005 – Lec #11: Memory Controller - 15
Late Read Sequencing

- Assert Row Address
- Assert RAS_L
  - Commence read cycle
  - Meet Row Addr setup time before RAS/hold time after RAS
- Assert Col Address
- Assert CAS_L
  - Meet Col Addr setup time before CAS/hold time after CAS
- Assert OE_L
- Valid Data Out after access time
- Disassert OE_L, CAS_L, RAS_L to end cycle

Sketch of Late Read FSM

FSM Clock?

Row Address to Memory
  - Setup time met?
  
  Assert RAS_L
    - Hold time met?

Col Address to Memory
  - Setup time met?
  
  Assert RAS_L
    - Hold time met?

Assert OE_L, RAS_L, CAS_L

Data Available (better grab it!)
DRAM WRITE Timing

- Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to write: early or late v. CAS

Key DRAM Timing Parameters

- \( t_{RAC} \): minimum time from RAS line falling to the valid data output.
  - Quoted as the speed of a DRAM
  - A fast 4Mb DRAM \( t_{RAC} = 60 \text{ ns} \)
- \( t_{RAC} \): minimum time from the start of one row access to the start of the next.
  - \( t_{RAC} = 110 \text{ ns} \) for a 4Mbit DRAM with a \( t_{RAC} \) of 60 ns
- \( t_{CAC} \): minimum time from CAS line falling to valid data output.
  - 15 ns for a 4Mbit DRAM with a \( t_{RAC} \) of 60 ns
- \( t_{PC} \): minimum time from the start of one column access to the start of the next.
  - 35 ns for a 4Mb DRAM with a \( t_{RAC} \) of 60 ns

Cs 150 - Fall 2005 - Lec #11: Memory Controller - 20
SDRAM Memory Controller

- Static RAM Technology
  - 6T Memory Cell
  - Memory Access Timing

- Dynamic RAM Technology
  - 1T Memory Cell
  - Memory Access Timing