Lecture 24: Peripheral Memory Circuits

Review: Read-Write Memories (RAMs)

- **Static – SRAM**
  - data is stored as long as supply is applied
  - large cells (6 fets/cell) – so fewer bits/chip
  - fast – so used where speed is important (e.g., caches)
  - differential outputs (output BL and !BL)
  - use sense amps for performance
  - compatible with CMOS technology

- **Dynamic – DRAM**
  - periodic refresh required (every 1 to 4 ms) to compensate for the charge loss caused by leakage
  - small cells (1 to 3 fets/cell) – so more bits/chip
  - slower – so used for main memories
  - single ended output (output BL only)
  - need sense amps for correct operation
  - not typically compatible with CMOS technology
Non-Volatile Memories
The Floating-gate transistor (FAMOS)

Device cross-section

Schematic symbol
Floating-Gate Transistor Programming

Avalanche injection

Removing programming voltage leaves charge trapped

Programming results in higher $V_T$. 

Sp11 CMPEN 411 L24 S.4
A “Programmable-Threshold” Transistor

The diagram illustrates the relationship between drain current ($I_D$) and gate-source voltage ($V_{GS}$) for a transistor in two states: "0"-state and "1"-state. The "ON" state is indicated by a point on the graph, and the "OFF" state is represented by another point. The difference between these states, $DV_T$, is shown, indicating the threshold voltage ($V_{TH}$) of the transistor.

The graph shows the drain current ($I_D$) on the vertical axis and the gate-source voltage ($V_{GS}$) on the horizontal axis. The points labeled "ON" and "OFF" correspond to specific values of $V_{GS}$ and $I_D$, illustrating the transistor's operation in these states.
Peripheral Memory Circuitry

- Row and column decoders
- Read bit line precharge logic
- Sense amplifiers
- Timing and control
- Speed
- Power consumption
- Area – pitch matching
Row Decoders

- Collection of $2^M$ complex logic gates organized in a regular, dense fashion

- (N)AND decoder for 8 address bits
  \[
  WL(0) = \overline{A_7} \& \overline{A_6} \& \overline{A_5} \& \overline{A_4} \& \overline{A_3} \& \overline{A_2} \& \overline{A_1} \& \overline{A_0}
  \]
  \[
  \ldots
  \]
  \[
  WL(255) = A_7 \& A_6 \& A_5 \& A_4 \& A_3 \& A_2 \& A_1 \& A_0
  \]

- NOR decoder for 8 address bits
  \[
  WL(0) = !(A_7 \| A_6 \| A_5 \| A_4 \| A_3 \| A_2 \| A_1 \| A_0)
  \]
  \[
  \ldots
  \]
  \[
  WL(255) = !(A_7 \| A_6 \| A_5 \| A_4 \| A_3 \| A_2 \| A_1 \| A_0)
  \]

- Goals: Pitch matched, fast, low power
Implementing a Wide NOR Function

- Single stage 8x256 bit decoder (as in Lecture 22)
  - One 8 input NOR gate per row x 256 rows = 256 x (8+8) = 4,096
  - Pitch match and speed/power issues

- Decompose logic into multiple levels
  
  ![WL(0) = !(!(A_7 | A_6) & !(A_5 | A_4) & !(A_3 | A_2) & !(A_1 | A_0))](image)
  
  - First level is the predecoder (for each pair of address bits, form \( A_i | A_{i-1}, A_i !A_{i-1}, A_i !A_i, \) and \( A_i !A_{i-1} \))
  - Second level is the word line driver

- Predecoders reduce the number of transistors required
  
  - Four sets of four 2-bit NOR predecoders = 4 x 4 x (2+2) = 64
  - 256 word line drivers, each a four input NAND – 256 x (4+4) = 2,048
    - 4,096 vs 2,112 = almost a 50% savings

- Number of inputs to the gates driving the WLs is halved, so the propagation delay is reduced by a factor of \(~4\)
Hierarchical Decoders

Multi-stage implementation improves performance

NAND decoder using 2-input pre-decoders
Dynamic Decoders

Precharge devices

2-input NOR decoder

2-input NAND decoder

Which one is faster? Smaller? Low power?
Pass Transistor Based Column Decoder

- **Read**: connect BLs to the Sense Amps (SA)
  - drive one of the BLs low to write a 0 into the cell

- **Writes**: fast since there is only one transistor in the signal path. However, there is a large transistor count \((K+1)2^K + 2 \times 2^K\)

- For \(K = 2\) \(\rightarrow 3 \times 2^2 \text{ (decoder)} + 2 \times 2^2 \text{ (PTs)} = 12 + 8 = 20\)
Number of transistors reduced to \((2 \times 2 \times (2^K - 1))\)

- for \(K = 2 \rightarrow 2 \times 2 \times (2^2 - 1) = 4 \times 3 = 12\)

Delay increases quadratically with the number of sections \((K)\)

- so prohibitive for large decoders

- can fix with buffers, progressive sizing, combination of tree and pass transistor approaches
Consider a memory with 10b address and 8b data

<table>
<thead>
<tr>
<th>Conf.</th>
<th>Data/Row</th>
<th>Row Decoder</th>
<th>Column Decoder</th>
</tr>
</thead>
</table>
| 1D    | 8b       | 10b = a 10x2^{10} decoder  
Single stage = 20,480  
Two stage = 10,320 |            |
| 2D    | 32b      | 8b = 8x2^{8} decoder  
Single stage = 4,096 T  
Two stage = 2,112 T | 2b = 2x2^{2} decoder  
PT = 76 T  
Tree = 96 T |
|       | (32x256 core) | | |
| 2D    | 64b      | 7b = 7x2^{7} decoder  
Single stage = 1,792 T  
Two stage = 1,072 T | 3b = 3x2^{3} decoder  
PT = 160 T  
Tree = 224 T |
|       | (64x128 core) | | |
| 2D    | 128b     | 6b = 6x2^{6} decoder  
Single stage = 768 T  
Two stage = 432 T | 4b = 4x2^{4} decoder  
PT = 336 T  
Tree = 480 T |
|       | (128x64 core) | | |
First step of a Read cycle is to precharge (PC) the bit lines to $V_{DD}$
- every differential signal in the memory must be equalized to the same voltage level before Read

Turn off PC and enable the WL
- the grounded PMOS load limits the bit line swing (speeding up the next precharge cycle)

Equalization transistor - speeds up equalization of the two bit lines by allowing the capacitance and pull-up device of the nondischarged bit line to assist in precharging the discharged line
Sense Amplifiers

- Amplification – resolves data with small bit line swings (in some DRAMs required for proper functionality)

- Delay reduction – compensates for the limited drive capability of the memory cell to accelerate BL transition

\[ t_p = \frac{C \Delta V}{I_{av}} \]

- Power reduction – eliminates a large part of the power dissipation due to charging and discharging bit lines

- Signal restoration – for DRAMs, need to drive the bit lines full swing after sensing (read) to do data refresh
Classes of Sense Amplifiers

- Differential SA – takes small signal differential inputs (BL and !BL) and amplifies them to a large signal single-ended output
  - common-mode rejection – rejects noise that is equally injected to both inputs

- Only suitable for SRAMs (with BL and !BL)

- Types
  - Current mirroring
  - Two-stage
  - Latch based

- Single-ended SA – needed for DRAMs
Differential Sense Amplifier

Directly applicable to SRAMs
Differential Sensing — SRAM

(a) SRAM sensing scheme

(b) two stage differential amplifier
Read/Write Circuitry

D: data (write) bus
R: read bus
W: write signal
CS: column select
  (column decoder)

Local W (write):
  BL = D, !BL = !D enabled by W & CS
Local R (read):
  R = BL, !R = !BL enabled by !W & CS
Approaches to Memory Timing

SRAM Timing
Self-Timed

Address Bus

Address transition initiates memory operation

DRAM Timing
Multiplexed Addressing

Address Bus

Row Addr. Column Addr.

msb’s lsb’s

RAS CAS

RAS-CAS timing

Address transition initiates memory operation
Reliability and Yield

- Memories operate under low signal-to-noise conditions
  - word line to bit line coupling can vary substantially over the memory array
    - folded bit line architecture (routing BL and !BL next to each other ensures a closer match between parasitics and bit line capacitances)
  - interwire bit line to bit line coupling
    - transposed (or twisted) bit line architecture (turn the noise into a common-mode signal for the SA)
  - leakage (in DRAMs) requiring refresh operation

- suffer from low yield due to high density and structural defects
  - increase yield by using error correction (e.g., parity bits) and redundancy

- and are susceptible to soft errors due to alpha particles and cosmic rays
Redundancy in the Memory Structure

Fuse bank

Row address

Redundant row

Redundant columns

Column address
Row Redundancy

- Fused Repair Addresses
- Enable Normal Wordline Decoder
- Normal Wordline Decoder Enable
- Normal Wordline
- Redundant Wordline
- Redundant Wordline
- Redundant Wordline
- Redundant Wordline
- Redundant Wordline

Column Redundancy

Diagram showing column redundancy with normal data columns and a redundant data column connected via fuses to data lines 0 through 7.
Error-Correcting Codes

Example: Hamming Codes

\[ P_1 P_2 B_3 P_4 B_5 B_6 B_7 \]
\[ P_1 \oplus B_3 \oplus B_5 \oplus B_7 = 0 \]
\[ P_2 \oplus B_3 \oplus B_6 \oplus B_7 = 0 \]
\[ P_4 \oplus B_5 \oplus B_6 \oplus B_7 = 0 \]

\[ 2^k \geq m+k+1. \quad m \# \text{ data bit, } \quad k \# \text{ check bit} \]

For 64 data bits, needs 7 check bits

\text{e.g. If } B_3 \text{ flips}

\[
\begin{align*}
1 & \quad 1 \\
1 & \quad = 3 \\
0 & 
\end{align*}
\]
### Performance and area overhead for ECC

<table>
<thead>
<tr>
<th>Word Length</th>
<th>Number ECC Bits</th>
<th>Area Increase for ECC bits</th>
<th>Delay (EXOR-Gate Tree Depth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>5</td>
<td>31%</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>6</td>
<td>19%</td>
<td>5</td>
</tr>
<tr>
<td>64</td>
<td>7</td>
<td>11%</td>
<td>6</td>
</tr>
<tr>
<td>128</td>
<td>8</td>
<td>6%</td>
<td>7</td>
</tr>
<tr>
<td>256</td>
<td>9</td>
<td>3.5%</td>
<td>8</td>
</tr>
</tbody>
</table>
Redundancy and Error Correction
Soft Errors

- Nonrecurrent and nonpermanent errors from:
  - alpha particles (from the packaging materials)
  - neutrons from cosmic rays

- As feature size decreases, the charge stored at each node decreases (due to a lower node capacitance and lower $V_{DD}$) and thus $Q_{\text{critical}}$ (the charge necessary to cause a bit flip) decreases leading to an increase in the soft error rate (SER)

<table>
<thead>
<tr>
<th>From Actel</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTBF (hours)</td>
</tr>
<tr>
<td>.13 µm</td>
</tr>
<tr>
<td>Ground-based</td>
</tr>
<tr>
<td>Civilian Avionics System</td>
</tr>
<tr>
<td>Military Avionics System</td>
</tr>
</tbody>
</table>

From Semico Research Corp.
See class website for web links
CELL Processor!
### CELL Processor!

<table>
<thead>
<tr>
<th></th>
<th>Sony Emotion Engine</th>
<th>Cell Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU Core ISA</strong></td>
<td>MIP64</td>
<td>64-bit Power Architecture</td>
</tr>
<tr>
<td><strong>Core Issue Rate</strong></td>
<td>Dual</td>
<td>Dual</td>
</tr>
<tr>
<td><strong>Core Frequency</strong></td>
<td>300MHz</td>
<td>~4GHz (est.)</td>
</tr>
<tr>
<td><strong>Core Pipeline</strong></td>
<td>6 stages</td>
<td>21 stages</td>
</tr>
<tr>
<td><strong>Core L1 Cache</strong></td>
<td>16KB I-Cache + 8KB D-Cache</td>
<td>32KB I-Cache + 32KB D-Cache</td>
</tr>
<tr>
<td><strong>Core Additional Memory</strong></td>
<td>16KB scratch</td>
<td>512KB L2</td>
</tr>
<tr>
<td><strong>Vector Units</strong></td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td><strong>Vector Registers (#, width)</strong></td>
<td>32, 128-bit + 16, 16-bit</td>
<td>128, 128-bit</td>
</tr>
<tr>
<td><strong>Vector Local Memory</strong></td>
<td>4K/16KB I-Cache + 4K/16KB D-Cache</td>
<td>256KB unified</td>
</tr>
<tr>
<td><strong>Memory Bandwidth</strong></td>
<td>3.2GB/s peak</td>
<td>25.6GB/s peak (est.)</td>
</tr>
<tr>
<td><strong>Total Chip Peak FLOPS</strong></td>
<td>6.2GFLOPS</td>
<td>256GFLOPS</td>
</tr>
<tr>
<td><strong>Transistor Count</strong></td>
<td>10.5 million</td>
<td>235 million</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>15W @ 1.8V</td>
<td>~80W (est.)</td>
</tr>
<tr>
<td><strong>Die Size</strong></td>
<td>240mm²</td>
<td>235mm²</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>250nm, 4LM</td>
<td>90nm, 8LM + L1</td>
</tr>
</tbody>
</table>
Embedded SRAM (4.6GHz)

- Each SRAM cell 0.99um²
- Each block has 32 sub-arrays,
- Each sub-array has 128 WL plus 4 redundant lines, Each block has 2 redundant BL,
Multiplier in CELL
Next Lecture and Reminders

- Next lecture
  - Power consumption in datapaths and memories
    - Reading assignment – Rabaey, et al, 11.7; 12.5