Outline

- Memory Arrays
- SRAM Architecture
  - SRAM Cell
  - Decoders
  - Column Circuitry
- Serial Access Memories

- You should be reading the book!! Ch 12.
Array Architecture

- $2^n$ words of $2^m$ bits each
- Good regularity – easy to design
- Very high density if good cells are used
Array Architecture

- $2^n$ words of $2^m$ bits each
- Good regularity – easy to design
- Very high density if good cells are used
Array Architecture

- $2^n$ words of $2^m$ bits each
- Good regularity – easy to design
- Very high density if good cells are used
6T SRAM Cell

- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read:
  - Precharge BL, BL’
  - Raise WL
- Write:
  - Drive data onto BL, BL’
  - Raise WL
SRAM Read

- Precharge both bitlines high
- Then turn on wordline, WL
- One of the two bitlines will be pulled down by the cell
- Ex: \( A = 0, A_b = 1 \)
  - BL discharges, BL’ stays high
  - But A bumps up slightly
- Read stability
  - A must not flip
  - \( N1 >> N2 \)

![Diagram of SRAM Read](image-url)
CMOS SRAM Analysis (Read)

Assume 1 is stored (Q=1)

\[
\frac{k_{n,M5}}{2} \left( \frac{V_{DD}}{2} - V_{Tn} \left( \frac{V_{DD}}{2} \right) \right)^2 = k_{n,M1} \left( \left( V_{DD} - |V_{Tn}| \right) \frac{V_{DD}}{2} - \frac{V_{DD}^2}{8} \right)
\]

\[
(W/L)_{n,M5} \leq 10 (W/L)_{n,M1} \quad \text{(supercedes read constraint)}
\]
SRAM Write

- Drive one bitline high, the other low
  - Depending on write data
- Then turn on wordline, WL
- Bitlines overpower cell with new value
- Example: A = 1, A_b = 0, BL = 0, BL’ = 1
  - Force A_b high, then A falls low
- Writability
  - Must overpower feedback inverter
  - N4 >> P2
CMOS SRAM Analysis (Write)

Assume 1 is stored

Write 0 to cell

\[ k_{n,M6}\left(\left(\frac{V_{DD} - V_{Tn}}{2}\right) - \frac{V_{DD}^2}{8}\right) = k_{p,M4}\left(\left(\frac{V_{DD} - |V_{Tn}|}{2}\right) - \frac{V_{DD}^2}{8}\right) \]

\[ \frac{i_{n,M5}}{2}\left(\frac{V_{DD}}{2} - \frac{V_{DD}^2}{2}\right)^2 = k_{n,M1}\left(\left(\frac{V_{DD} - |V_{Tn}|}{2}\right) - \frac{V_{DD}^2}{8}\right) \]

\[ (W/L)_{n,M6} \geq 0.33 \ (W/L)_{p,M4} \]

\[ (W/L)_{n,M5} \geq 10 \ (W/L)_{n,M1} \]

Penn ESE 370 Fall 2015 - Khanna
SRAM Column Example

Read

Write
Array Architecture

- $2^n$ words of $2^m$ bits each
- Good regularity – easy to design
- Very high density if good cells are used
Decoders

- $n:2^n$ decoder consists of $2^n$ $n$-input AND gates
  - One needed for each row of memory
  - Build AND from NAND or NOR gates

Static CMOS
Large Decoders

- For $n > 4$, NAND gates become slow
  - Break large gates into multiple smaller gates
Predecoding

- Many of these gates are redundant
  - Factor out common gates into predecoder
  - Saves area
  - Same path effort
Array Architecture

- $2^n$ words of $2^m$ bits each
- Good regularity – easy to design
- Very high density if good cells are used
Column Circuitry

- Some circuitry is required for each column
  - Bitline conditioning
    - Precharging, input data to bitline
  - Sense amplifiers
  - Column multiplexing (talked about last time)
Bitline Conditioning

- Precharge bitlines high before reads

\[ \phi \]

BL  BL'
Bitline Conditioning

- Precharge bitlines high before reads

![Diagram of BL and BL']

- What if pre-charged to Vdd/2?
  - Pros: reduces read-upset
  - Challenge: generate Vdd/2 voltage on chip
Sense Amplifiers

- Bitlines have many cells attached
  - Ex: 32-kbit SRAM has 128 rows x 256 cols
  - 128 cells on each bitline
- $t_{pd} \propto (C/I) \Delta V$
  - Even with shared diffusion contacts, 64C of diffusion capacitance (big C)
  - Discharged slowly through small transistors (small I)
- Sense amplifiers are triggered on small voltage swing (reduce $\Delta V$)
Differential Pair Amp

- Differential pair requires no clock
- But always dissipates static power
Clocked Sense Amp

- Clocked sense amp saves power
- Requires sense_clk after enough bitline swing
- Isolation transistors cut off large bitline capacitance
Twisted Bitlines

- Sense amplifiers also amplify noise
  - Coupling noise is severe in modern processes
  - Try to couple equally onto bit and bit_b
  - Done by *twisting* bitlines

![Diagram showing twisted bitlines]
Serial Access Memories

- Serial access memories do not use an address
  - Shift Registers
  - Tapped Delay Lines
  - Serial In Parallel Out (SIPO)
  - Parallel In Serial Out (PISO)
  - Queues (FIFO, LIFO)
Shift Register

- *Shift registers* store and delay data
- Simple design: cascade of registers
Denser Shift Registers

- Flip-flops aren’t very area-efficient
- For large shift registers, keep data in SRAM instead
- Move read/write pointers to RAM rather than data
  - Initialize read address to first entry, write to last
  - Increment address on each cycle
Serial In Parallel Out

- 1-bit shift register reads in serial data
  - After N steps, presents N-bit parallel output
Parallel In Serial Out

- Load all N bits in parallel when $\text{shift} = 0$
  - Then shift one bit out per cycle
Queues

- Queues allow data to be read and written at different rates.
- Read and write each use their own clock, data
- Queue indicates whether it is full or empty
- Build with SRAM and read/write counters (pointers)
FIFO, LIFO Queues

- **First In First Out (FIFO)**
  - Initialize read and write pointers to first element
  - Queue is EMPTY
  - On write, increment write pointer
  - If write almost catches read, Queue is FULL
  - On read, increment read pointer

- **Last In First Out (LIFO)**
  - Also called a *stack*
  - Use a single *stack pointer* for read and write
Idea

- Memory for compact state storage
  - Minimize area per bit → maximize density
  - Requires careful sizing
- Share circuitry across many bits
  - Precharge, Amplifiers
- Serial address memories
  - Use pointers to access memory
    - Eg. FIFO queue
Admin

- Homework 7 due Wednesday
- Shuffling of assignment dates to give you more time
  - HW 8 due 11/23 (Lab summary and write up)
  - Proj 2.M due 11/25 (Wednesday before Thanksgiving)
  - Proj 2 due 12/2