Design Example: High-Performance SRAM

C.K. Ken Yang
UCLA
yang@ee.ucla.edu
Courtesy of BA, MAH
Overview

• **Reading**
  – Rabaey 12.3, 12.5
  – Papers: Mori, Amrutur

• **Introduction**

  The power consumed in SRAMs can be a significant portion of the overall power for digital chips. This is especially important in portable applications where low power is a primary design target.

  This lecture discusses a case study on a fast and low-power SRAM. This design employs many of the previously discussed techniques to reduce power. We also introduced generating appropriate timing signals to optimize the sensing delay and reduce power known as replica sensing and bit-line clamping. It also employs some adiabatic techniques such as low-swing logic gates reduce power in long wires of the decoder blocks and the important issues associated with such designs.
Low-Power SRAM

- 2K x 16b SRAM, arranged as 32 64 x 16b blocks
- 0.25μm dual-V_{th} 3-metal CMOS process
- Active power dissipation of 0.9mW at 100MHz, 1V Vdd
- Low power through half-swing signaling and charge-recycling
Chip Floorplan

- Blocks arranged in quadrants
- Only the accessed quadrant’s GWL fires
- Only the accessed block’s LWL fires
Decoder Logical Structure

- Predecoder output and GWL are half-swing lines
  - Reduces power required to drive long wires (large capacitive loading)
- Block select and LWL are full swing
- Block select drives gate input of half-swing AND gate
  - Want nMOS to turn off hard for timing accuracy
High-Performance SRAM

• Low-swing gates for decoders
  – Charge recycling
  – Designed in a low-power process
    • Low $V_{th}$ devices
    • High $V_{th} = \frac{1}{2} Vdd = 0.5V$

• SRAM clocking

• Half-swing bitlines

• Results and power analysis
Lower L.E. Decode Stage

- Let’s first take a look at a common technique used to improve the logical effort for a NAND gate in the decoder.

- Assuming pulsed inputs, all inputs to these gates start low.
- C is a large fanout node and is driven by a big device.
- Pull off and merge all the bottom devices into one that is n times bigger.
- Only one output of NAND falls → drive strength of C n times bigger, but current is < n times; NAND has LE of 2 input gate.
Low-Swing Gates

• Why low swing gates?
  – Look at the basic CMOS equations:
    \[ \Delta t = \frac{C \Delta V}{i} \quad \text{Energy} = CV_{\text{Supply}} \Delta V \]
    – Making \( \Delta V \) small reduces the delay and energy

• Problems:
  – Converting low-swing to high-swing
    • Nothing comes for free – usually takes time and/or energy
    • Can view as part of logical effort
  – Effective Fanout = L.E. \( C_{\text{load}} \Delta V_{\text{out}} / C_{\text{in}} \Delta V_{\text{in}} \)
Half-Vdd Gate

- Inputs are half-swing pulses
  - Gate input swings from Vdd/2 to Vdd
  - Source input swings from Vdd/2 to Gnd
  - No current penalty for swing in asserted state! (A=Vdd, B=Gnd)
- Output swings full rail
- Vdd = $1V = 2 \times V_{thH}$
  - Important since nMOS needs to turn off when A is Vdd/2
- Pulsed inputs are necessary due to leakage in nMOS

### Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Vdd</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Vdd</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Vdd</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Noise in Half-Vdd Gate

- Two half-select cases when the gate is more sensitive to noise
  - First case: $A = V_{dd}/2$, $B = \text{Gnd}$
    - Output is Vdd
    - nMOS is on the verge of turning on
    - Requires high switching point $\rightarrow$ lower noise margin

![Diagram of half-Vdd gate with A swept from 0 to Vdd and B grounded.](image)
Noise in Half-Vdd Gate

- Second half-select case: $A = V_{dd}$, $B = V_{dd}/2$
  - Output = $V_{dd}$
  - pMOS device off!
  - nMOS device leaking
  - Cannot have static inputs $\rightarrow$ pulsed inputs

\[ A = V_{DD} \]
\[ B = \text{swept} \]

![Graph](A = V_{dd}, B swept from 0 to Vdd)
Half-Swing Pulse-Mode Gate with Leaker

- High-Vth pMOS leaker reduces incorrect firing probability
  - Improves noise margin
- Still have leakage current problem
  - But only when pulses are high (short time)
Generating High-to-Low Swing Signals

- Lower speeds due to lower input drive
  - Even with Vdd input since nMOS source is at Vdd/2 and body effected
  - PN sizing ratio closer to 1 for (a)
  - Need >3 PN sizing ratio for (b)
  - Use low Vth transistors where needed
Predecoder Gate

- Generates half-swing pulse output (either positive or negative)
- Self-resetting using simple inverter chain
Block Decoder Gate

- Self resetting delay element is the replica bitline
- Block select pulse width is set by the replica bitline
- Controls the pulse width of the LWL thru the local row decoder
Local/Global Row Decoder Gate

- Simple gate relies on the inputs to start the gate reset
- Reset through the small low-Vt PMOS devices
Oops - GWL Gates Aren’t Self-Resetting

Simulation

Sampled
High-Performance SRAM

- Low-swing gates for decoders
- SRAM clocking
  - Replica techniques
- Half-swing bitlines
- Results and power analysis
SRAM Clocking Goals

- Clocks can be used to
  - Speed up the access
  - Reduce the power
- Speed up access
  - Much of the delay is waiting for the bitlines to flip
    - May need to wait for the output of the sense amp to flip too
  - We can make the delay smaller by “equalizing” the bitlines
    - Short bitlines together before access
    - Then don’t need to wait until bitlines cross
- Power
  - Turn off the sense current when not needed
  - Turn on only after an access
- Determine when an access occurs
  - For caches, read/write every cycle.
  - Asynchronous access, by using address transition detection (ATD)
Address Transition Detector

- Detects a change in the address bits.
  - This produces a “clock” that starts the access.
- Produces a pulse to indicate the event.
  - Basically an XOR per address bit
    - Comparing current and previous address.
    - Pulse width is the length of the delay.
  - Followed by a high Fan-In OR
Review of SRAM Read Timing

- The timing has only a few critical points.
  - Address transition triggers the timing event.
  - $t_{pchg}$ needs to be long enough to equalize and precharge the bitlines.
  - $t_B$ needs to be long enough for bitlines to split sufficiently.
  - Total access time is the time until the sense amplifier output flips the output wires.
  - The timing margin depends on the type of sense amplifier.
Static Differential Pair Amplifiers

- Notice that $t_A$ is not critical.
  - If the precharge/equalization pulse is too long (negative $t_A$),
    - It would only delay the bitline splitting (increase delay)
  - If it is too short,
    - Delay is a little longer, since the bitline were not equalized
  - In both cases the memory gets the right answer
- A similar situation happens for the Sense_EN, $t_B$.
  - If it is early, the amplifier waits until the signal arrives.
  - If it is late, the output settles a little late
- Timing margins are less critical.
Clocked Amplifiers

• The timing of $t_B$ is critical
  – If the signal arrives too early, the bitlines haven’t split in voltage sufficiently.
  – If the signal arrives too late, it still functions correctly.
    • Output settles later.
  – Timing margin on one side.
• Because $t_B$ is important, so is $t_A$.
  – If the precharge/equalization pulse is too long (negative $t_A$), it would delay the bitline splitting.
    • Require a longer $t_B$, otherwise an error occurs.
  – If it is too short, potential problem.
    • Since the bitline were not equalized fully, it takes longer for the splitting to occur. $t_B$ must be longer.
  – In both cases, the memory can malfunction.
• Timing margins are very critical.
Impact on Power

• Bitline capacitance
  – Burn more power when it swings larger than necessary voltage.
  – Wordline only needs to be HIGH for sufficient voltage splitting.
    • Shut down the wordline when the voltage has split sufficiently.

• DC amplifier burns more power
  – If it is enabled too early.
  – Or if it stays enabled too late

• Tradeoff between power and voltage/timing margin.
Precise Clock Generation

- Assume you want to use a true clocked sense-amplifier
  - How do you generate the clock?

- Problem is that inverters might not match memory cell
  - Memory cell has very narrow devices and lots of diffusion capacitance
Delay Tracking

• Use adjustable delay cells
  – Use an adjustable capacitance
  – Analog or digital control
• But the basic problem is still there.
  – Tracking can be bad across corners.
  – Main reason is because inverters match inverters well… but not any other types of gates.
  – Use the same type of gate for proper matching.
    • Dynamic gate example
• Bit line is even more difficult.
Replica Bitline Tracking
Sense Clock

• Match similar elements
  – Need a full swing replica of bitline delay
Bitline Replica

- Ratio bitline capacitance
  - Replica cell same as memory cell
  - Bitline is simply 1/10 as long

![Diagram showing Replica bitline path and Main bitline path with ratios 1/10 and 1.](image)
Replica Layout

- Need to match real cells
  - Need dummy lines, to make sure local environment is the same
- Have a dummy wordline that fires every cycle
- Still need some margin
Replica Delay

• The other option is to use full bitline capacitance
  – And use the current of 10 cells
  – Can adjust timing by adjusting the number of cells that are on
Use Replica Bitline to Control WL Pulse Width
High-Performance SRAM

- Low-swing gates for decoders
- SRAM clocking
- Half-swing bitlines
  - Charge recycling
  - Noise margin and coupling issues
- Results and power analysis
Vdd/2 Bitlines

- Most low power SRAM techniques only decrease read power
- Decrease write power by using Vdd/2 bitlines
- Possible read instability
- Solution: increase Vcc above Vdd

![Diagram of Vdd/2 Bitlines](image-url)
Vdd/2 Bitline Implications

- Higher Vcc makes it harder to write into the cell
  - Solution: Dynamically lower Vcc during writes
  - Requires extra circuit in local wordline driver - 10% area hit
- Charge recycle between bitlines and write data bus
  - Bitlines swing from Vdd/2 to Gnd - negative pulse
  - Write data bus swings from Vdd to Vdd/2 - positive pulse
  - Need to nominally match capacitances - encode write data bus
  - 2:4 ‘encoding’ reduces write data bus switching, match BL capacitance
  - Requires ‘decoding’ on the SRAM block end
Write Decoder

- Winner-takes-all circuit prevent incorrect firing due to noise
- Note the use of PMOS style half-swing pulse-mode gates
- Write data inputs are positive pulses to balance BLs
Noise and Signal Integrity

- Half-swing pulse-mode gates have lower noise margins
- Lots of long parallel wires on chip
  - Write data
  - Read data
  - Predecoded address lines
Layout Fixes

- Interleave positive and negative pulse lines

- Twist lines in a one-hot bus
High-Performance SRAM

- Low-swing gates for decoders
- SRAM clocking
- Half-swing bitlines
- Results and power analysis
Power Results

- Small deviation from ideal scaling at high Vdd indicate low leakage current in half-swing pulse-mode gates

![Power vs Vdd graph](image)

- Measured power scales approximately as $V^2$

- Read:Write = 1:1

- 100 MHz
Power Breakdown

- Read and write power are nearly identical

<table>
<thead>
<tr>
<th></th>
<th>Read Full-swing</th>
<th>Read Half-swing</th>
<th>Write Full-swing</th>
<th>Write Half-swing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1.10 mW)</td>
<td>(0.90 mW)</td>
<td>(1.66 mW)</td>
<td>(0.90 mW)</td>
</tr>
<tr>
<td>Decoder</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R-bus + SA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bitline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write circuitry</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoder + WA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bitline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

- A design example that uses a number of the digital design techniques we have discussed so far.
  - Unique to an application – SRAM lends itself to a number of these optimizations.
  - Possible in other applications as well.
- Novel logic family
  - Reduced voltage swing
  - Use of High-Vt CMOS
- Charge recycling
  - Decoder and bitlines
- Sense amplifier clocking
  - Replica delay matching
- Noise reduction