EECS 151/251 A
Discussion 11

April 5, 2024

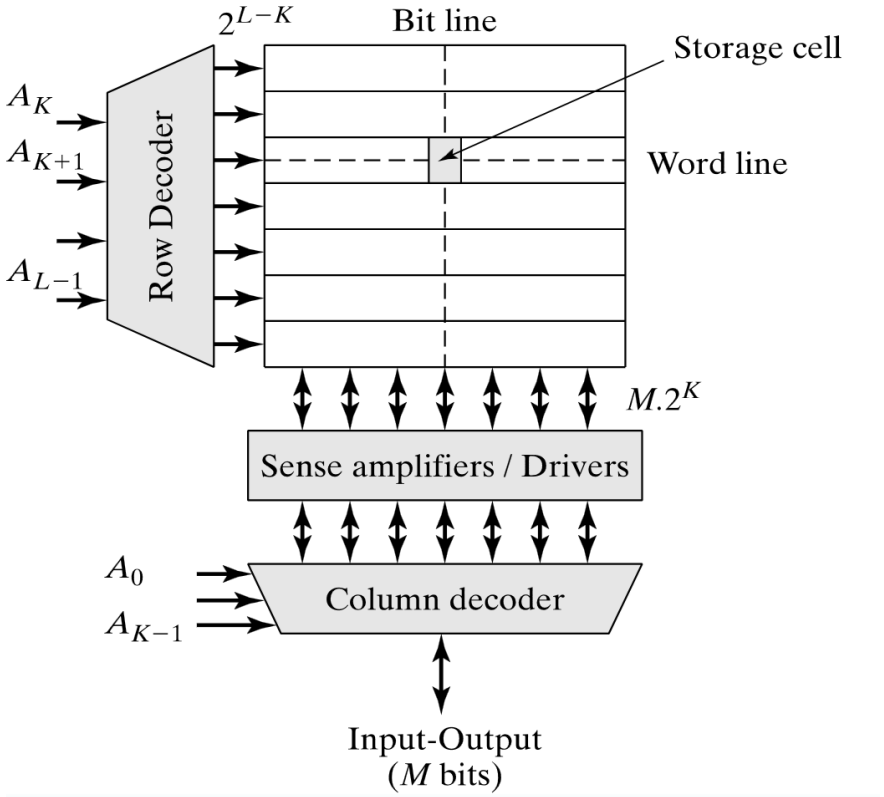
Berkeley
UNIVERSITY OF CALIFORNIA

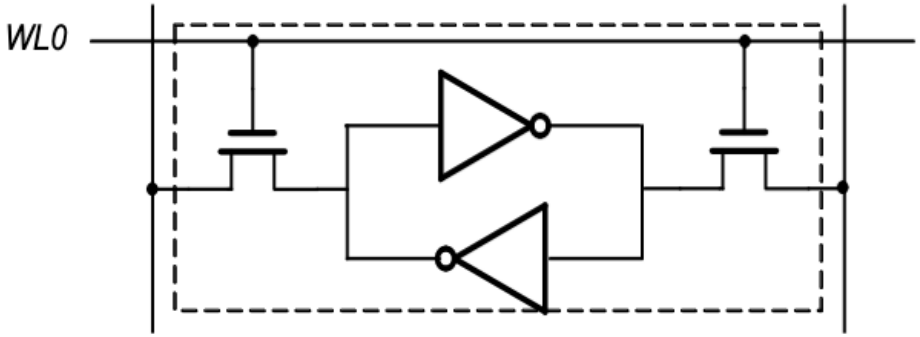
Content

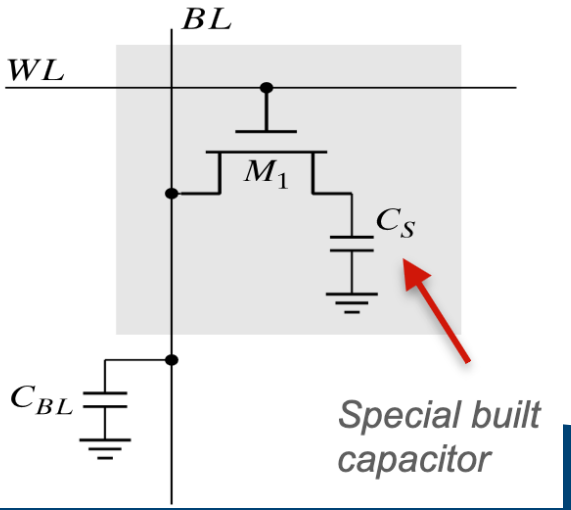
- Memory Blocks
- Power

Memory

- Volatile Memory
  - Lost on power off
  - Static (SRAM) vs Dynamic (DRAM)
- SRAM
  - 6 transistors
  - 2 bit lines on each side
- DRAM
  - Essentially a capacitor
  - Destructive read, need to write-back
  - Needs refresh

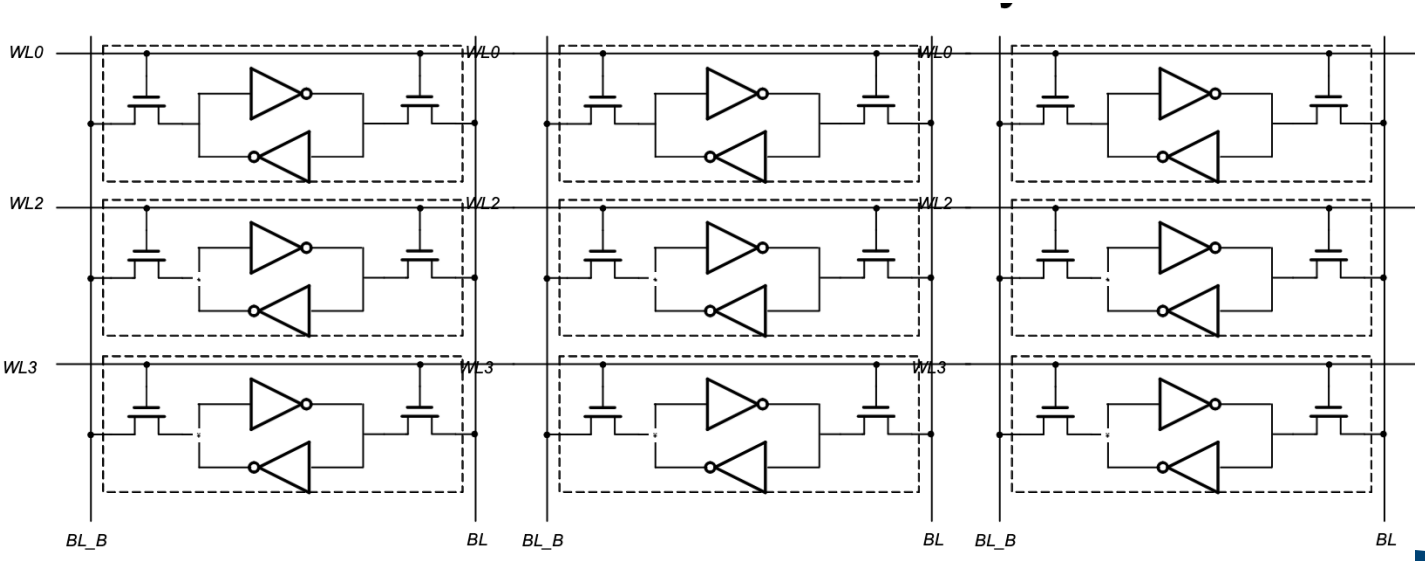






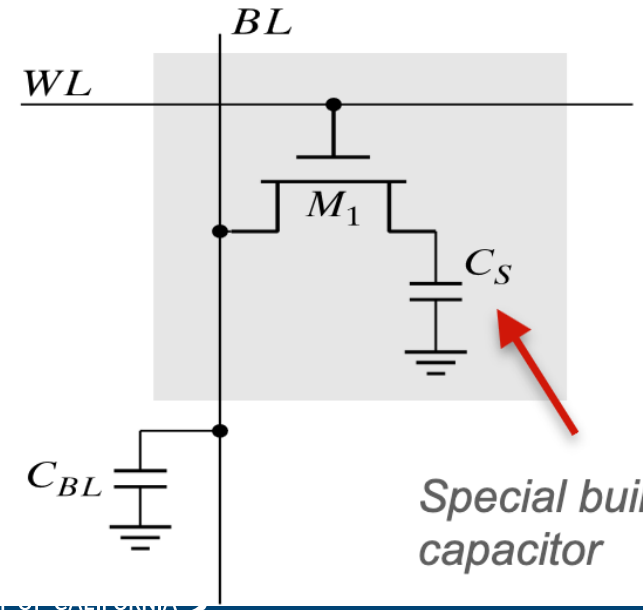
SRAM

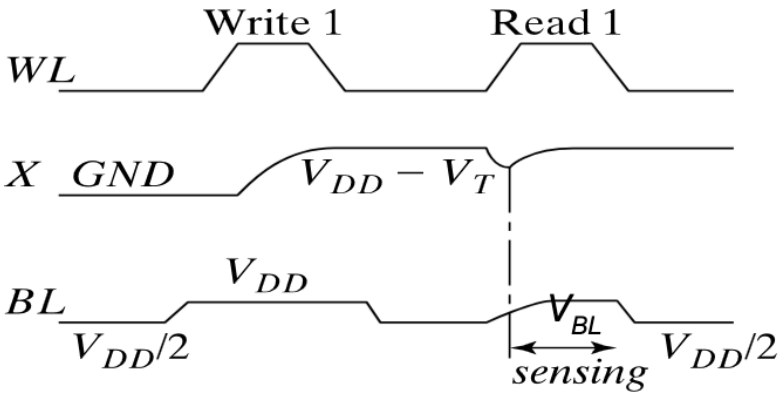
- Read
  - Bit lines are pre-charged to Vdd
  - Word Line is driven High
  - Differential Sensing to get value, value is not destroyed
- Write
  - Differentially drive Bit Lines (0 and 1 or 1 and 0)
  - Word line is driven high, cells flip



DRAM

- Write
  - Cs is charged by setting WL and BL high, Discharged by setting WL high and BL low
- Read
  - Pre-charge BL to Vdd/2, and then sense change after WL is written high. Whatever the bit is, write it back to the cell

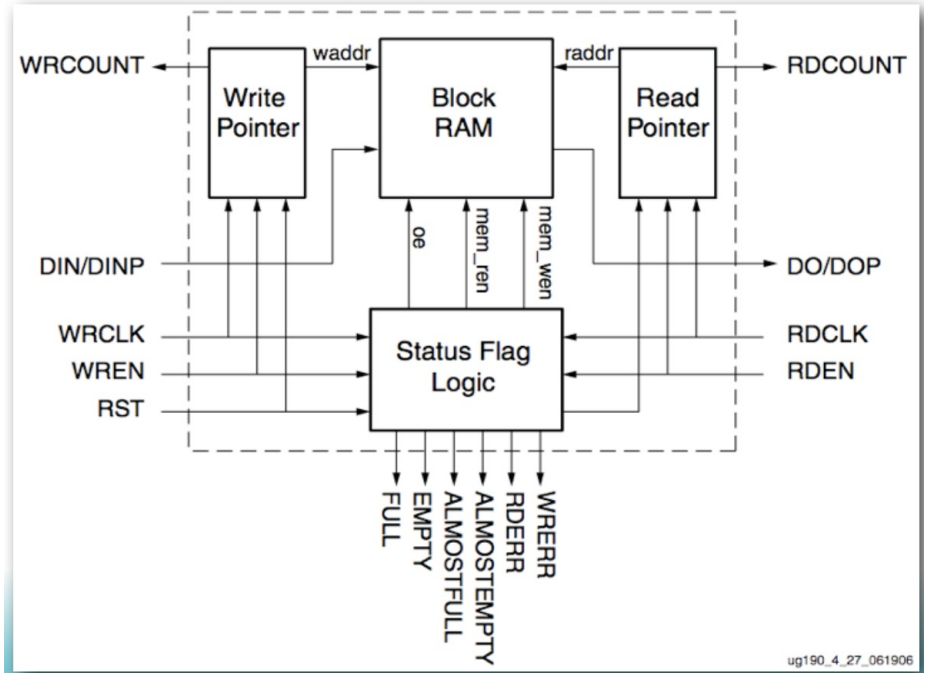




V_{BIT}= 0 or $(V_{DD} - V_T)$

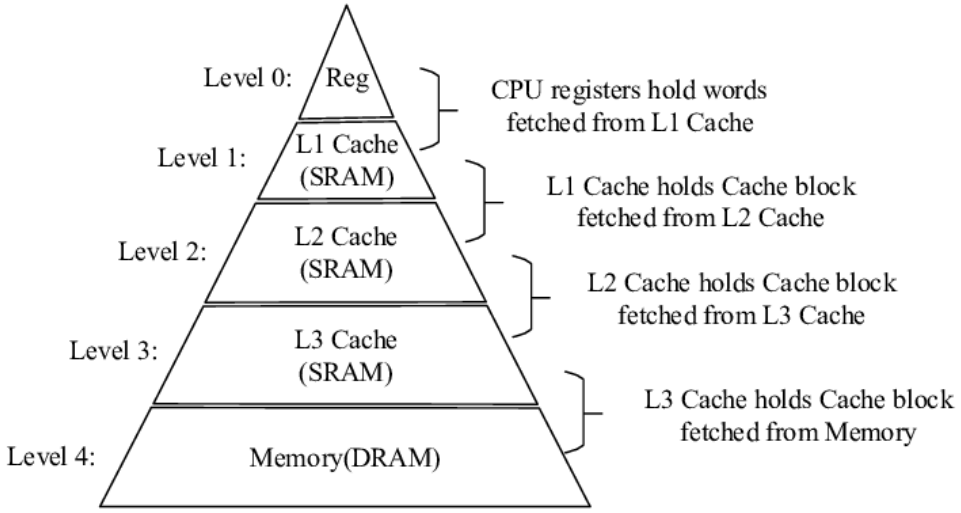
Advanced Memory Blocks

- Multi-ported Memory
  - Duplicate all of the decoders, bit lines, word lines, and sensors/logic
  - Can also duplicate Memory and use a multiplexer
- Building Larger Memory from smaller blocks
  - Split addresses, use bits of address enable which smaller memory block is used
- FIFO
  - 2 pointers, read and write
  - If equal after write -> FULL, if equal after read -> EMPTY



Caches

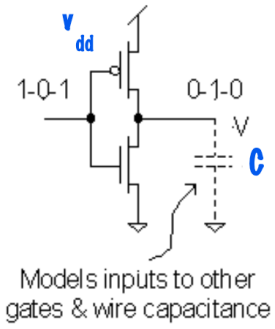
- On-chip SRAM is faster but more expensive than DRAM
- Want to use a memory Hierarchy (L1, L2, L3 cache, DRAM, Flash, …)
- Caches take advantage of temporal and spatial locality
- Have design choices to increase the hit rate / minimize the miss penalty
  - Cache size, Cache block size
  - Fully Associative vs. Set Associative vs. Direct Mapped
  - LRU (typical) vs MRU
  - Write Through vs Write Back
  - Prefetching

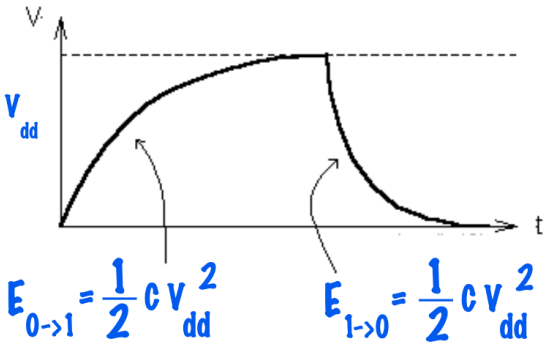


https://www.researchgate.net/publication/362707415/figure/fig1/AS:11431281080715807@1661405291147/A-classical-three-level-cache-hierarchy.png

Energy and Power

- Energy (W) is the amount of work you do
- Power is the rate at which you do work (J = W/s)
  - P = IV
- Switching Energy
  - Every 1->0, 0->1 dissipates energy!
  - $\frac{1}{2}$ C Vdd^2





https://www.researchgate.net/publication/362707415/figure/fig1/AS:11431281080715807@1661405291147/A-classical-three-level-cache-hierarchy.png

Dynamic vs Static vs Short Circuit

- Dynamic Power:
  - Power lost due to switching (actually doing work)

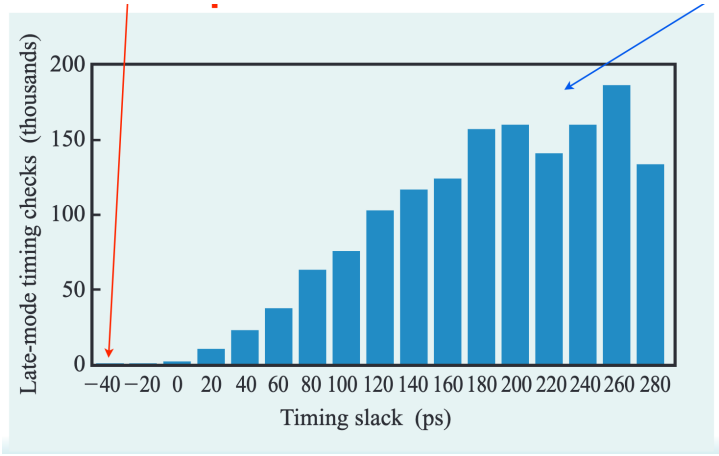
$$P_{sw} = 1/2\ \alpha\ C\ V_{dd}^{2}\ F$$

- Short Circuit Power
  - When gate switches, pull-up and pull-down are briefly on at the same time, P = $I_{sc}V$
- Leakage/Static Power
  - If Logic gate is on but not switching, it can also burn power since capacitors are not ideal!
  - P = $I_{off}V$
  - I_{off} and I_{on} are both inversely proportional to V_t!
  - Why can't we just make I_{off} as small as possible?

https://www.researchgate.net/publication/362707415/figure/fig1/AS:11431281080715807@1661405291147/A-classical-three-level-cache-hierarchy.png

Improving Efficiency

- Parallelize and reduce Vdd and F
  - You get the same throughput, double the area, but ¼ of the power! (½ the power density)
- Dynamically scaling Vdd and F
  - DVFS is common in most BIOS systems
- Adding sleep transistors for logic that is not in use -> reduce leakage
- Only using high voltage on the critical path (You can either vary the Vdd or the Vt, usually do Vt)



Berkeley
UNIVERSITY OF CALIFORNIA

https://www.researchgate.net/publication/362707415/figure/fig1/AS:11431281080715807@1661405291147/A-classical-three-level-cache-hierarchy.png

Question 1

Problem 2: Inverter Power

Consider an inverter with 10 fF load capacitance including the parasitic capacitance. Assume V_{dd} is 0.9 V.

(a) When the output of the inverter changes from 0 to 1, how much energy is dissipated?

(b) When the output of the inverter changes from 1 to 0, how much energy is dissipated?

(c) If the input of the inverter is the clock signal, how much power does it consume? Assume clock period is 1 ns.

(d) If the input of the inverter flips on a positive clock edge with the probability $\alpha = 0.1$, how much power is it expected to consume? Assume clock period is 1 ns.

Solution

Solution:

(a)

Energy charged in the capacitor:

$$\int IV\,dt = \int \frac{dQ}{dt}V\,dt = \int C\frac{dV}{dt}V\,dt = \int_0^{V_{dd}} CV\,dV = \frac{1}{2}CV_{dd}^2$$

Energy provided from source:

$$\int IV_{dd}dt = V_{dd}\int C\frac{dV}{dt}dt = V_{dd}\int_0^{V_{dd}} C\,dV = CV_{dd}^2$$

Energy dissipated:

$$CV_{dd}^2 - \frac{1}{2}CV_{dd}^2 = \frac{1}{2}CV_{dd}^2 = 4.05 \text{ fJ}$$

(b)

$$\frac{1}{2}CV_{dd}^2 = 4.05 \text{ fJ}$$

(c)

$$\frac{1}{2}CV_{dd}^2 \times 2f = CV_{dd}^2 f = 8.1 \text{ µW}$$

(d)

$$\frac{1}{2}\alpha CV_{dd}^2 f = 405 \text{ nW}$$

Question 2

Problem 3: Race to Halt

One scheme for potentially improving energy efficiency if static power is a significant proportion of the total power consumption is a technique known as "race to halt". Basically, we run the circuits at maximum speed to finish the computation as quickly as possible, then cut off the power so that we don't suffer the static power loss.

Suppose we have a CPU that takes 10 seconds to run a particular application, consuming 12 W, where some propotion δ of the total power is consumed by dynamic power, with the remaining $\sigma = 1 - \delta$ the proportion lost to static power consumption. Assuming there are no other applications running on the CPU and that these two proportions cover the entire power budget of the CPU, we would like to find a scheme that would minimize power consumption.

As an alternative to the race to halt method, we could also consider a more traditional frequency/voltage scaling method for reducing power consumption. The technology we are working with can tolerate a V_{DD} reduction by at most 25%, and we can assume that the voltage to delay scaling is linear (e.g. a 2x reduction in supply voltage will need a 2x decrease in clock frequency).

Explore race to halt versus frequency/voltage scaling. Assume that the voltage and frequency scaling will only affect dynamic power consumption, and does not affect the amount of static power consumption (which is not an unrealistic approximation). For what δ would race to halt be better than frequency/voltage scaling?

Solution

We can express the energy consumed by the race to halt processor as follows:

$$(12)(10)(\delta)$$

The energy consumed by the scaling processor is expressed as

$$(12)(10/0.75)(1-\delta)+(12)(\delta)(0.75)^3$$

To find the crossover point, we set the two equal and solve for δ,

$$(12)(10)(\delta)=(12)(10/0.75)(1-\delta)+(12)(\delta)(0.75)^3$$

This converges at $\delta \approx 0.58$.