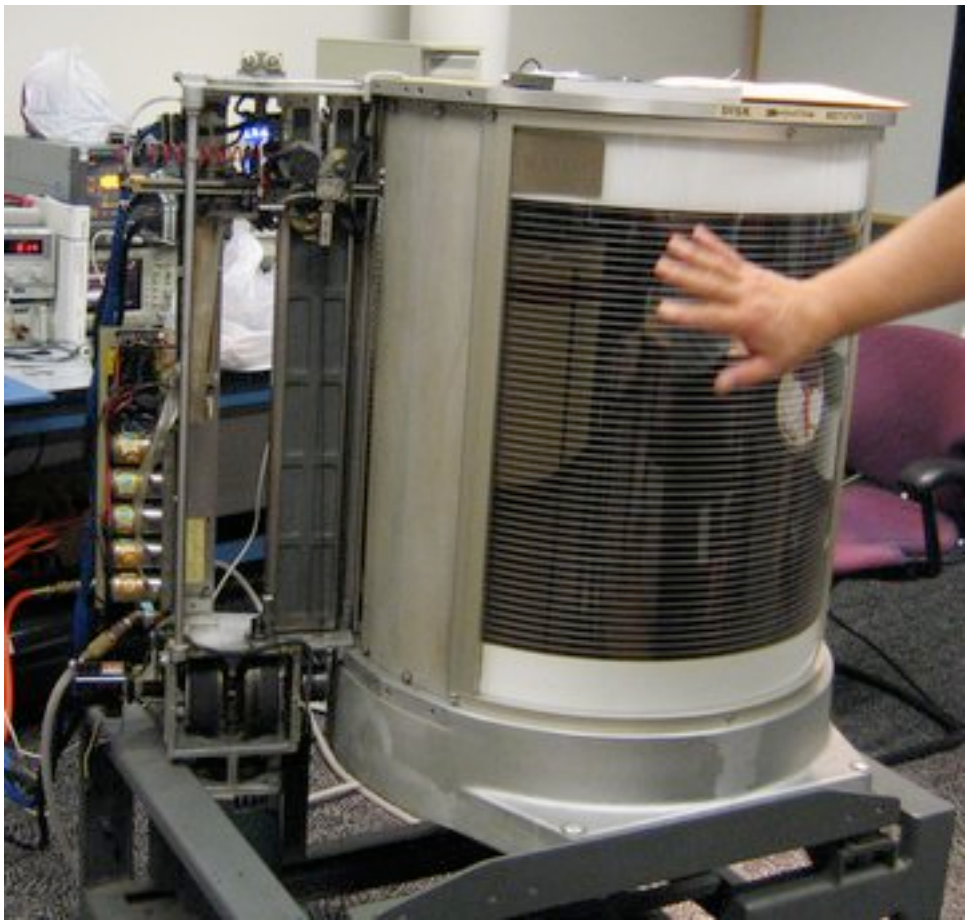
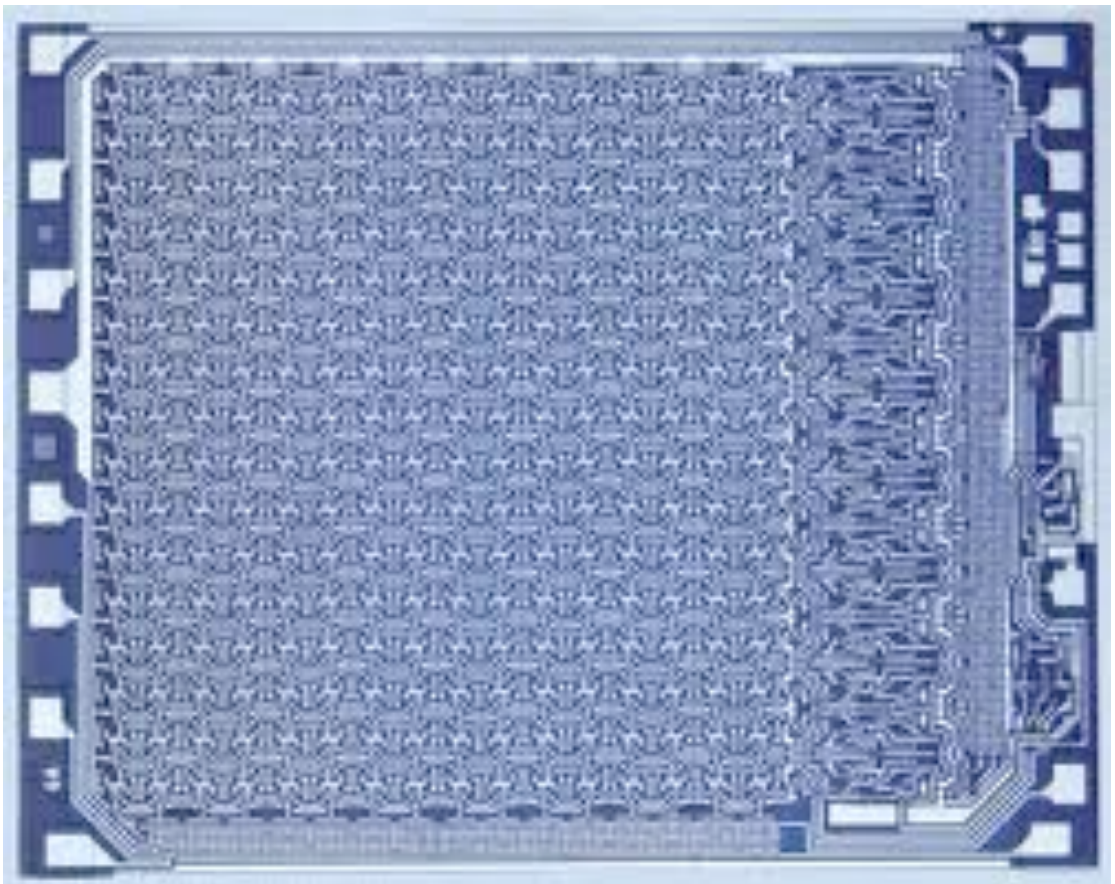
information storage & processing



IBM 350 RAMAC, the first hard disk

it stored about 4.4Mb

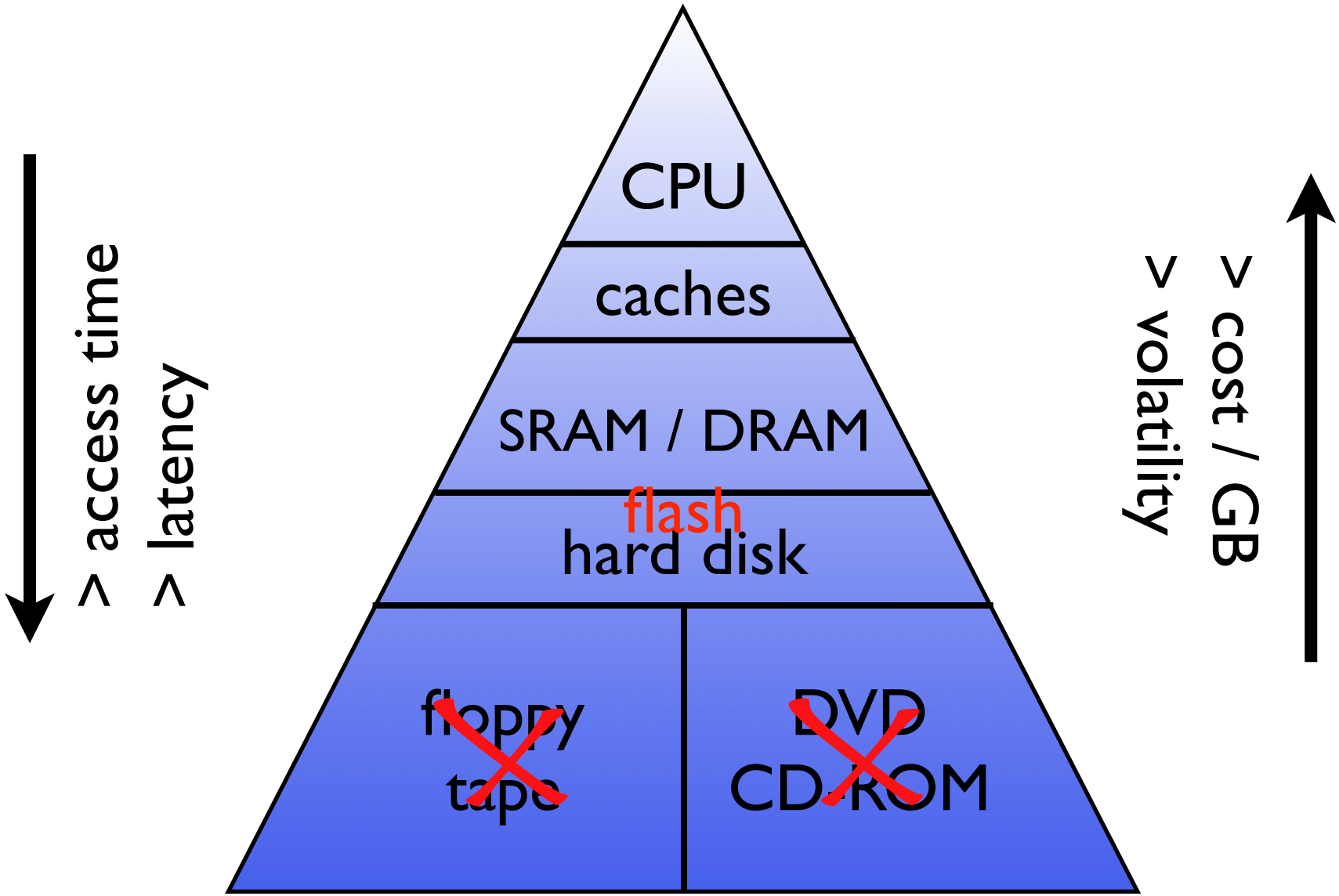
wikipedia.org - "RAMAC"



first 256 ***bit*** static RAM.

thenonist.com

traditional PC architecture



terminology

RAM

random access memory

ROM

read-only memory

access time & latency?

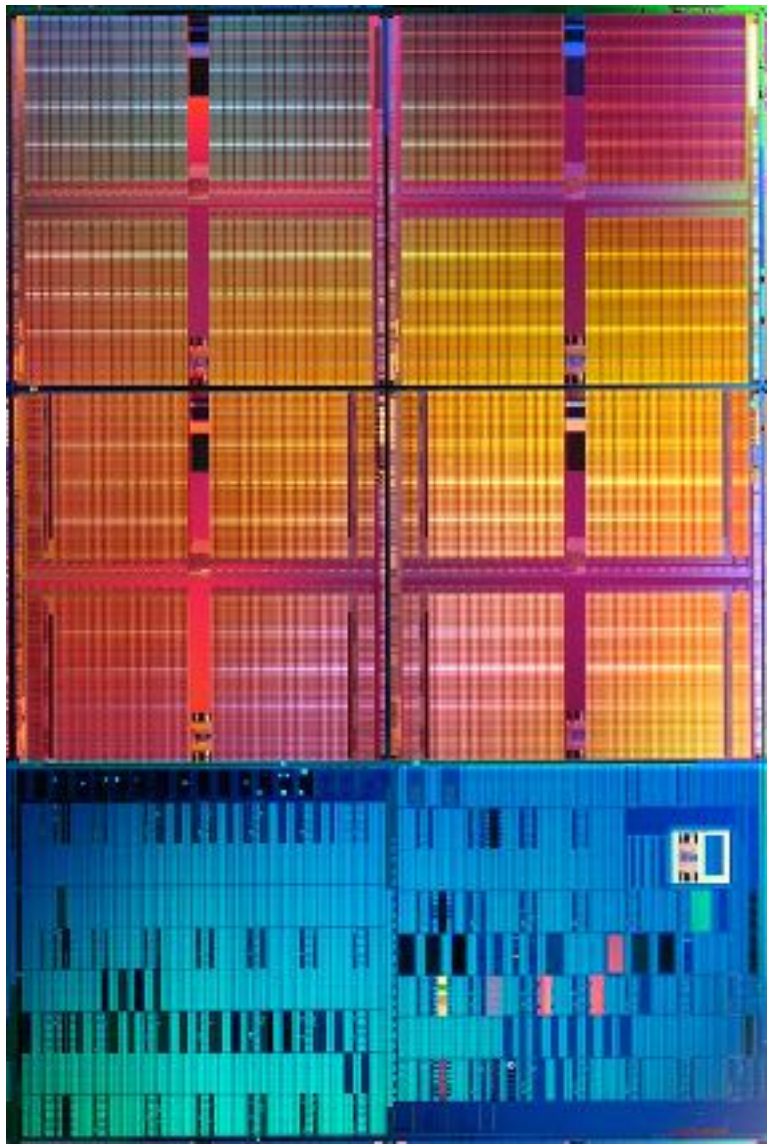
time between request for info & info returned

$ / GB

primary figure of merit.
most other things can be worked around

nonvolatility?

retains data without power



45nm SRAM die intel.com

every bit has a role

cache - *reduce latency to main memory*

- *small* memories close to CPU
- even faster than main memory
- temp storage of frequently accessed items

SRAM / DRAM - *main memory*

- blazingly fast, relatively large
- volatile!!

HDD - *mass storage*

- higher latency
- enormous capacity, nonvolatile

flash

- very fast, nonvolatile
- write endurance, price, capacity

removable

- portability, backup

EDSAC / wikipedia.org

the need for hard disks & flash (tech)

volatility of semiconductor memories!

- *some* sort of nonvolatile storage necessary
- why not just battery backup of SRAM?

cost per GB

- SRAM/DRAM are too expensive
- Flash is too *borderline* too expensive
- cache RAM is more expensive

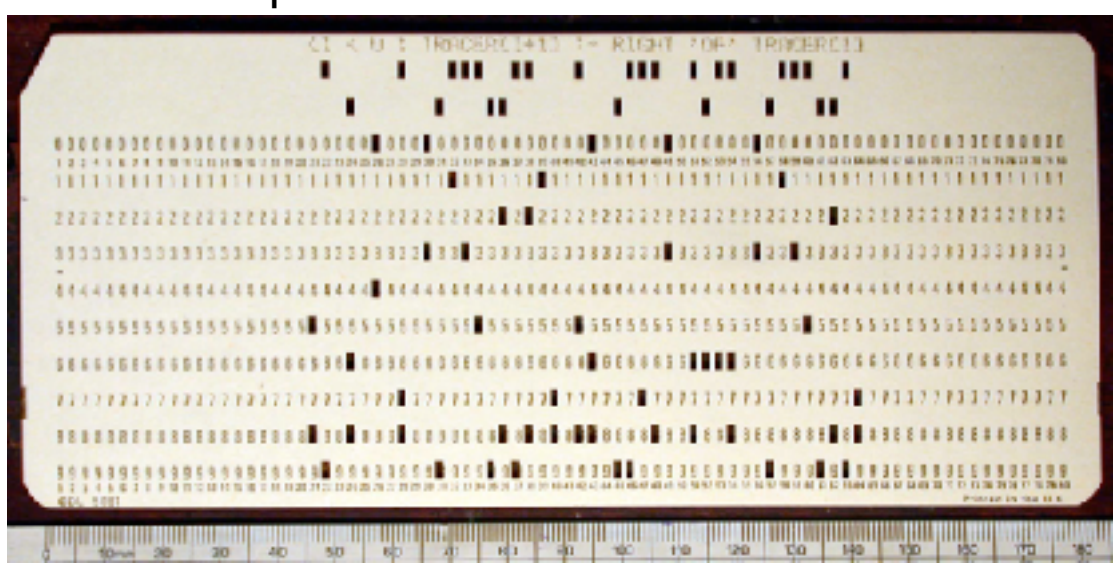
size & throughput

- higher latency, but bandwidth is huge
- enormous sizes

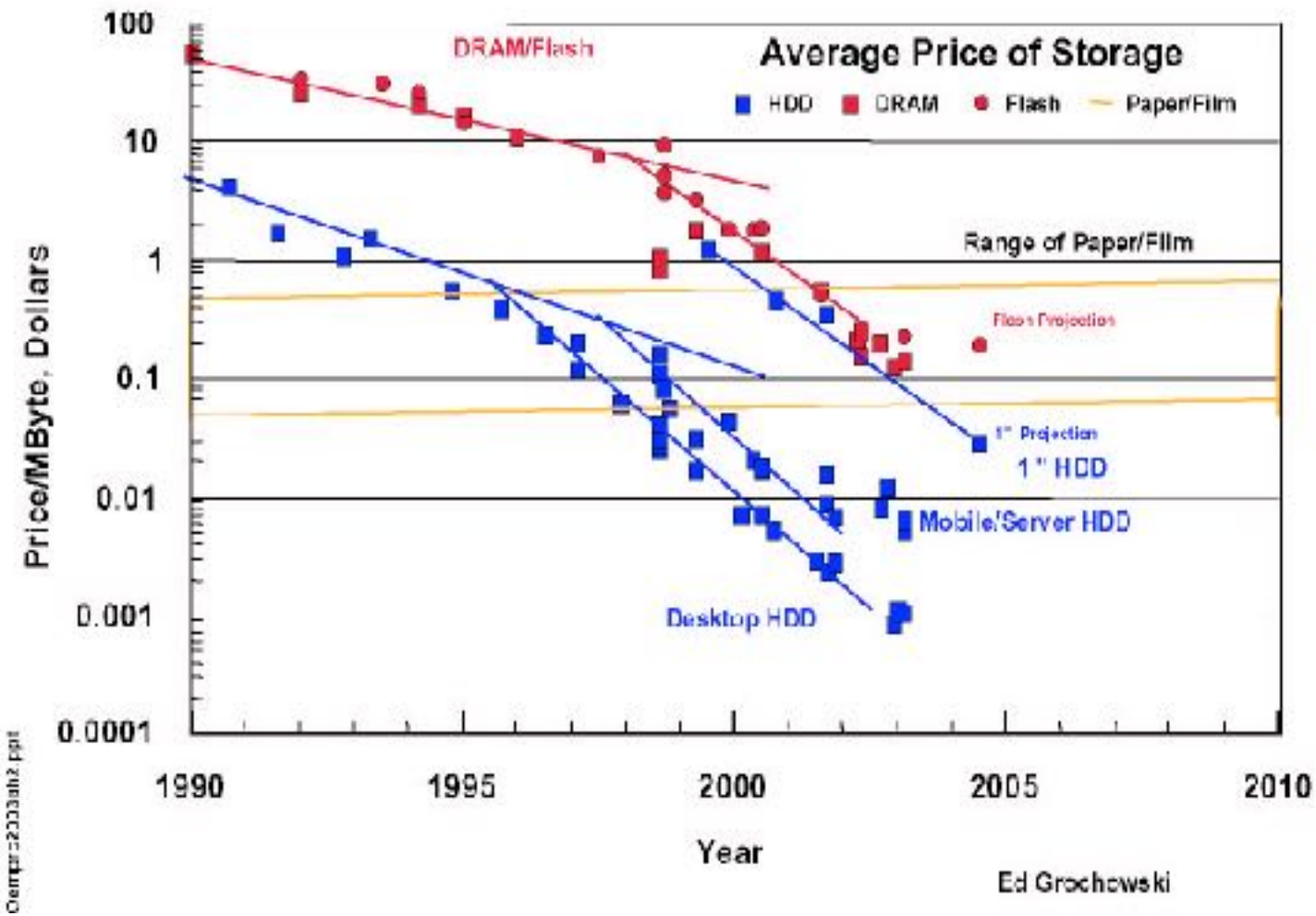
endurance

- essentially unlimited cycling
- radiation hard

punched cards are nonvolatile



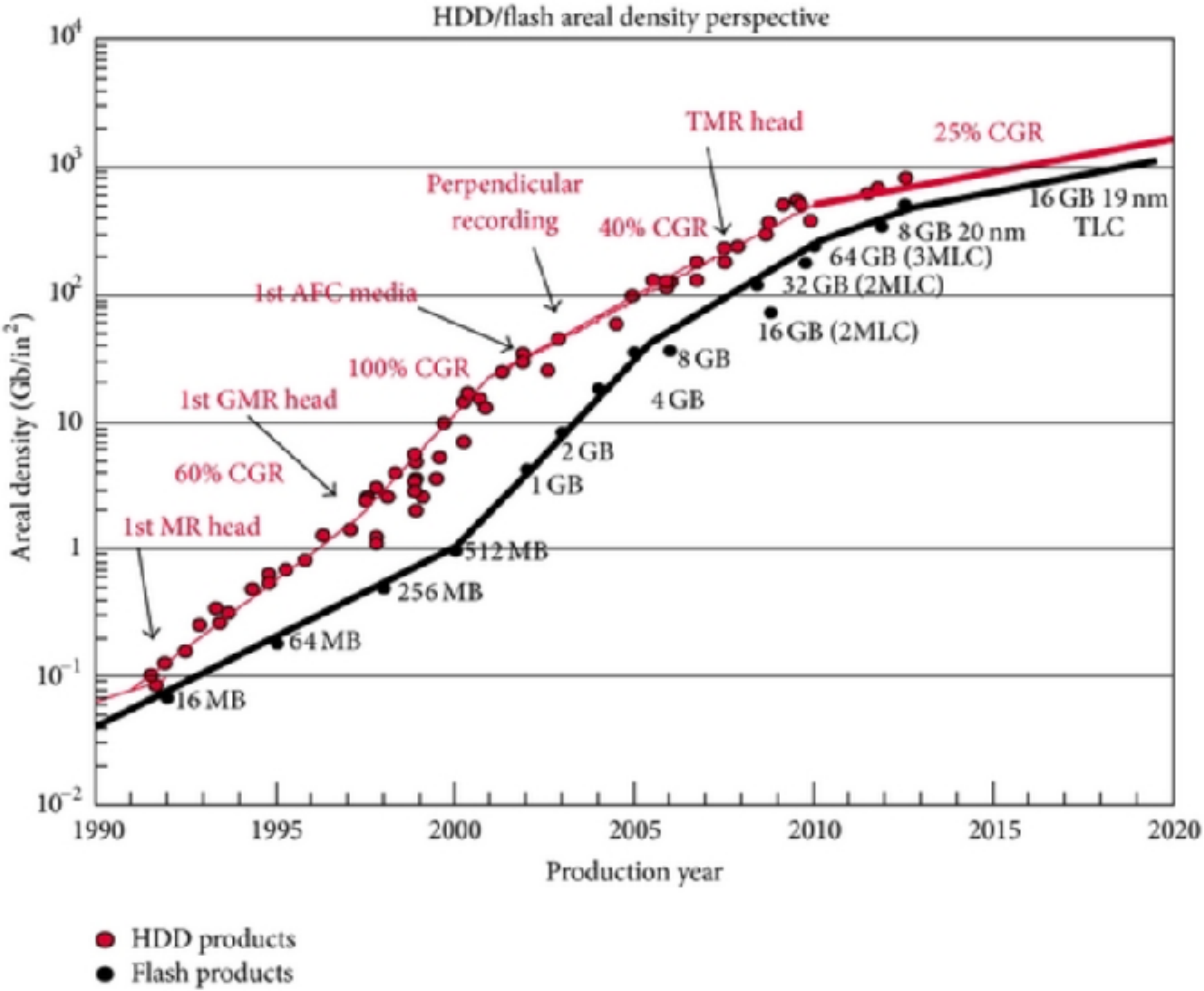
price is the real advantage.



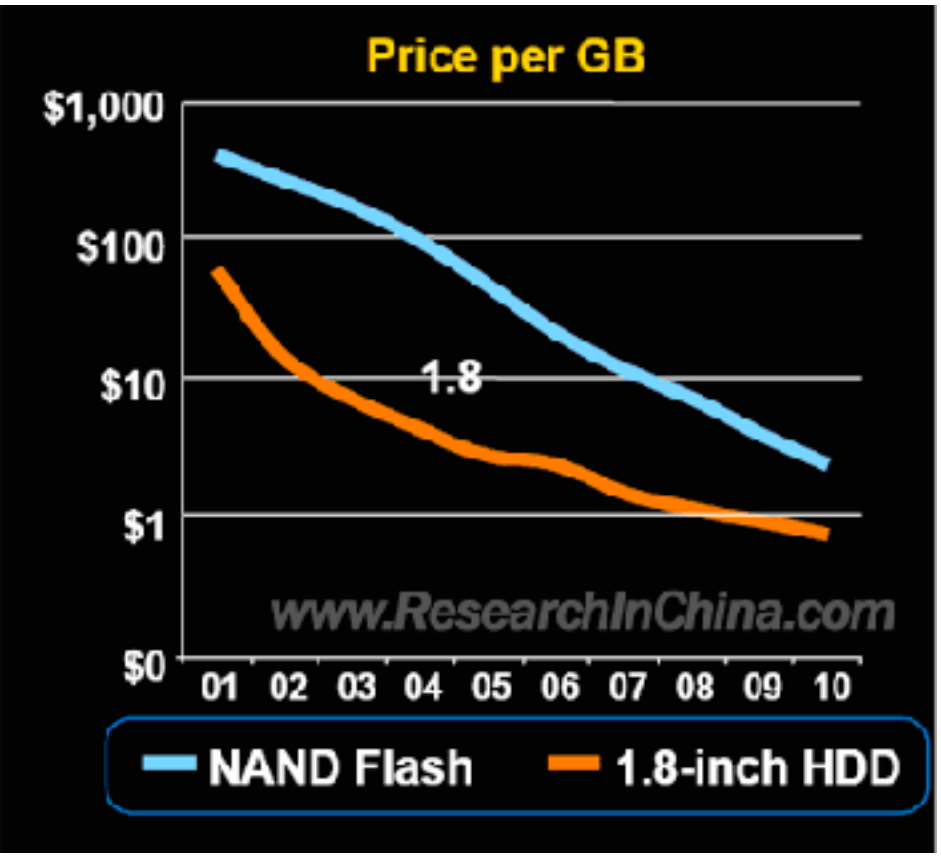
flash is beating the HDD in some apps

e.g., phone, tablet

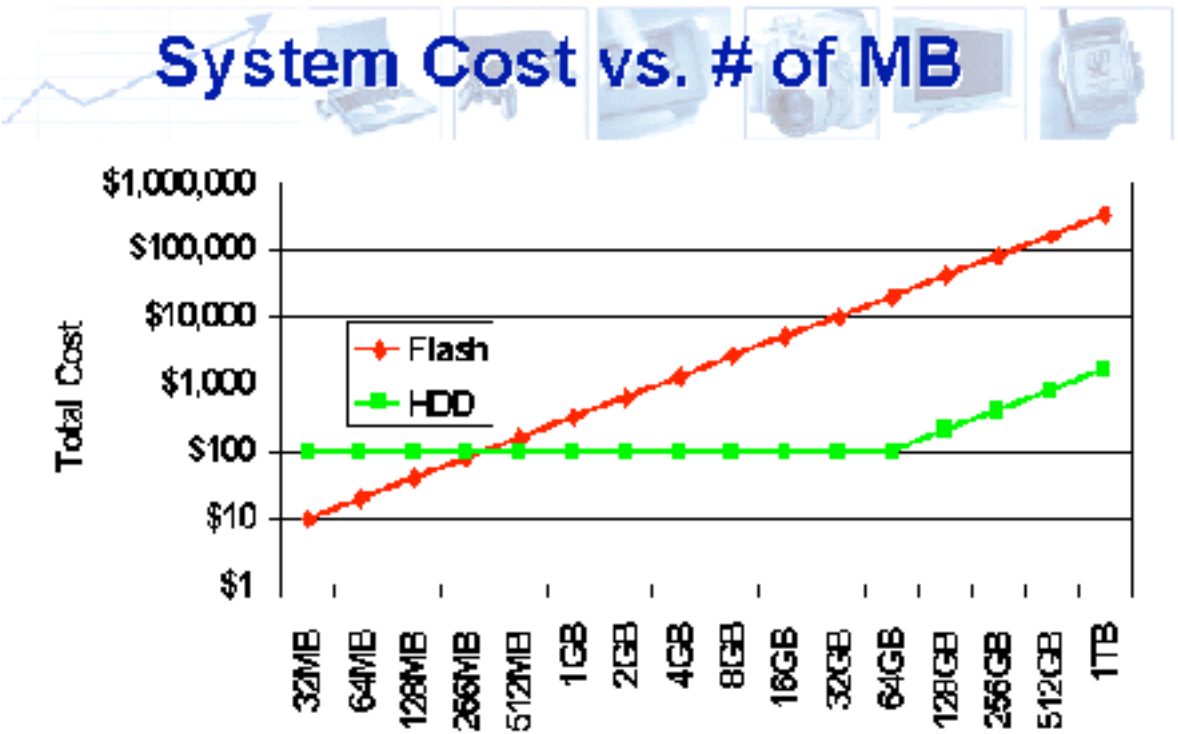
power consumption may be the larger issue


HDD/flash areal density perspective
Areal density (Gb/in^2)
10^4
10^3
10^2
10
1
10^{-1}
10^{-2}
1990
1995
2000
2005
2010
2015
2020
Production year
TMR head
25% CGR
Perpendicular recording
40% CGR
1st AFC media
100% CGR
1st GMR head
60% CGR
1st MR head
16 GB 19 nm TLC
8 GB 20 nm
64 GB (3MLC)
32 GB (2MLC)
16 GB (2MLC)
8 GB
4 GB
2 GB
1 GB
512 MB
256 MB
64 MB
16 MB
HDD products
Flash products

flash is competitive with HDD except for very large drives already true for smaller sizes for a long time (phone, tablet ...)

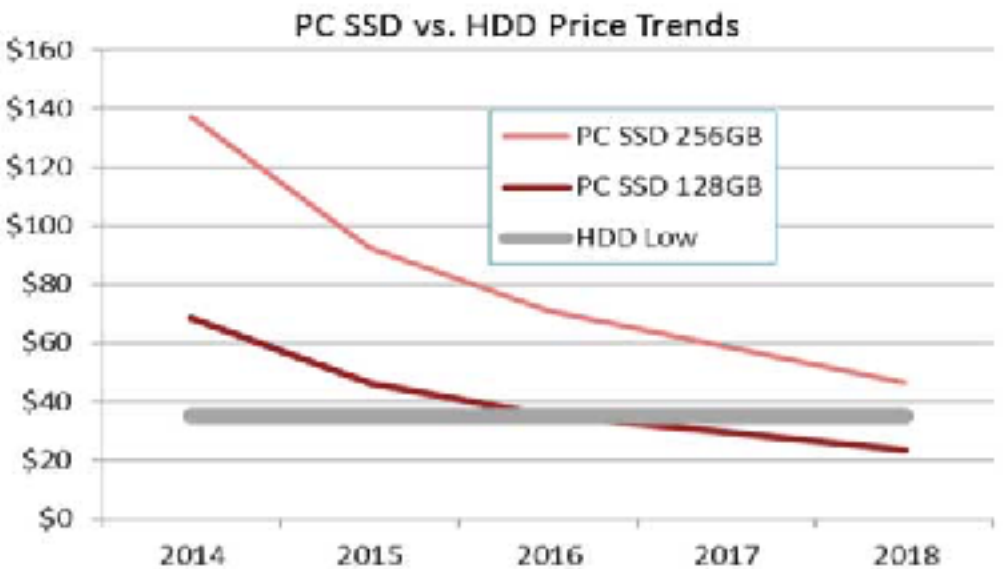


http://www.researchinchina.com/Htmls/Report/2008/3390.html

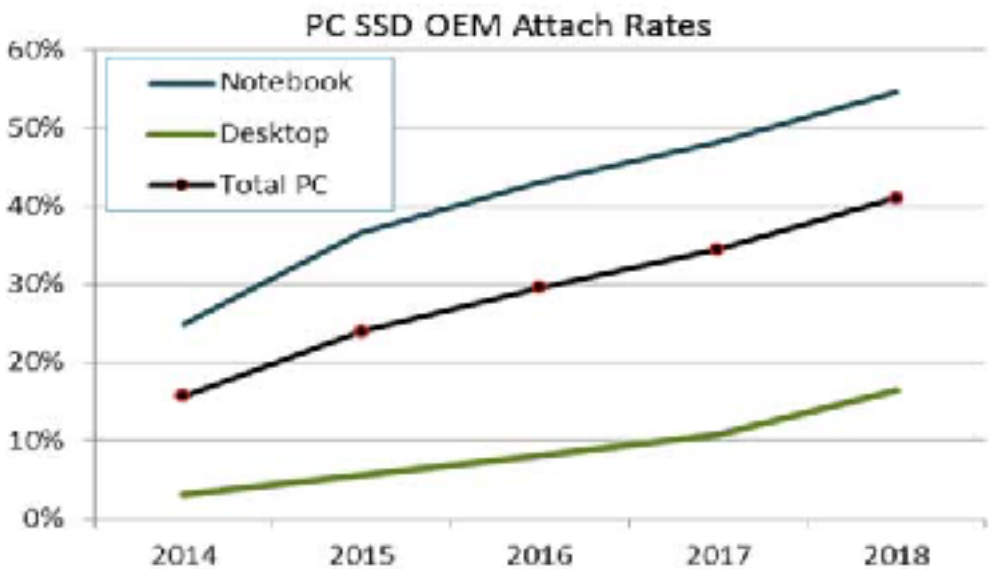


2005, http://www.storagesearch.com/semico-art1.html

Inflection point reached when cost of utility drops below HDD

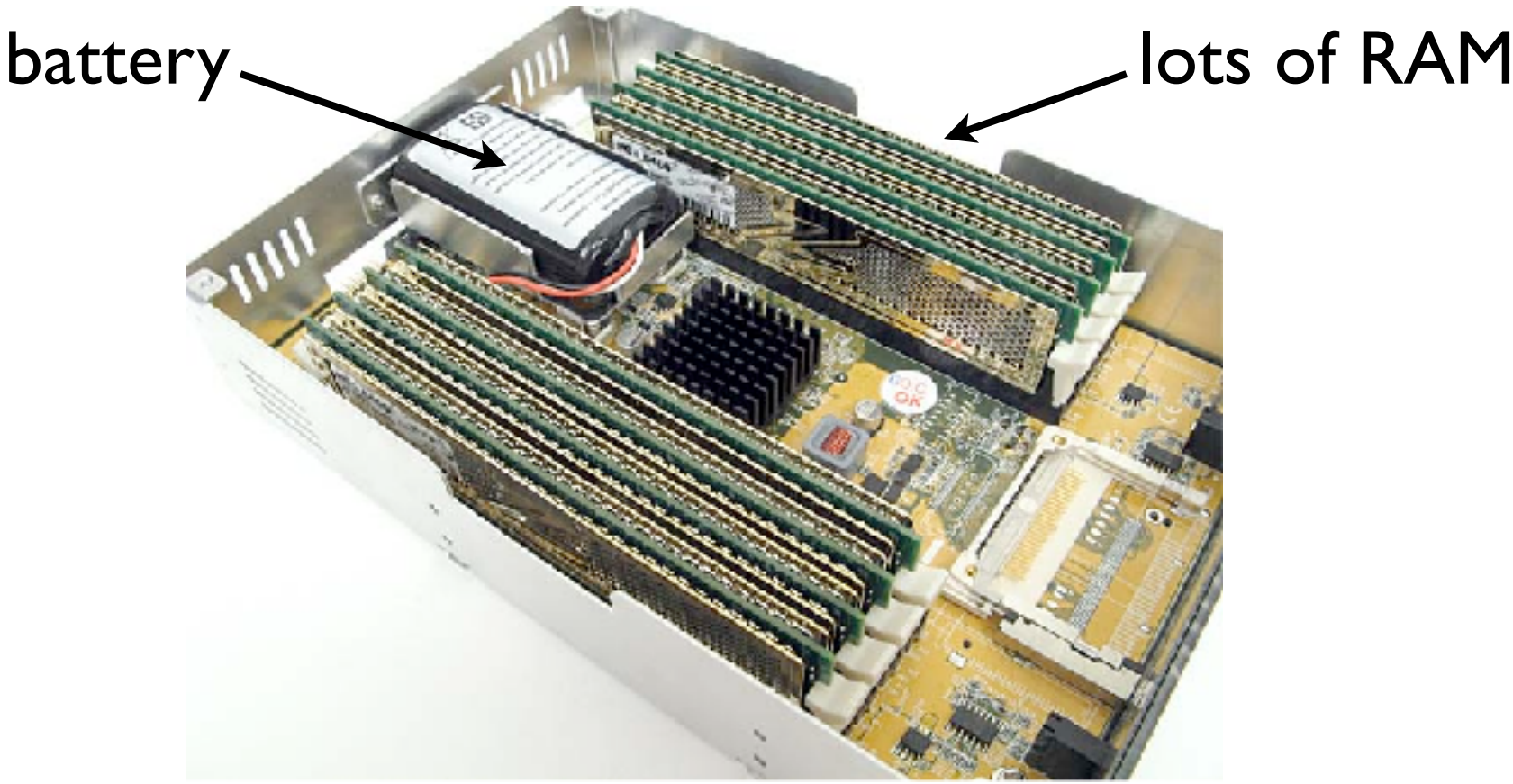


Driving an increased rate of SSD adoption into PCs



Back in the day, disks were expensive.

Sometimes, we would trick the system into using RAM as a disk to avoid swapping floppies.



now RAM disks make a comeback ... and then flash

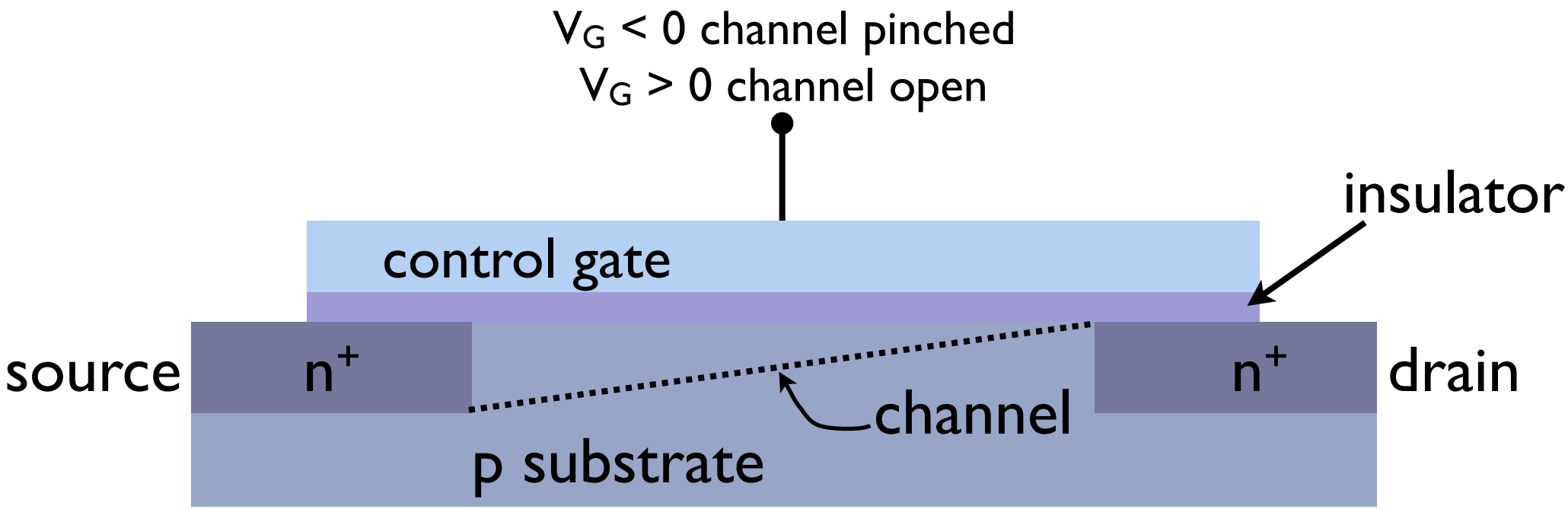
On to logic and memory ...

... we're going to need transistors

how a MOSFET transistor works

we need this for flash and logic.

use electric field to suppress dopant carriers!



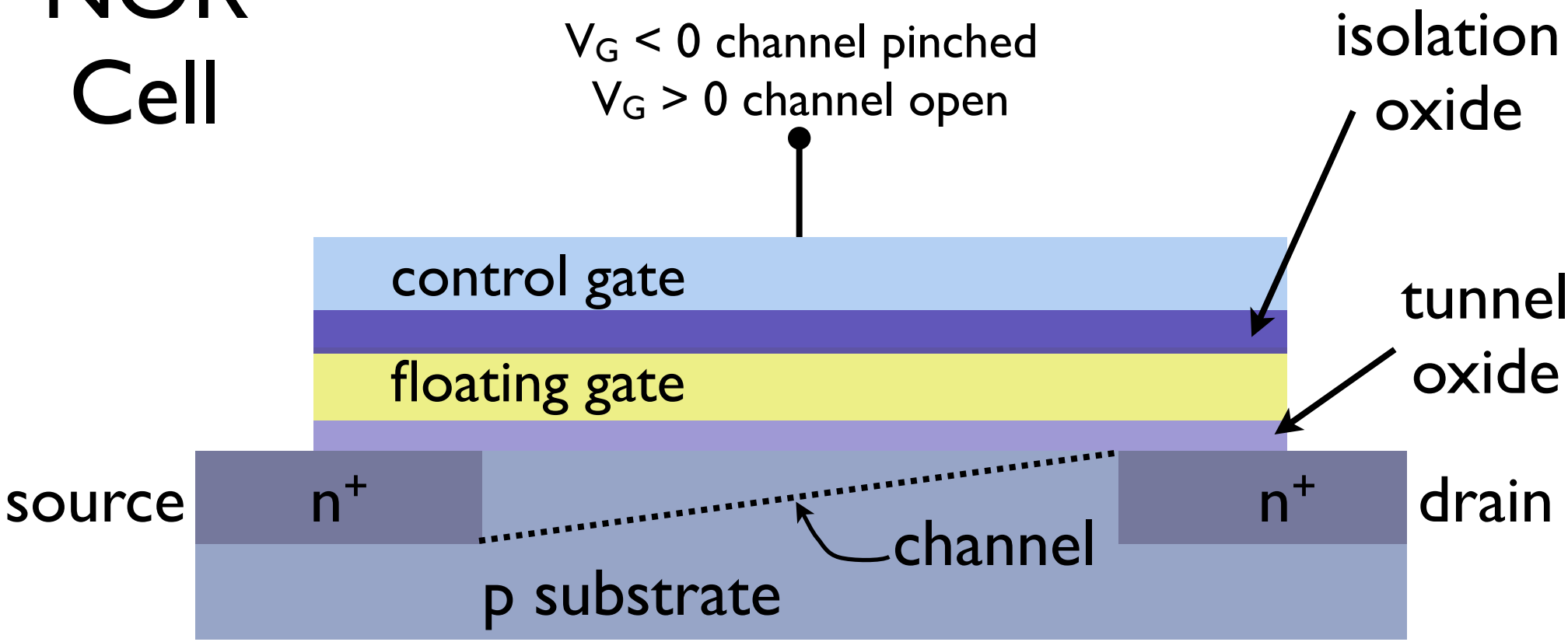
V_G < 0 channel pinched = no current = LO

V_G > 0 channel open = big current = HI

(use a single set supply voltage - forget amplification, just bi-stability)

the basics of Flash

NOR Cell

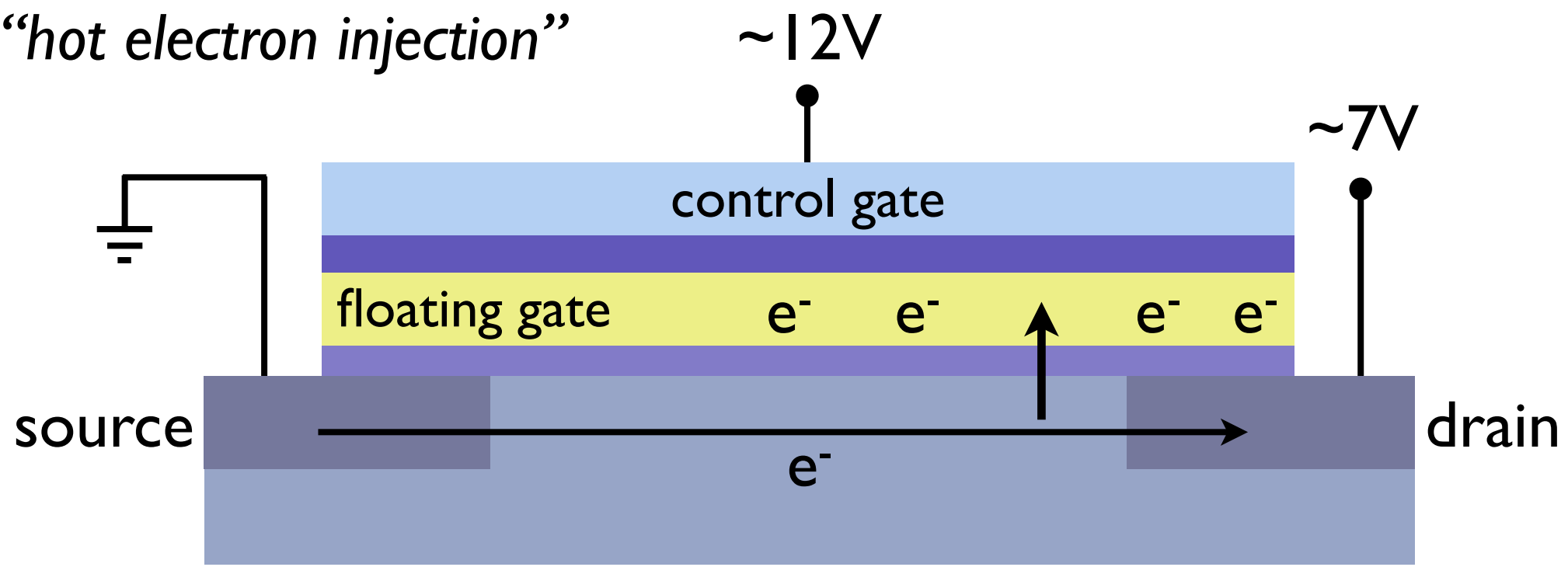


- like a MOSFET
- uses 2 gates

NOR cells are truly random-access, unlike NAND

writing

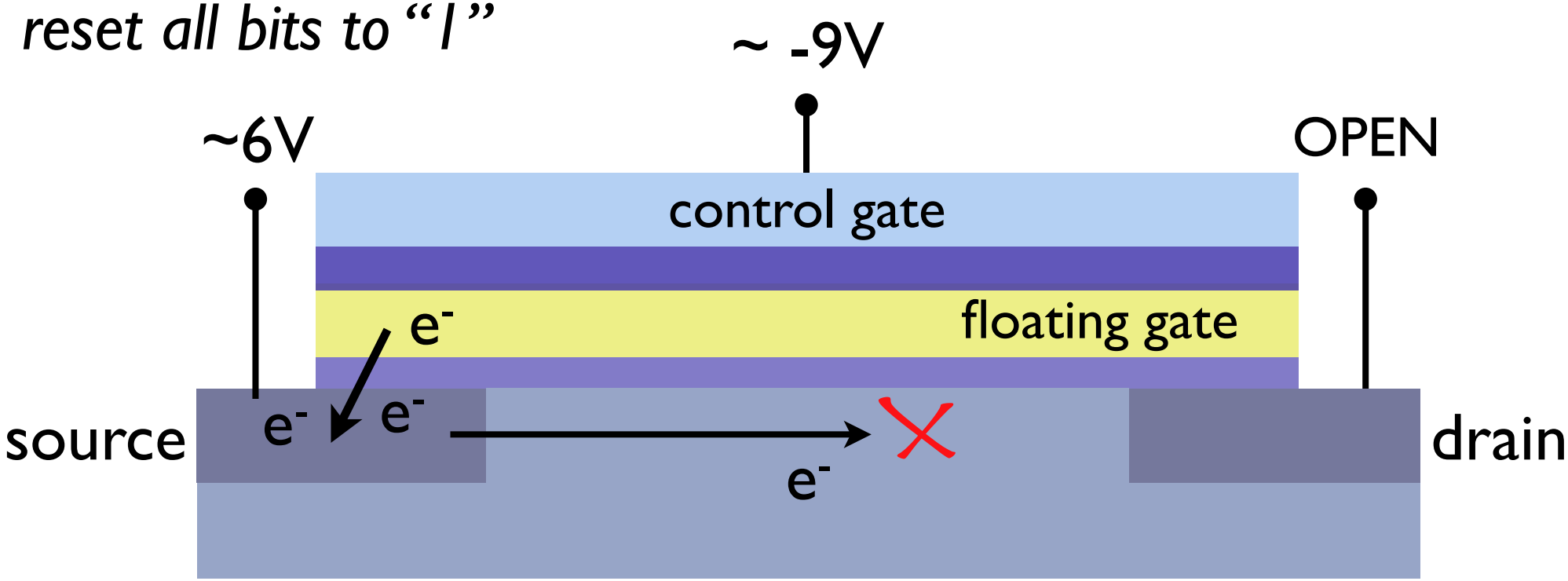
"hot electron injection"



- ~7V to drain
  pull e^- through channel
- ~12V to control gate / open channel
  injects e^- into floating gate through tunnel oxide
- floating gate now charged

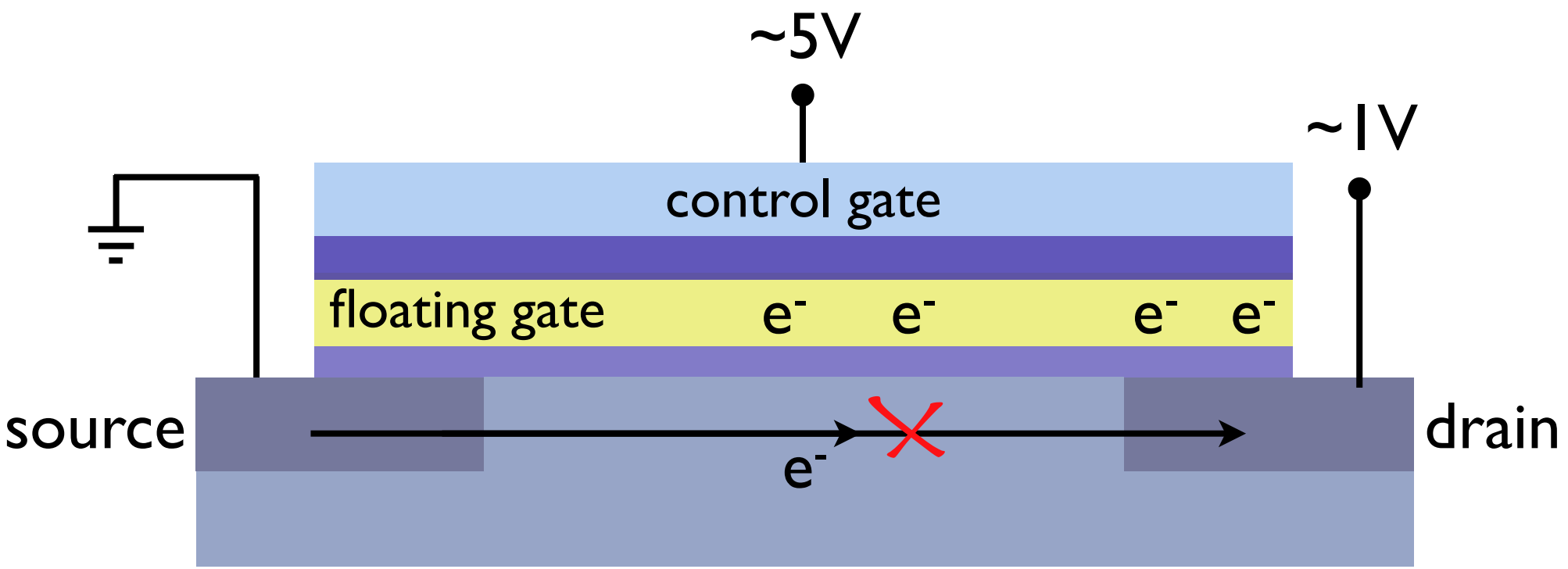
erasing

reset all bits to "1"



- -9V to control
  pinch off channel
- ~6V to source
- suck electrons out of floating gate into source
  Fowler-Nordheim tunneling

reading



- 5V to control
- 1V to drain
- floating gate charged = channel is pinched off = "0"
- floating gate *discharged* = channel open = "1"

  presence of charge modulates I_{SD} !

pros & cons

+ no mechanical limitations

+ lower latency

  = attractive for speed, noise, power consumption, reliability.

- cost/GB still higher (but decreasing rapidly!)

- finite number of erase/write (typically 10^6 cycles guaranteed)

  unable to support an OS (swap!)

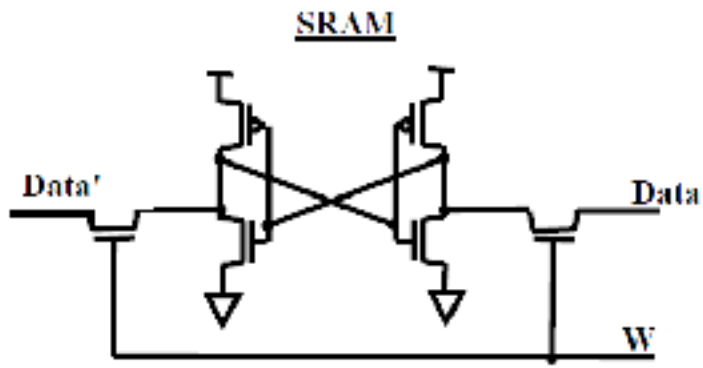
  warranties on flash-based disks trending ≥ HDD

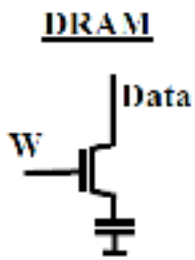
- way too slow for main or cache memory.

S/D ram

- **SRAM** = static ram = store in the state of a flipflop or latch (bistable elements
- **DRAM** = dynamic ram = store via charge on capacitor (more common, simple)
- **FLASH** = store via charge on transistor gate

- **SRAM** = faster but less dense than DRAM (more parts)
- **DRAM** = denser but slower, needs refresh for charge





DRAM

a transistor and a capacitor are paired in a cell

this cell represents a single bit of data.

capacitor holds the bit of information -- a 0 or a 1

transistor acts as a switch

lets the control circuitry on the memory chip read the capacitor or change its state.

http://irvs-embedded-projects.blogspot.com/2010/09/flash-memory-and-ram-random-access.html

One can also do this with resistors.

bistable resistor needed
- set state with voltage/current
- read with small current

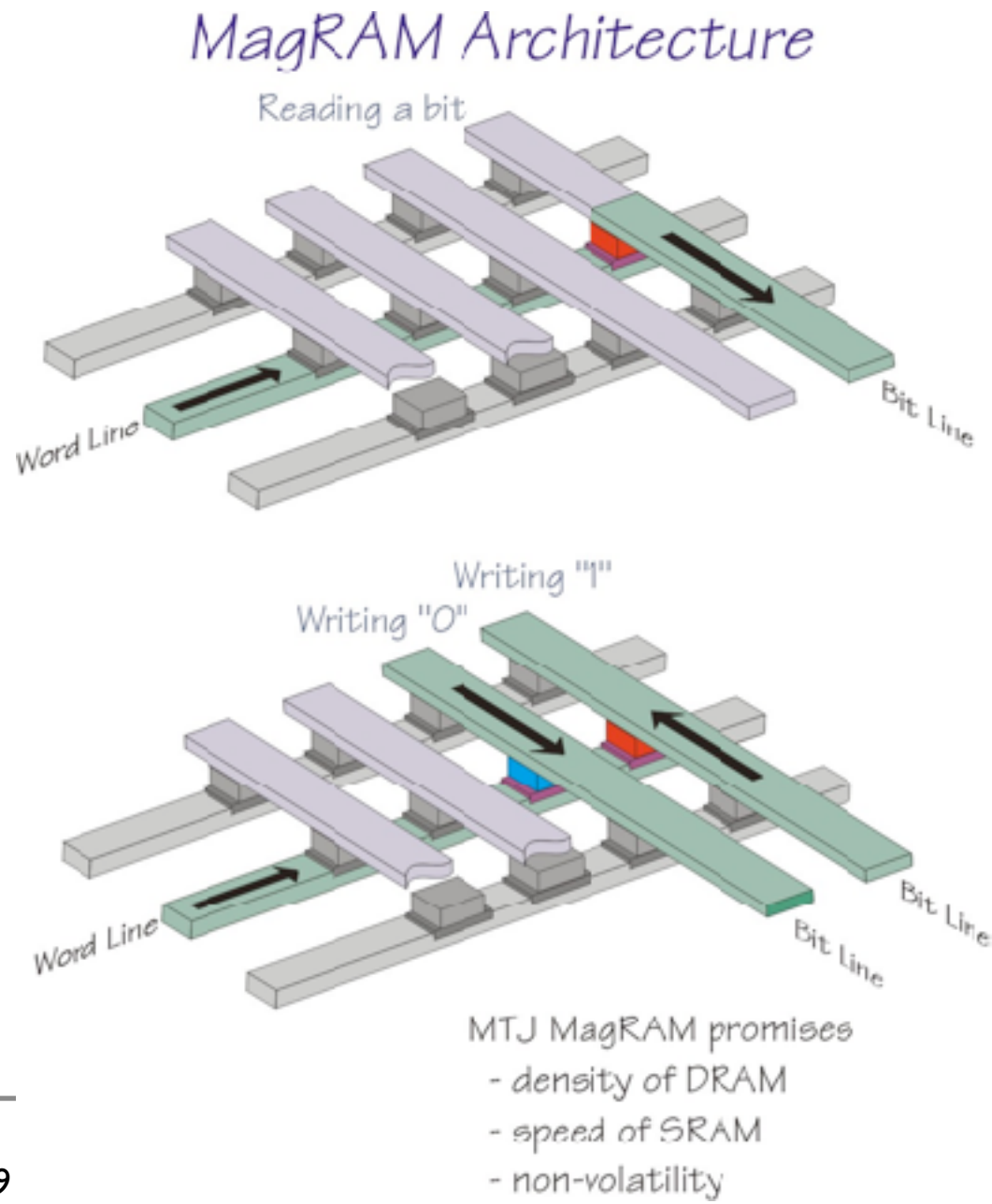
can do this with magnets

.
.
.

MRAM

could be a true universal memory!
(problem: "sneak paths" in grid; diodes needed)

pic: research.ibm.com



Of course, there are details

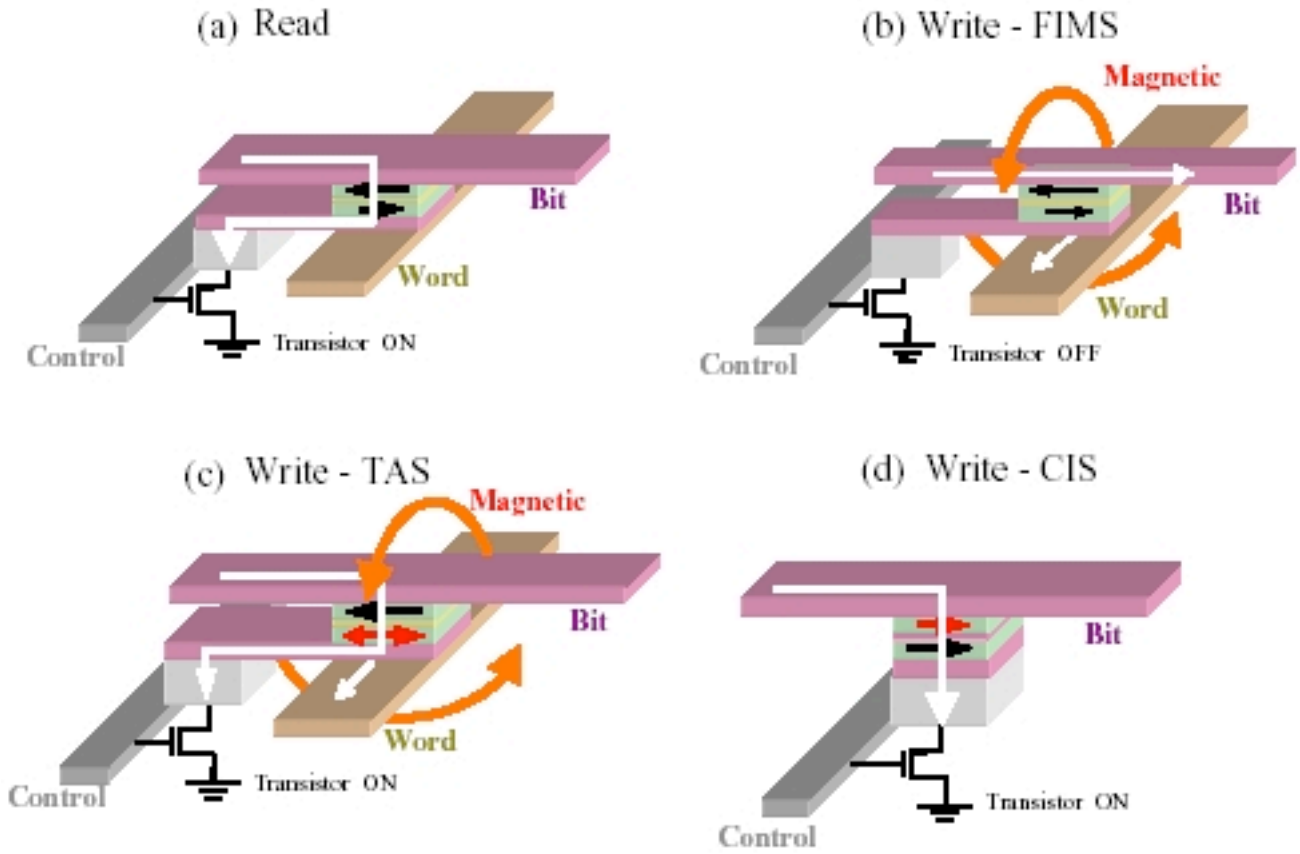


Figure 2: Read and write principles of MRAM with various architectures: (a) Read (common principle); (b) Write in FIMS (Field Induced Switching) mode; (c) Write in Thermally Assisted Switching (TAS) mode; (d) Write in Current Induced Switching (CIS) mode)

pic: http://www.spintec.fr/spip.php?article53

in any of these cases ...

SRAM, DRAM are volatile memories

– bits are not held without power supply.

– generically, a grid addressing structure

– crossing lines: use 1 horz and 1 vert line to uniquely access any cell

MRAM is nonvolatile

– no refresh or power required to retain bits

• crossing lines are 'word' and 'bit' lines.

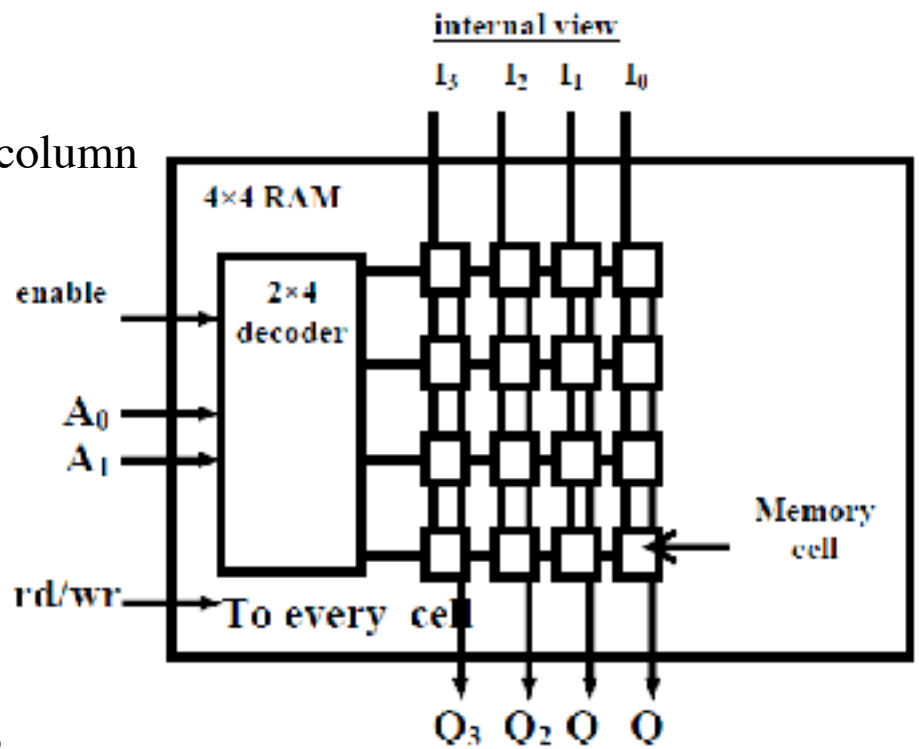
• decoder required every N^2 cells

– a word consists of several memory cells, each storing 1 bit

– each input and output data line connects to each cell in its column

– rd/wr connected to every cell

– when row is enabled by decoder, each cell has logic that
  stores input data bit when rd/wr indicates write
  outputs stored bit when rd/wr indicates read



http://irvs-embedded-projects.blogspot.com/2010/09/flash-memory-and-ram-random-access.html

How to use transistors for logic?

NAND gate

- all mathematical operations can be derived from addition
- we can do general addition in binary
- thus, we can use HI/LO voltages as 1/0 for all math.

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

- What do you need? Switches, basically.
- transistor = voltage-controlled switch!

- All other Boolean logic gates (i.e., AND, OR, NOT, XOR, XNOR) can be created from a suitable network of NAND gates.

- Actually AND, OR, and NOT are enough for all logic functions

- A **logic gate** performs a logical operation on one or more logic inputs and produces a single logic output.
  - e.g., 1 AND 1 = 1

pic/table: wikipedia

AND & NAND (negated AND) gates

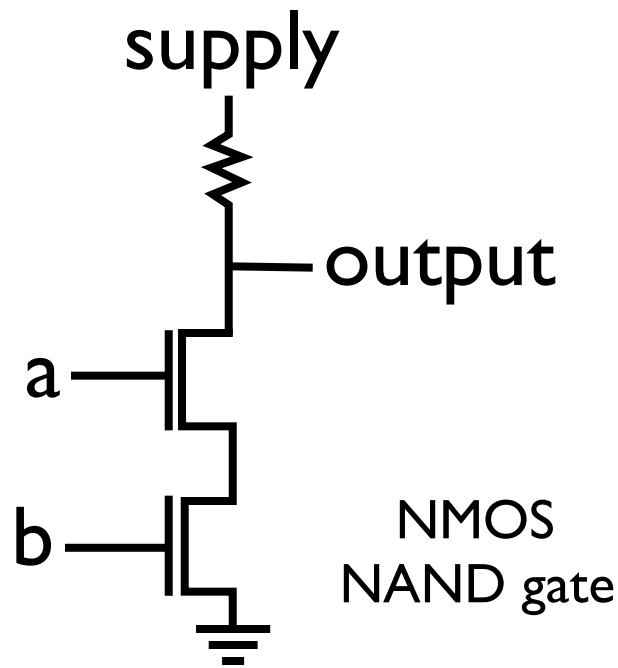
AND
(are both 1?)

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

NAND
(inverted AND)

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

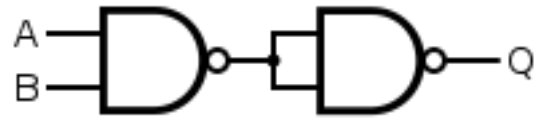
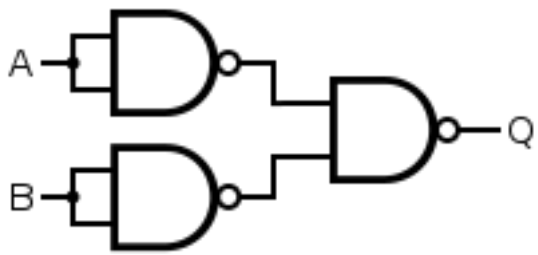
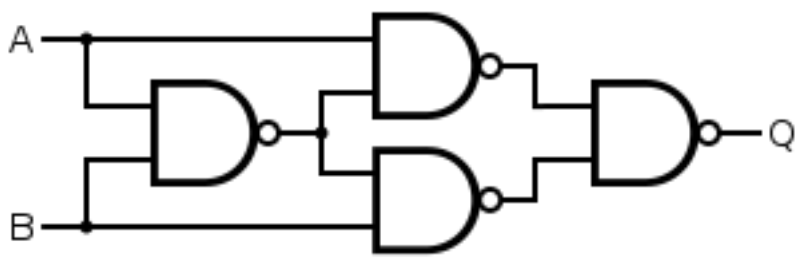
can chain two
NANDS together
to make AND



pics/tables: wikipedia

Gate simulator: http://www.neuroproductions.be/logic-lab/index.php?id=52

All from NAND ... simple, modular construction

Gate		NAND construction
	AND (are both 1)	
	NOT (invert)	
	OR (is any a 1)	
	XOR (is only 1 a 1)	

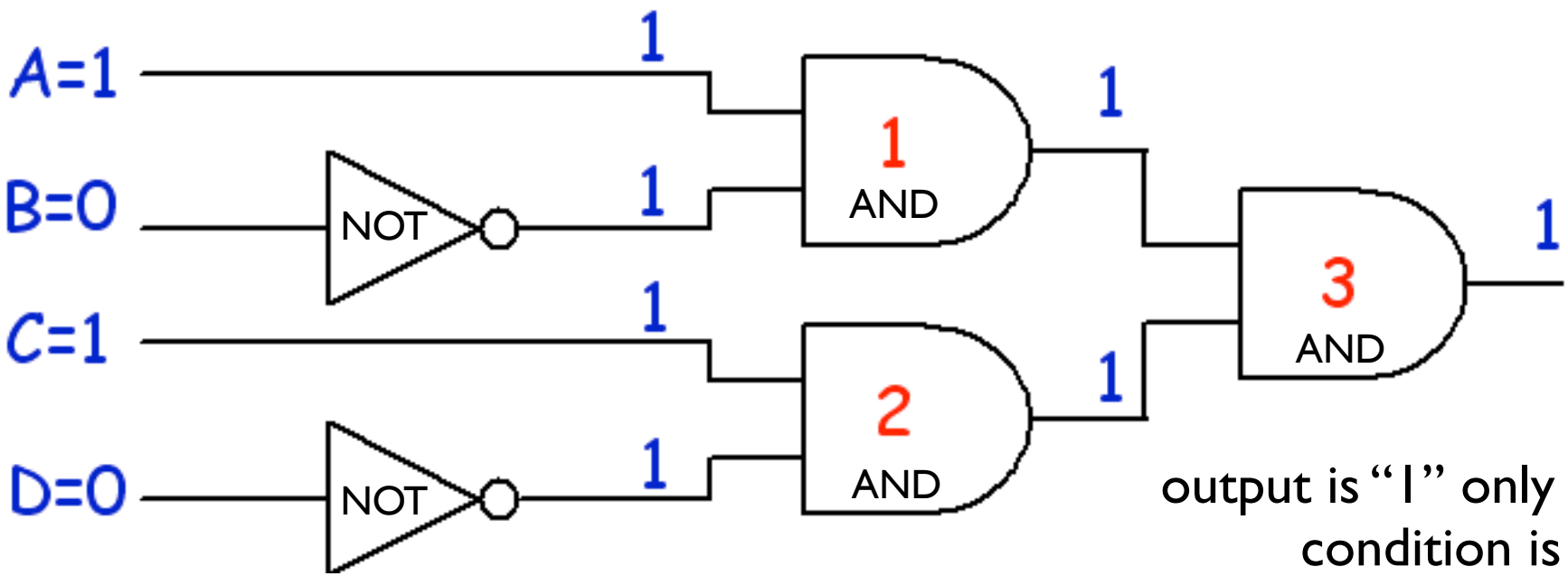
pics: wikipedia

(A NAND (A NAND B)) NAND (B NAND (A NAND B))

Example computation: is the input equal to a certain value?

four signals A-D, each is one bit (hi/lo or 1/0)
as a group: four digit binary number DCBA.

say we want to detect when these four digits form a certain value
e.g., is DCBA=$0101_2=5_{10}$
need four inputs, two inverters (NOT), and three ANDs



output is "1" only when input condition is met!
switch any input, output is zero

http://knol.google.com/k/max-iskram/digital-electronic-design-for-beginners/1f4zs8p9zgq0e/23#

Example computation: addition

Computers are adding machines
- subtract = add with invert
- multiply by serial addition
- all math can be derived from adding

The circuits they use are based on the **half-adder**.

This incorporates with the rules for binary addition

$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 0 \text{ carry } 1$$

binary review

3	2	1	0	digit
2^3	2^2	2^1	2^0	power of 2
1	0	1	1	binary

decimal: $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

The gate needs two outputs, a **sum** and a **carry**.

sum is the output of an XOR gate (we can't have 1 + 1 = 1, ruins carry)

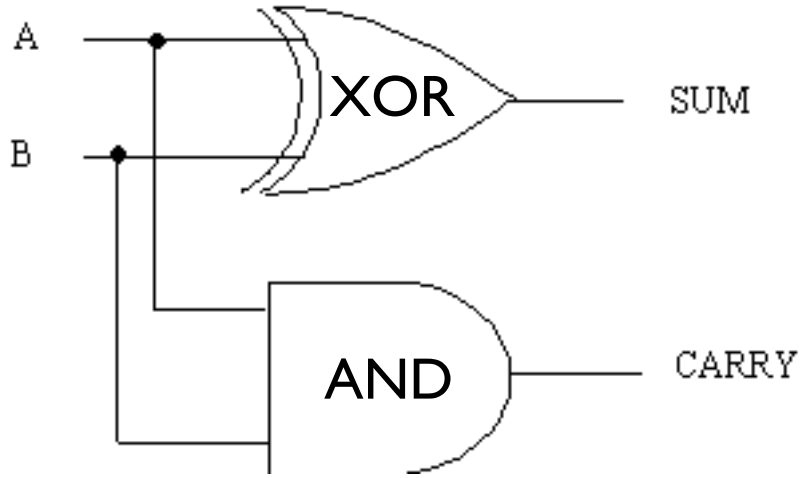
carry output is an AND gate.

http://www.antonine-education.co.uk/Electronics_AS/Electronics_Mod2/topic_2_1/using_nand__or_nor_gates_to_make.htm

XOR

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

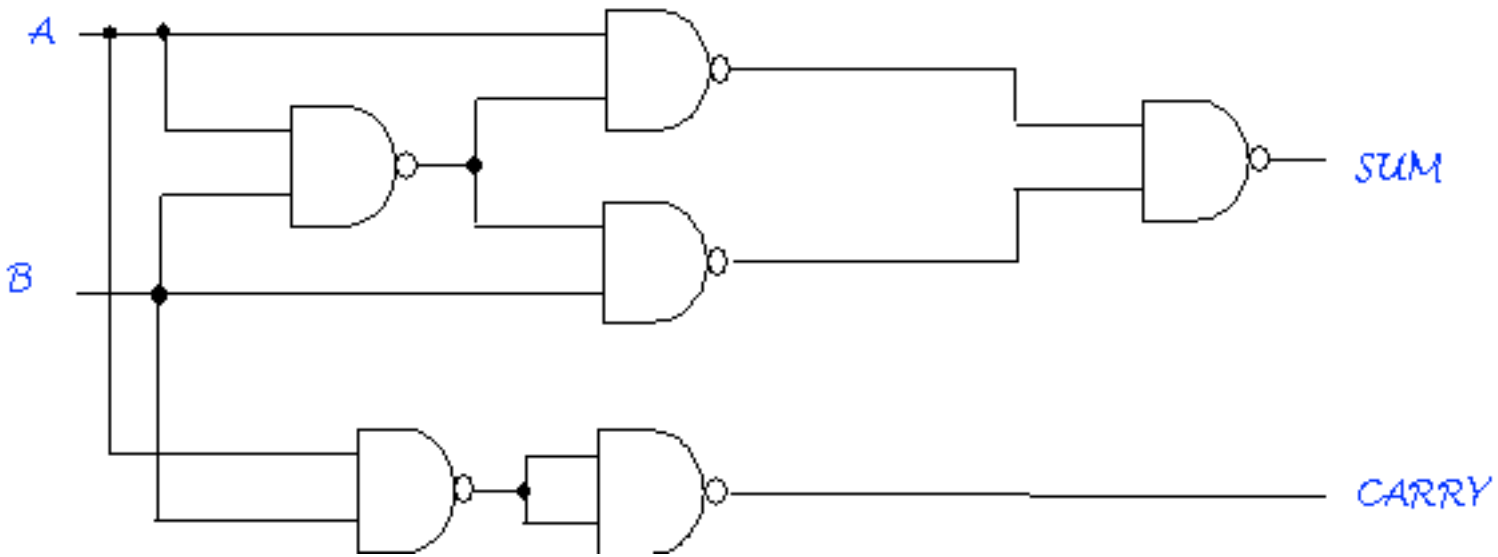
add
except for carry to
next digit, correct



AND

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

carry
only non-zero if
both inputs high,
i.e., when needed



all NAND
version

http://www.antonine-education.co.uk/Electronics_AS/Electronics_Mod2/topic_2_1/using_nand__or_nor_gates_to_make.htm

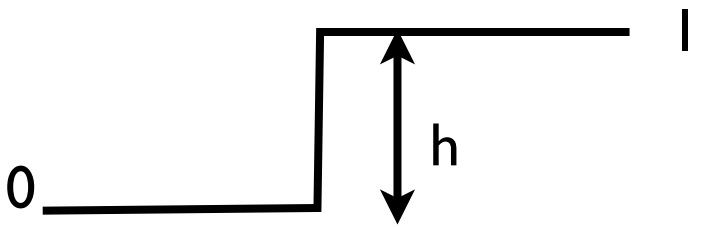
A quick overview of information physics

power dissipation is a huge bottleneck
for continuing Moore's law

what are the fundamental limits?

energy per bit = energy required to change state
(aside from refresh, transistor power, etc)

A quick overview of information physics



$$\Delta E = mgh$$

say we want to store a bit physically.

just need 2 distinct potential states.

could be gravitational!

$$P \sim e^{-\Delta E / k_B T}$$

probability of thermal energy inducing accidental switch:

$$P \sim e^{-3\times 10^{16}}$$

1 gram, 300K, 1cm

$$P \sim 10^{-14}$$

H atom, 2nm

somewhat impractical.
stability criteria: energy diff ~40kT

Energy difference needed per bit?

Shannon: energy per bit is **kT ln2** as theoretical minimum
we are *way* above this. but consider errors ...

say <1 per 10 years at 10^{10} operations per sec
10^{-29} failure probability implied

at the minimum we have

$$e^{-\Delta E/k_B T} \sim 10^{-2k_B T \ln 2/k_B T} \sim 4\%$$

not enough for thermal stability in real world

need energy diff of at least ~30 times this for proper stability
i.e., want 40kT energy diff between states for stability
~1 electron volt per charge (order of magnitude)

How about charge? (DRAM, FLASH)

cap charged = 1
cap discharged = 0

$$\Delta E = QV = \frac{1}{2}CV^2$$

1 electron at 2V = 2eV/charge = 80kT at room temp!

probability of accidental switch ~ 10^{-43}

say 10^{10} transistors, cycling at 10GHz (charges/sec)
after 10^9 seconds (30 years), 10^{29} operations

VERSUS 10^{-43} failure rate! good enough?

(problem: statistical noise due to discrete charges)
(if we use at least a 100, only ~10% stat noise)

How about spin? (Hard disks)

up or down spin in a magnetic field $\Delta E = 2\mu_B B$

$\mu_B \approx 60\mu eV/T$

stability requires field: $B \sim \frac{40k_B T}{2\mu_B} \sim \frac{0.5eV}{60\mu eV/T} \sim 8600T$

permanent magnets ~ 1T

use more spins.
40nm cube ~ 10,000 spins

$B \sim \frac{40k_B T}{2 \times 10^4 \mu_B} \sim 0.86T$

doable! flip side: this is the *limit* for magnets

Energy loss per step of computation?

energy for storage is one thing.
how much will we dissipate per clock cycle?

$$\text{energy loss/step} = \text{kT}\,\frac{\text{min time taken/step}}{\text{time/step actually taken}} = \text{kT}\,\frac{\text{max speed}}{\text{actual speed}}$$

reason for speed throttling on processors ...
say 3GHz possible, run at 1GHz.
real step = 1ns, max ~0.3ns
kT ~ 25meV at 300K

$$\sim 8\text{meV/step} \sim 10^{-12}\,\text{W/comp} \qquad \sim 1\text{mW/sec}$$

clearly not the main source of power dissipation ... a lot of room!

Recall from our radiation discussion ...

$$Q = 2\pi \frac{\text{total energy of oscillator}}{\text{rate of energy loss per radian}} = \omega_o \frac{\text{energy stored}}{\text{power loss}}$$

our computer is basically an oscillator ...
say we have 10^9 bits, 40kT per bit: ~ 0.1nJ to store per cycle
our loss is ~1pJ/s, frequency 1GHz

$$Q \sim (2\pi \text{GHz}) \frac{0.1\text{nJ}}{1\text{pJ/s}} \sim 10^{11}$$

comparing: about as good as a laser cavity, way better than most circuits.

So: Moore's "law"?

fundamentally, we are fine for a while

from physics & information theory, not close to hard limits
we are close to hard limits for our current *specific* technologies
hard disks, RAM, FLASH, transistors in general

fundamentally new *technologies* are needed
this may or may not require new fundamental science
it *will* require fundamentally new engineering

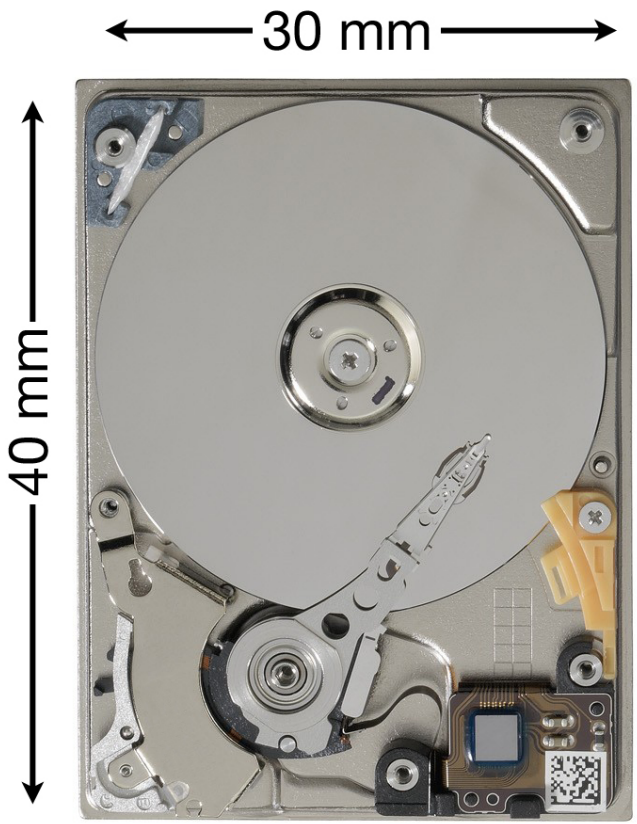
How about hard disks?

how do hard disks work, more or less?

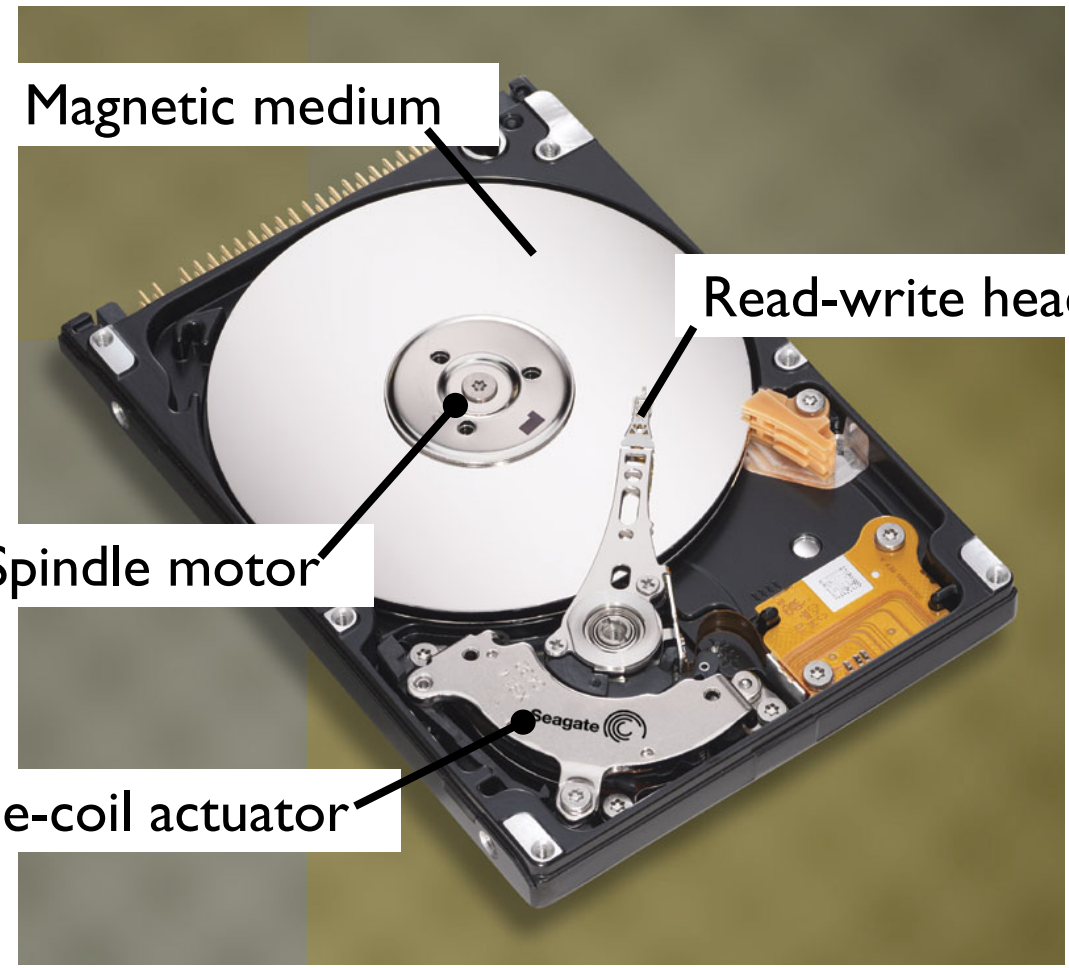
wikipedia.org - "Hard_Disk"

spinning (~10^4 rpm) part holds data.
sliding part reads and writes data.

hard disk drives



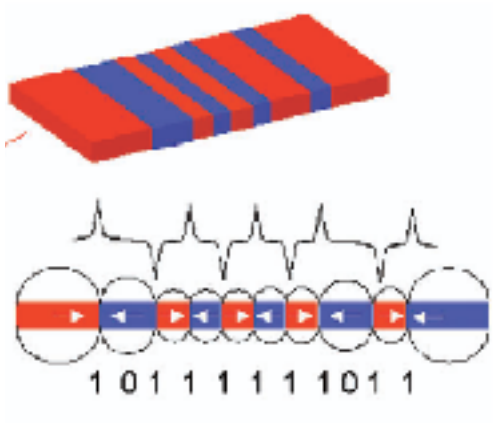
8 Gbit 1" drive for cameras

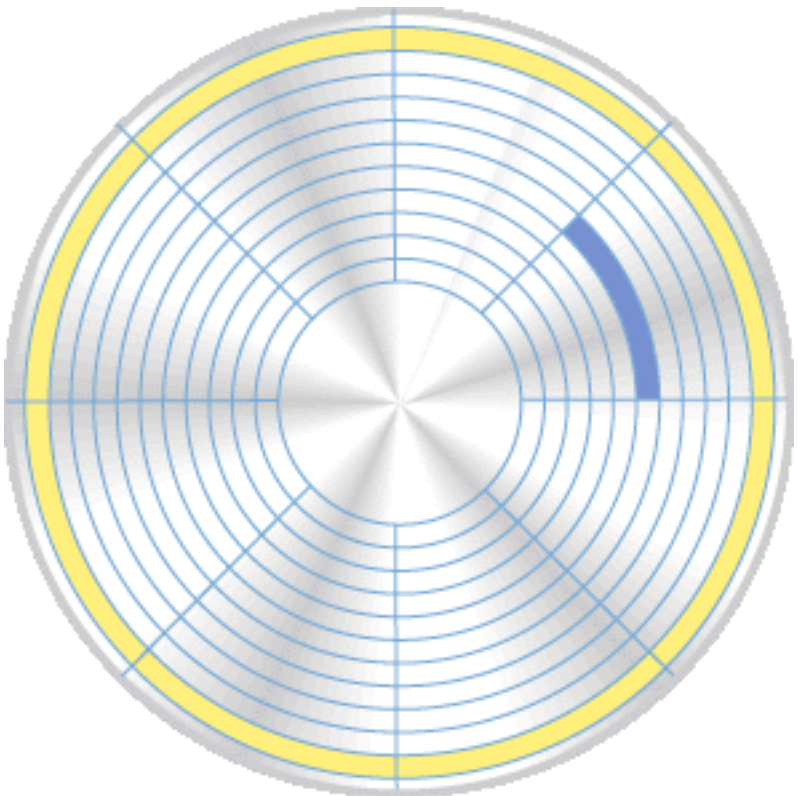


160 Gbit 2.5" perpendicular drive for laptops

images from M. Coey

media basics

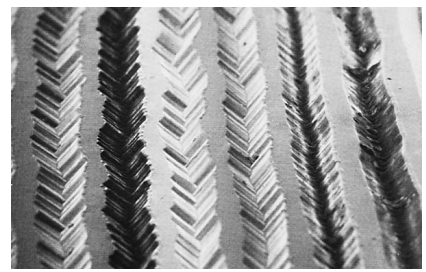




Hard disk
- tiny magnetized regions
- direction (N/S) stores bit
- magnetic sensor reads bits

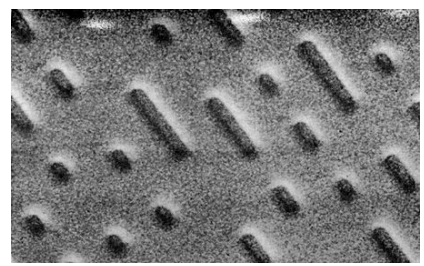
LP records
- tiny bumps
- needle moves

actual record grooves



CDs
- pits store bits
- optical reflectivity

actual CD surface



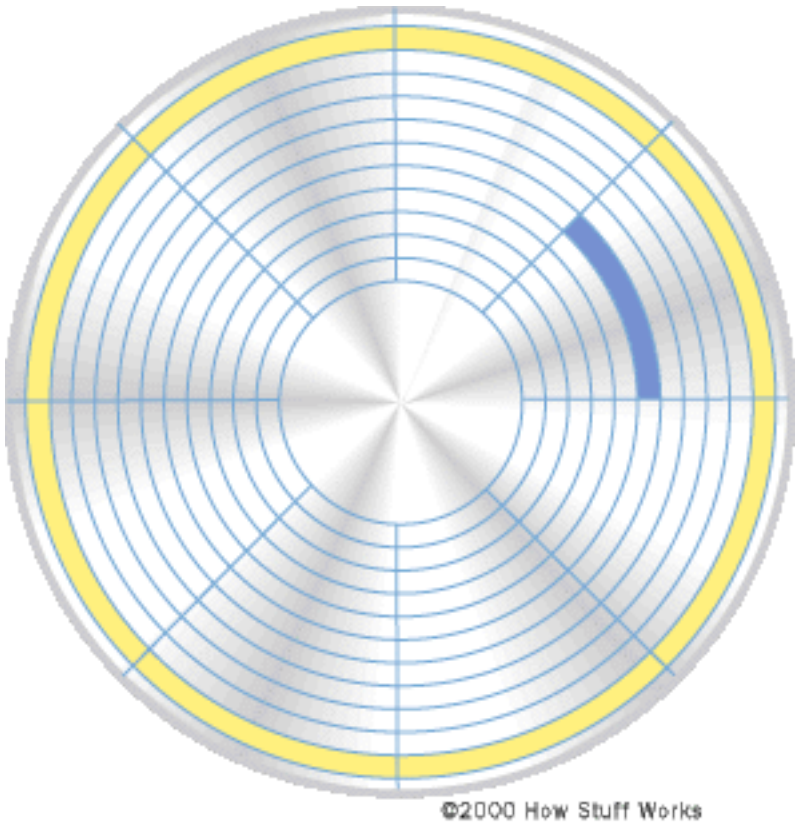
media basics

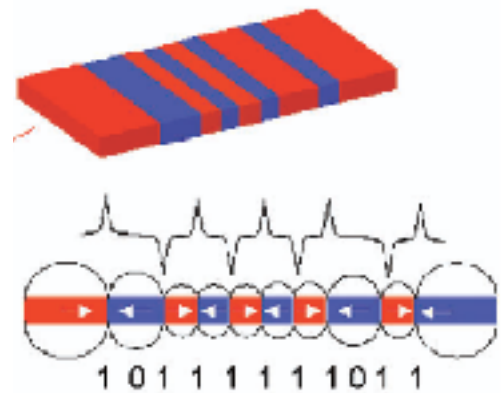
hard disk platters are round.

so how is data arranged?

tracks = concentric circles
sectors = wedge of a track

sector has fixed # bytes

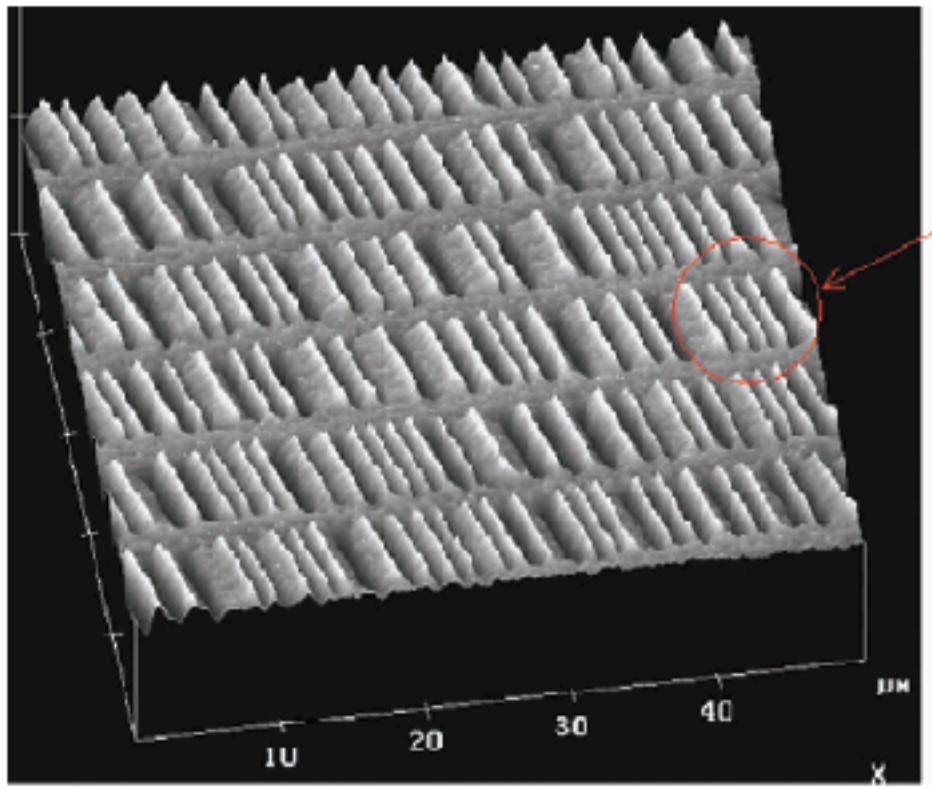


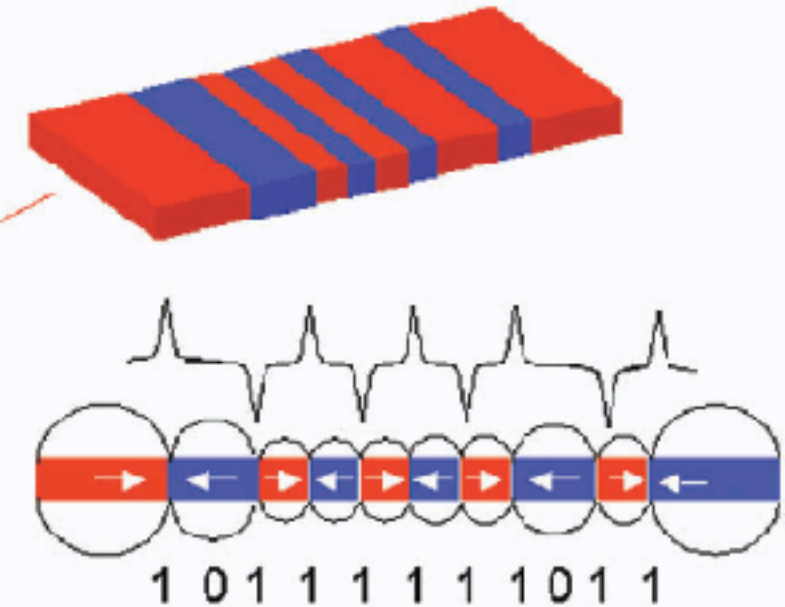


media basics

mfm image

sees transition field





Jimmy Zhu, *Materials Today*, July/Aug 2003

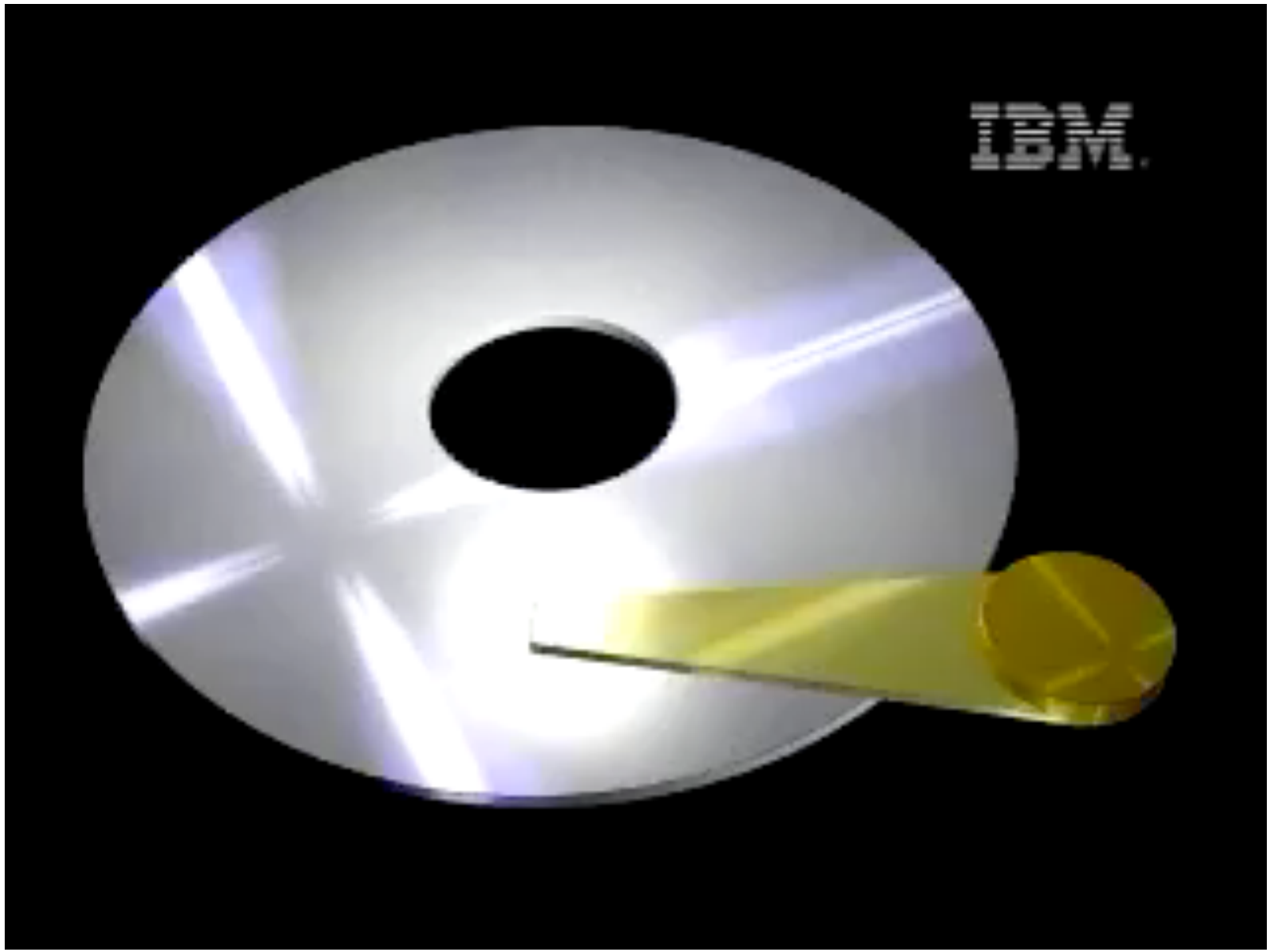
CoCrPt alloy

platters - Al or glass substrate

typical magnetic region

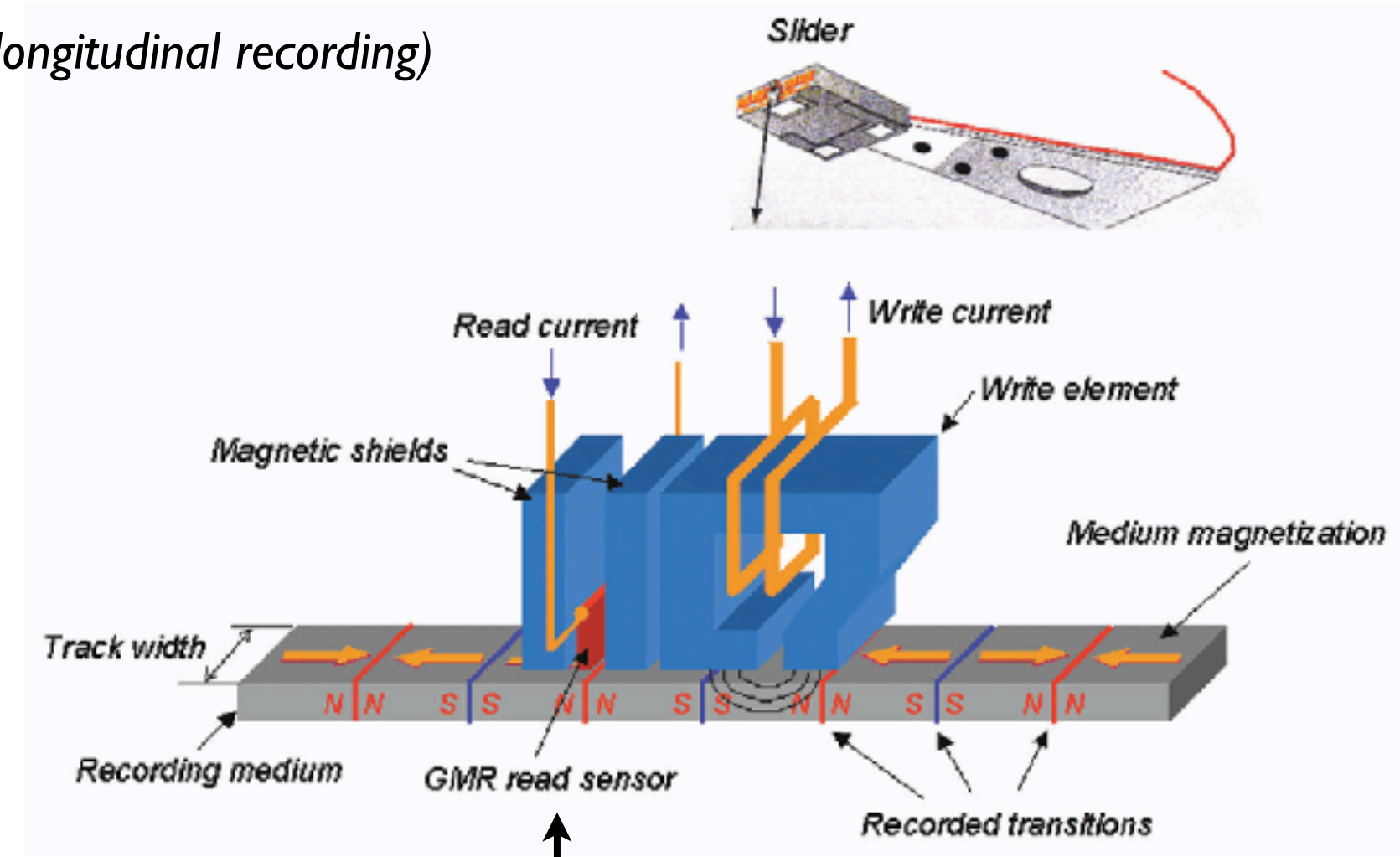
~200-250 nm wide, ~25-30 nm down-track

100 billion bits (Gigabits) per in^2


IBM.

reading and writing basics

(longitudinal recording)



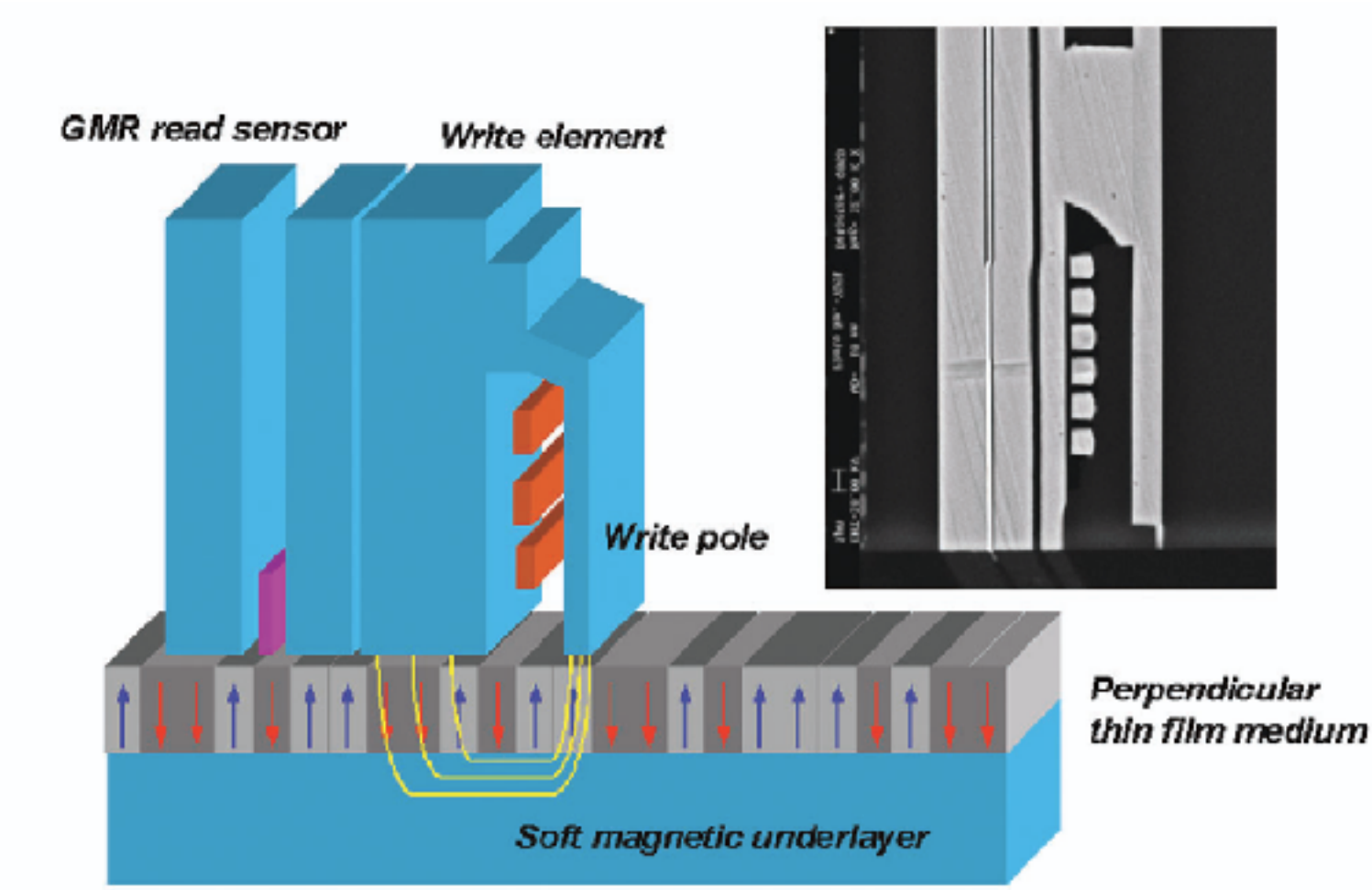
sensor - *magnetoresistive*

Jimmy Zhu, *Materials Today*, July/Aug 2003

reading and writing basics

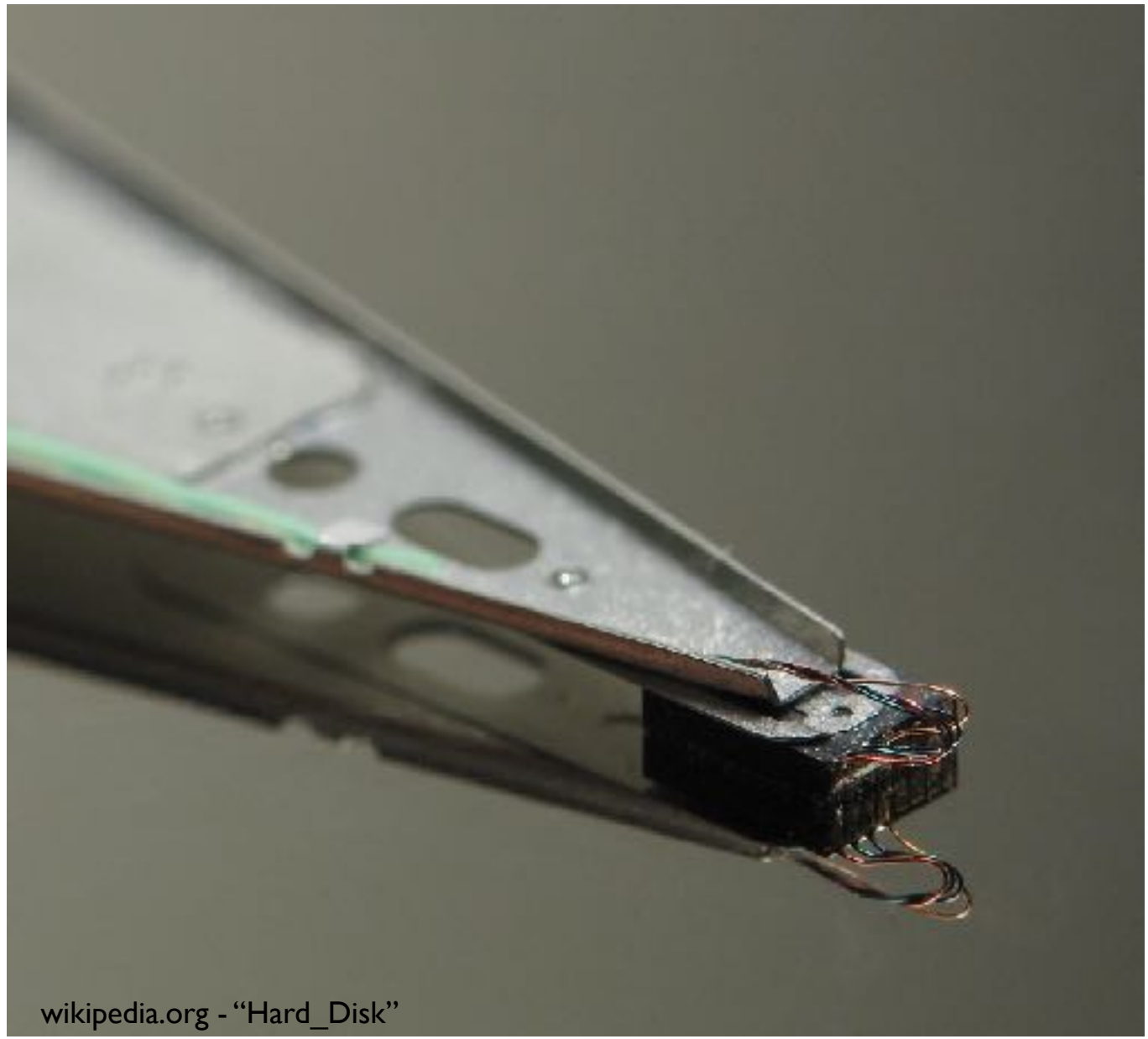
(perpendicular recording)

Jimmy Zhu, *Materials Today*, July/Aug 2003



soft underlayer becomes part of the flux guide
... careful concentration of flux ...

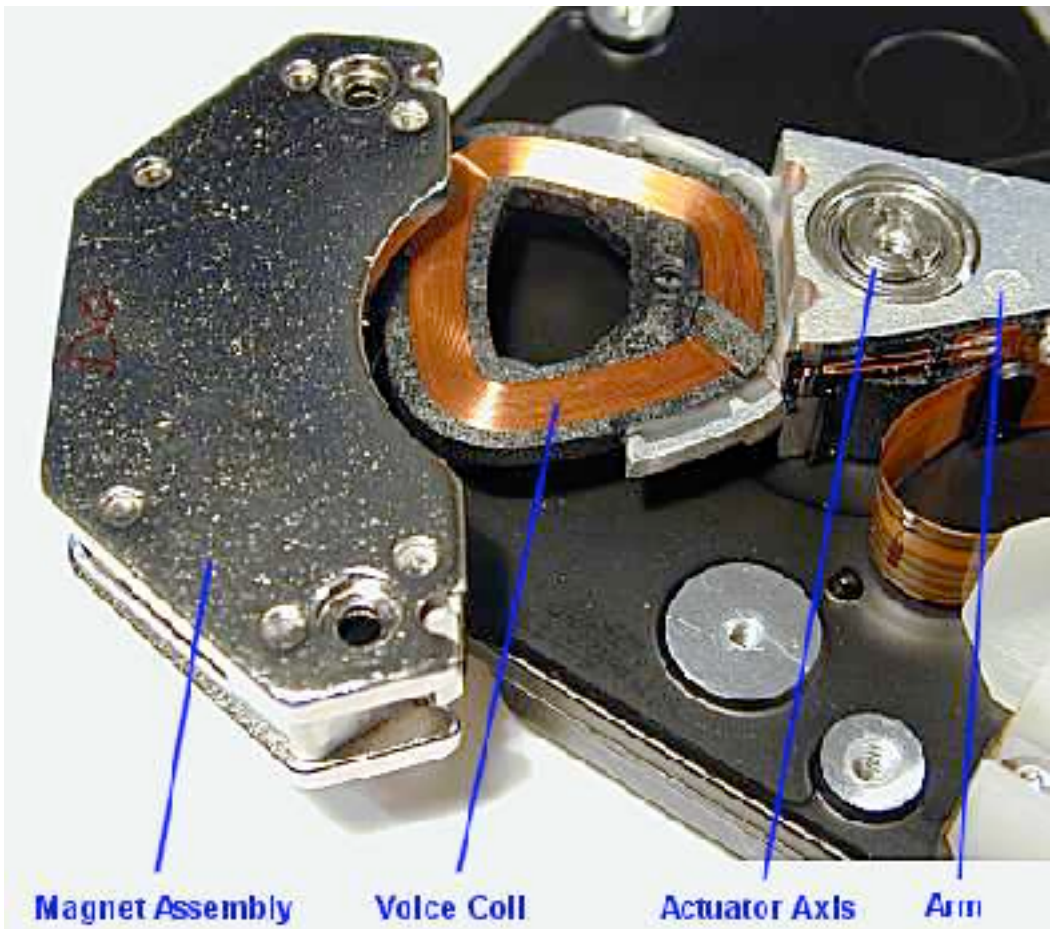
read head (and its reflection)



wikipedia.org - "Hard_Disk"

positioning basics

- current powers voice coil†
- field generated moves head L or R
- more precise than stepper motor



www.pcguide.com/ref/hdd/op/actActuator-c.html

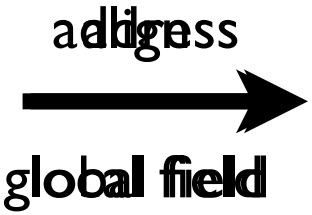
IBM 62PC "Piccolo" HDD, ~1979 - an early 8" disk

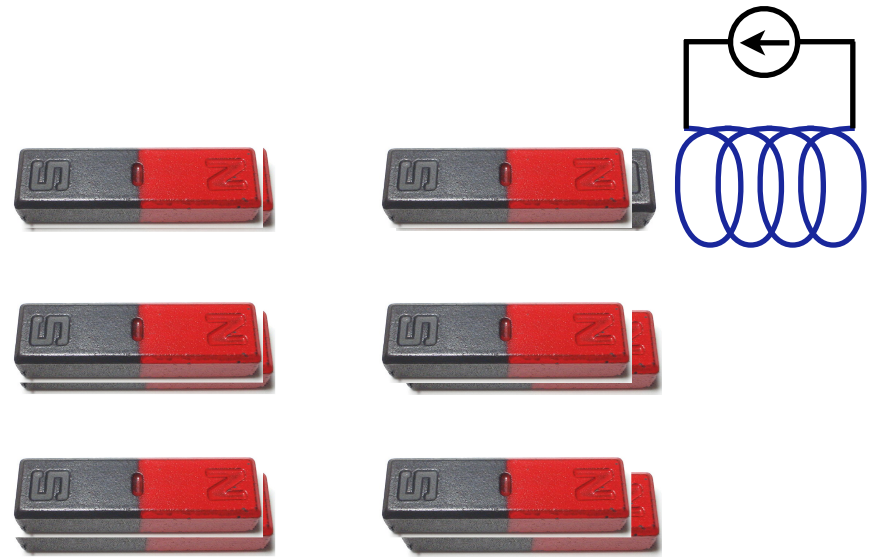
wikipedia.org - "Hard_Disk"

† this is the same way a speaker cone moves

why magnets?

microscopic view





magnets remember their state

once magnetized, they stay that way

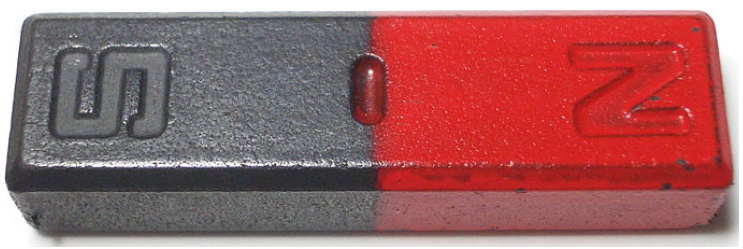
with a little bit of energy, we can control them

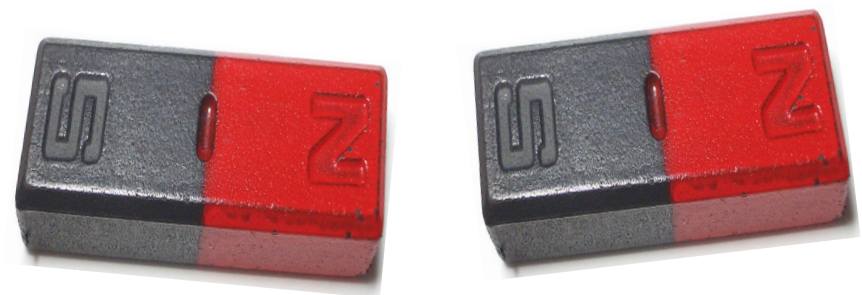
switch from N to S

why magnets?

what happens when
you break a magnet?

you get two magnets





now: do this 25 more times

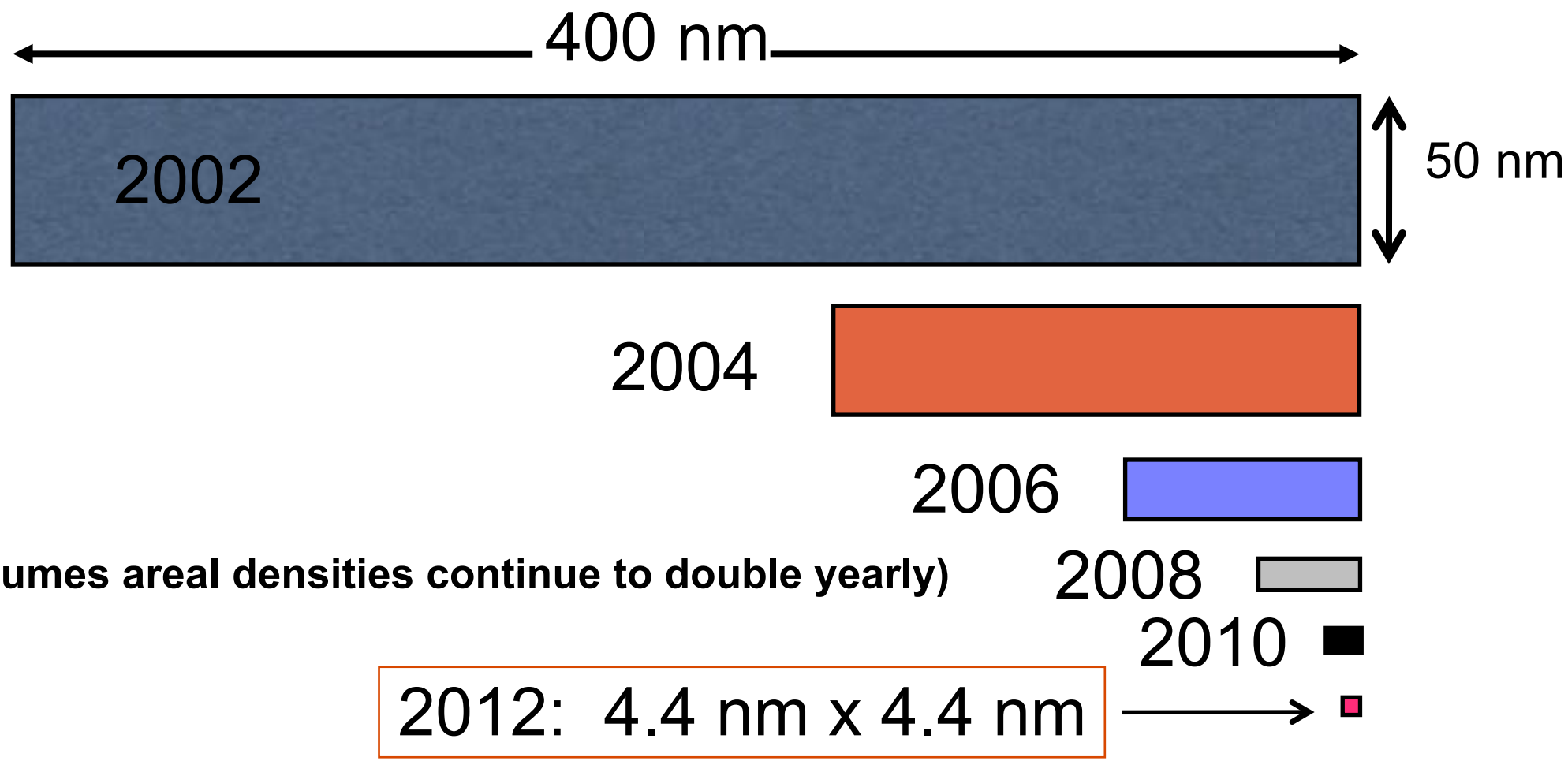
➠ 33 million magnets, all 50nm across

about 1,000 times thinner than a hair

we can make *really tiny* magnets

smaller is *better*, to a point

The incredible shrinking bit!
Predicted relative sizes of HDD storage bits



(assumes areal densities continue to double yearly)

so what's the problem?

at some point, they are no longer stable

heat makes them 'wiggle'

like drops of water on a griddle

bits are no longer reliable

so we need stronger magnets ...

... which need more field to magnetize

... which needs more power

HUGE challenge in nanoscale materials science!

power consumption is not an advantage

latency ...

fundamental limits of magnetism & thermal stability?