CS 250
VLSI System Design

Lecture 5 – System Context

2010-9-20

John Wawrzynek and Krste Asanovic´

with John Lazzaro

TA: Yunsup Lee

www-inst.eecs.berkeley.edu/~cs250/

Intel Sandy Bridge: IDF 2010

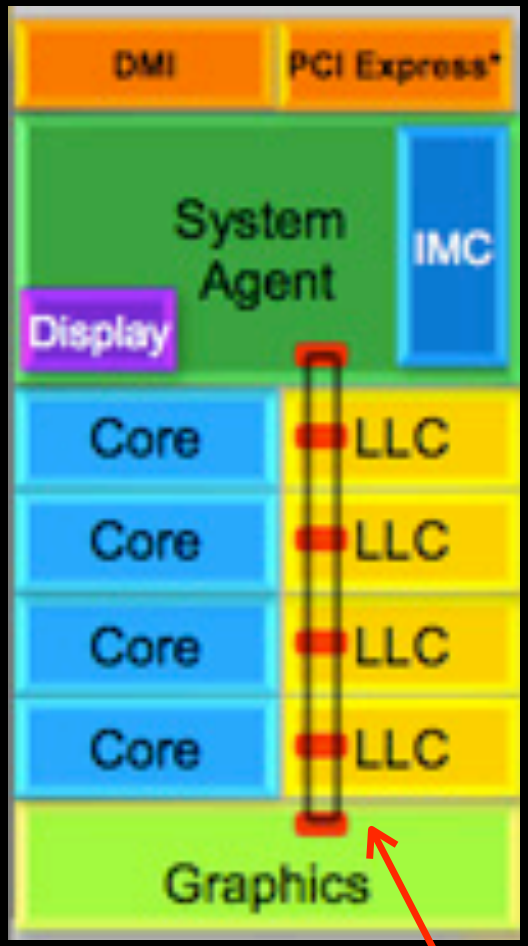
All on chip:

4 x86 cores

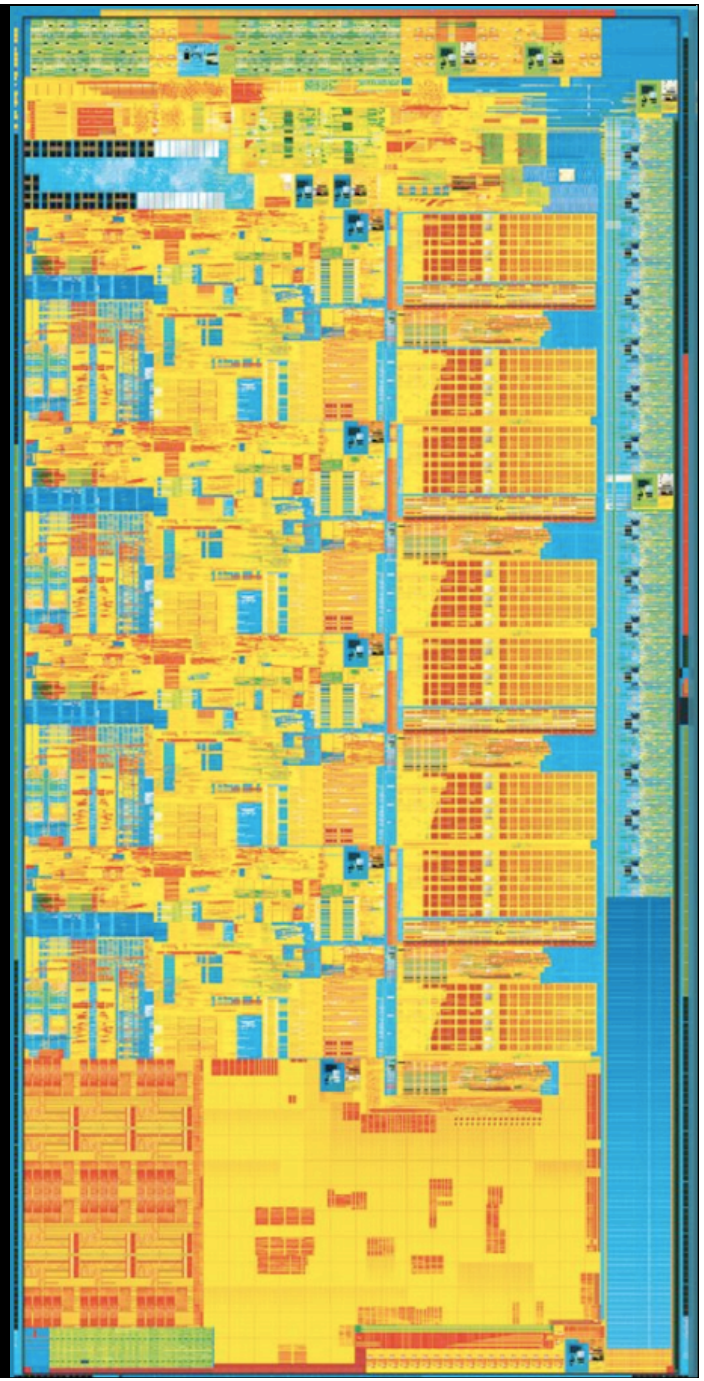
GPU

North Bridge

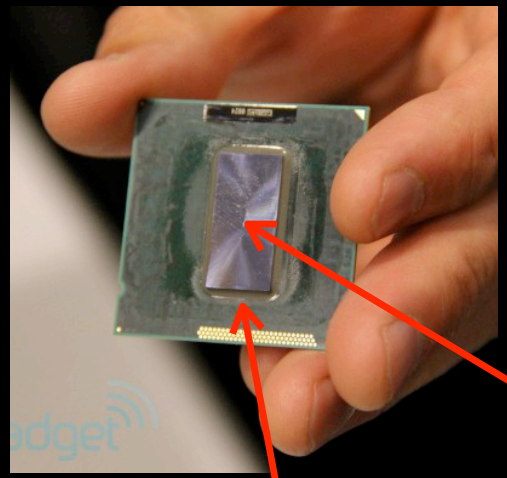
DRAM controller

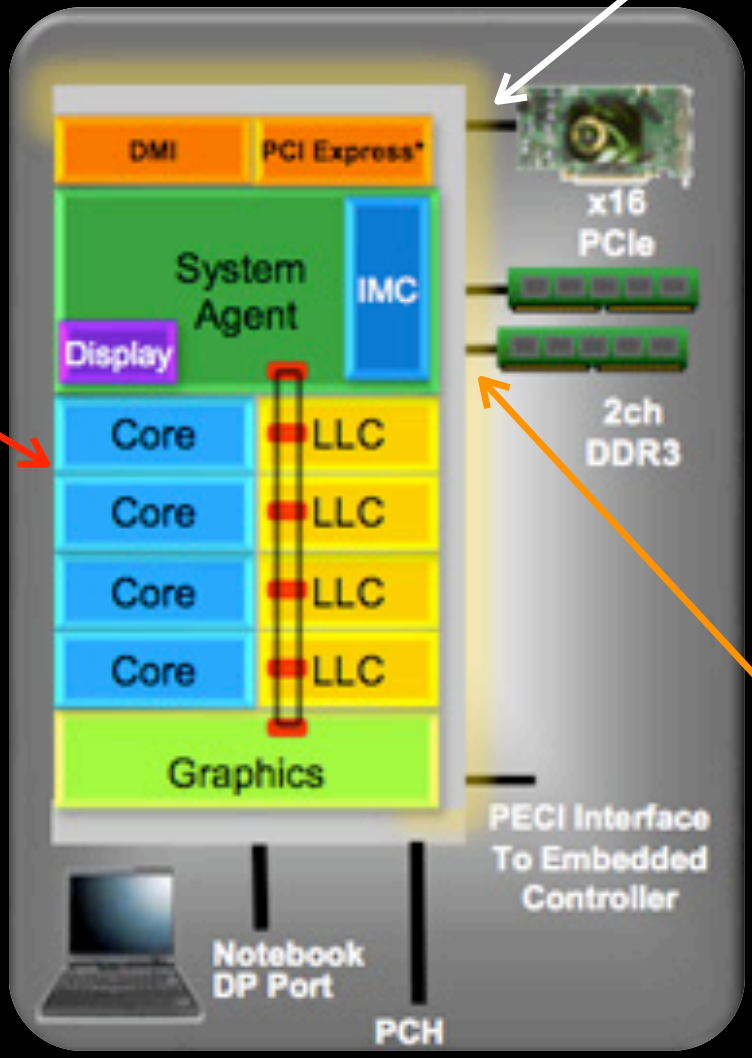


On chip ring network



Today: serial I/O + DRAM



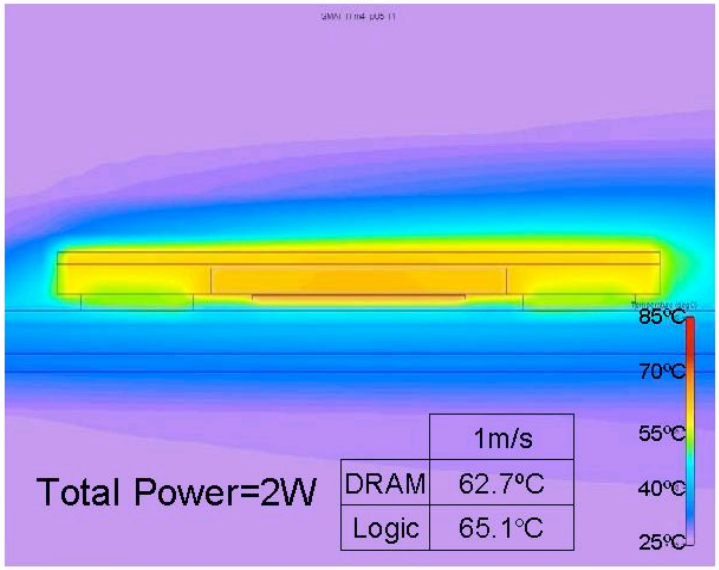


PCIe: At the bottom, serial point-to-point links.

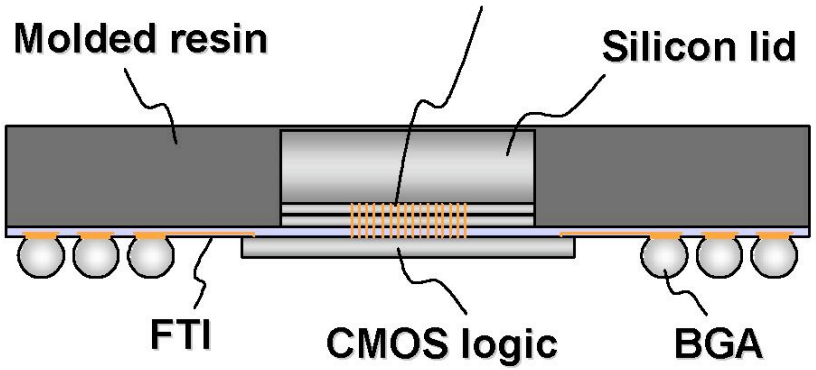
SDRAM: A parallel bus with a single master.

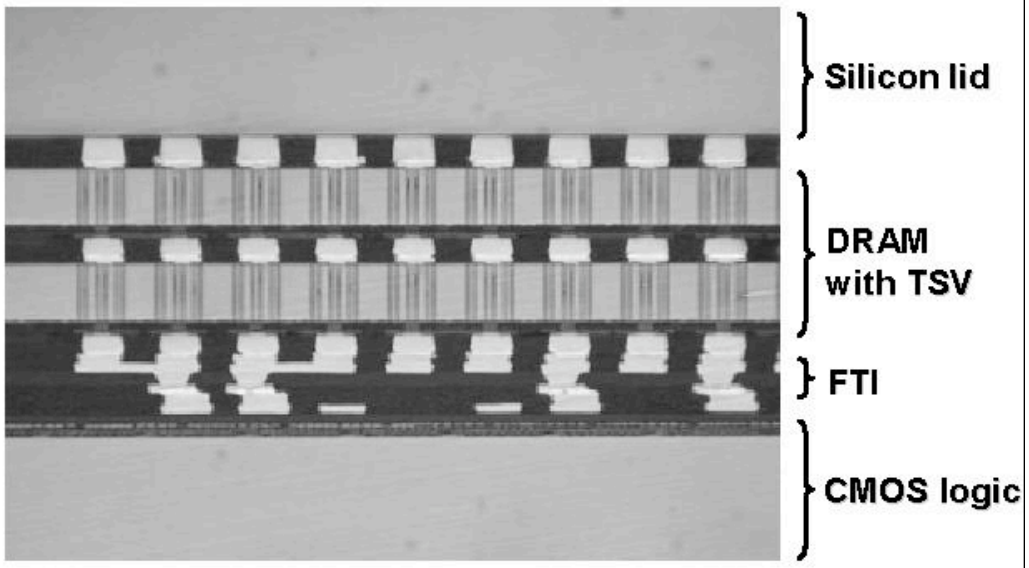
Beyond PC boards

DRAM die size	10.7 mm × 13.3 mm
DRAM die thickness	50 μm
TSV count in DRAM	1,560
DRAM capacity	512 Mbit/die × 2 strata
CMOS logic die size	17.5 mm × 17.5 mm
CMOS logic die thickness	200 μm
CMOS logic bump count	3,497
CMOS logic process	0.18 μm CMOS
DRAM-logic FTI via pitch	50 μm
Package size	33 mm × 33 mm
BGA terminal	520 pin / 1mm pitch



1 Gbit stacked DRAM with TSV (512 Mbit × 2 strata)





A 3D Stacked Memory Integrated on a Logic Device Using SMAFTI Technology

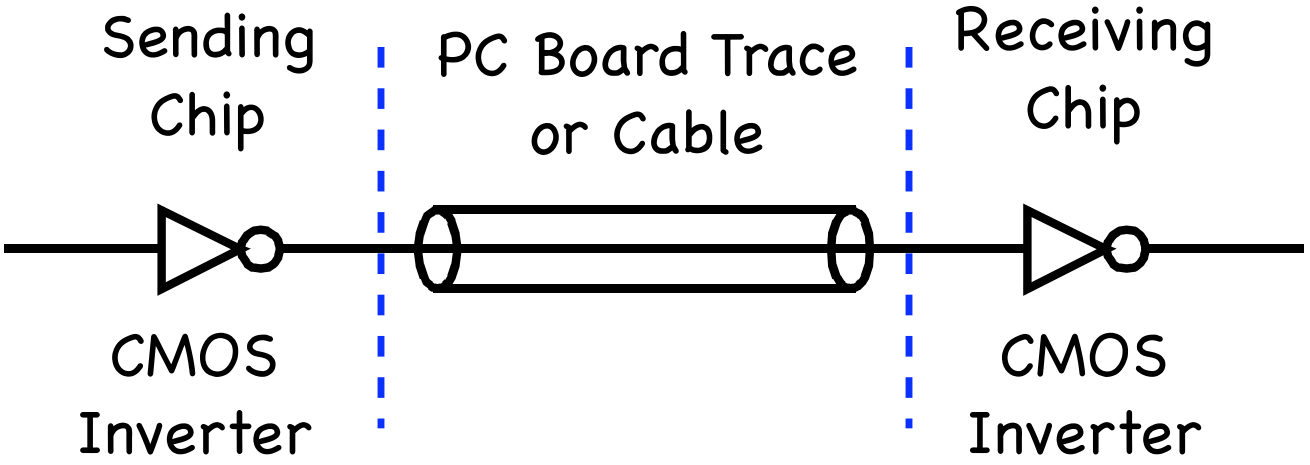
Yoichiro Kurita[1], Satoshi Matsui[1], Nobuaki Takahashi[1], Koji Soejima[1], Masahiro Komuro[1], Makoto Itou[1], Chika Kakegawa[1], Masaya Kawano[1], Yoshimi Egawa[2], Yoshihiro Saeki[2], Hidekazu Kikuchi[2], Osamu Kato[2], Azusa Yanagisawa[2], Toshiro Mitsuhashi[2], Masakazu Ishino[3], Kayoko Shibata[3], Shiro Uchiyama[3], Junji Yamada[3], and Hiroaki Ikeda[3]
[1]NEC Electronics, [2]Oki Electric Industry, and [3]Elpida Memory
1120 Shimokuzawa, Sagamihara, Kanagawa 229-1198, Japan
y.kurita@necel.com

Part I: High-Speed Serial I/O

High-Speed Serial I/O

- Why standard approaches are slow
- Incident-wave signaling
- Line equalization and eye diagrams
- Clock recovery
- Coding and framing

Standard CMOS I/O



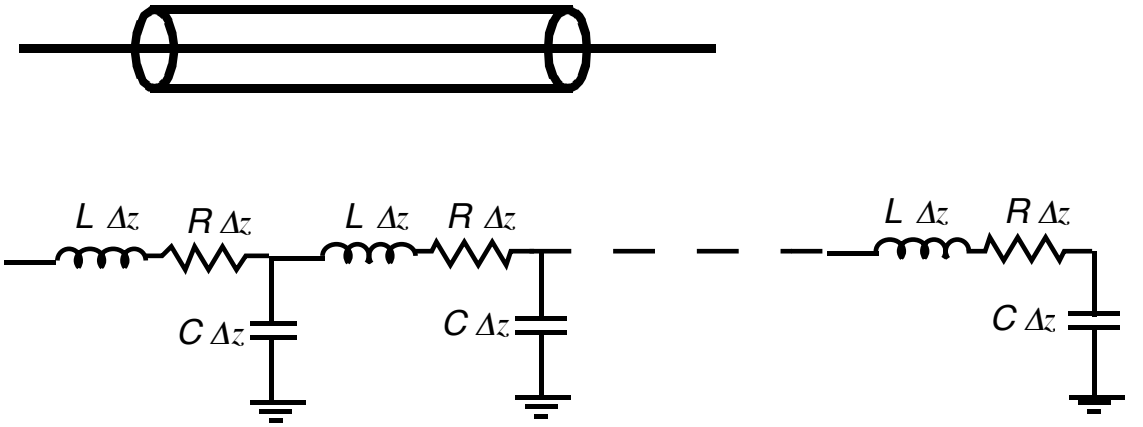
Slow (100 MHz rates, or less).
Power hungry (1nJ/bit, or more).
Bandwidth decreases with trace/cable length.

Simple models will help us understand why, and how to do better.

Cable Model

Trace/Cable can be modeled as a distributed RLC circuit.

Looking into a long cable, a circuit sees a characteristic impedance that is independent of the cable length.

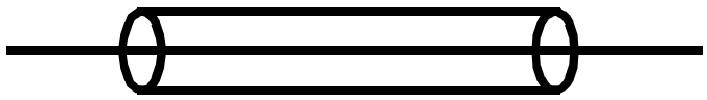


$Z_0 = 50\Omega$ → ← $Z_0 = 50\Omega$

A typical trace/cable has a characteristic impedance of about 50 ohms.

Cable Model

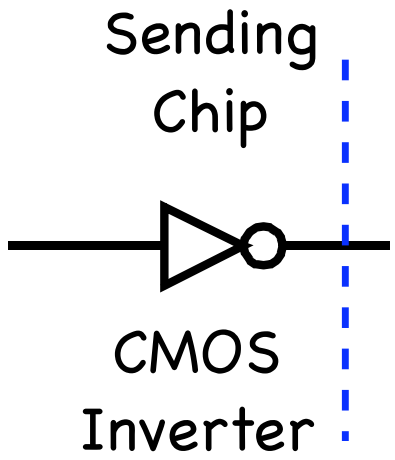
The highest frequencies of a pulse edge on a wire travel approach the "speed of light" of the wire medium (c_w).



And so, the fast rising edge of a pulse takes about L/c_w seconds to traverse a wire of length L.

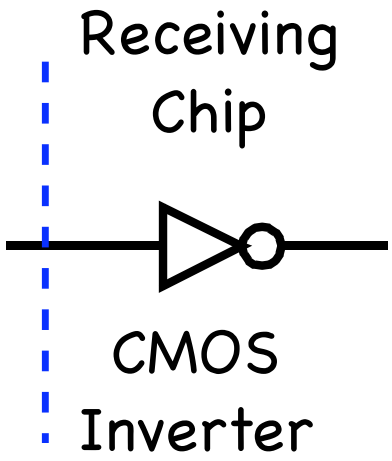
For our example, assume L/c_w = 4ns.

Inverter models.



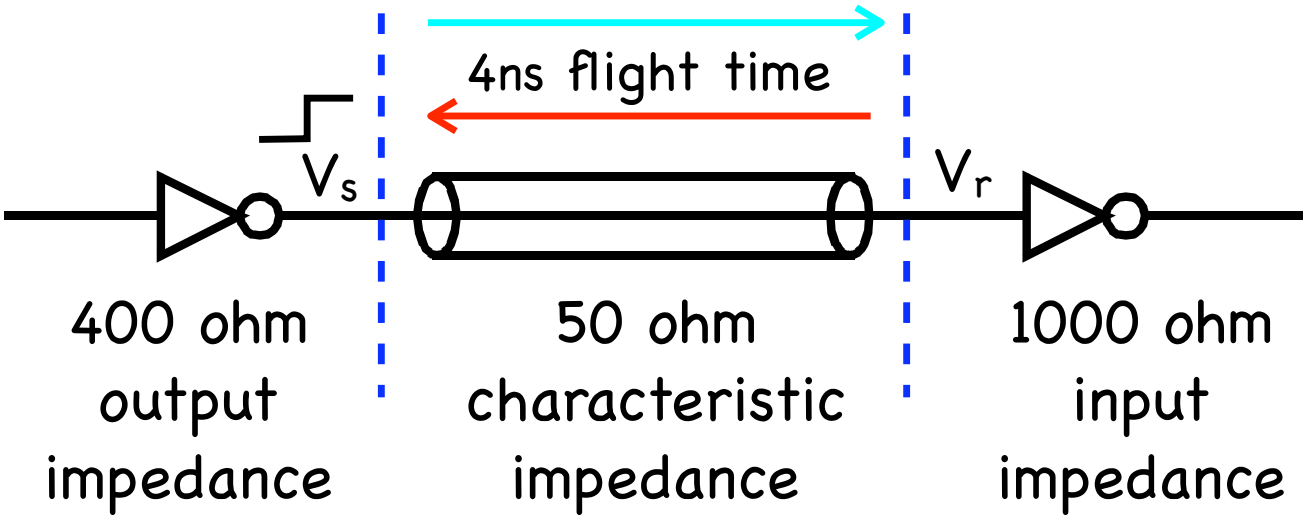
Typical output impedance is 400 ohms.

Typical input impedance of pad is 1000 ohms.

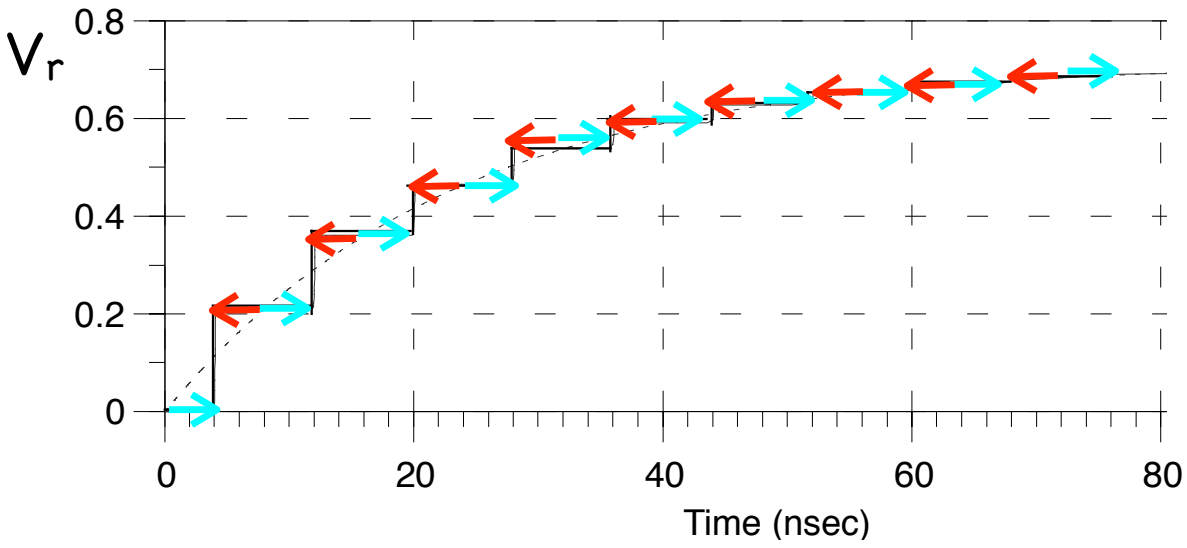


We now have the tools to model a rising edge sent from chip to chip.

Standard CMOS I/O



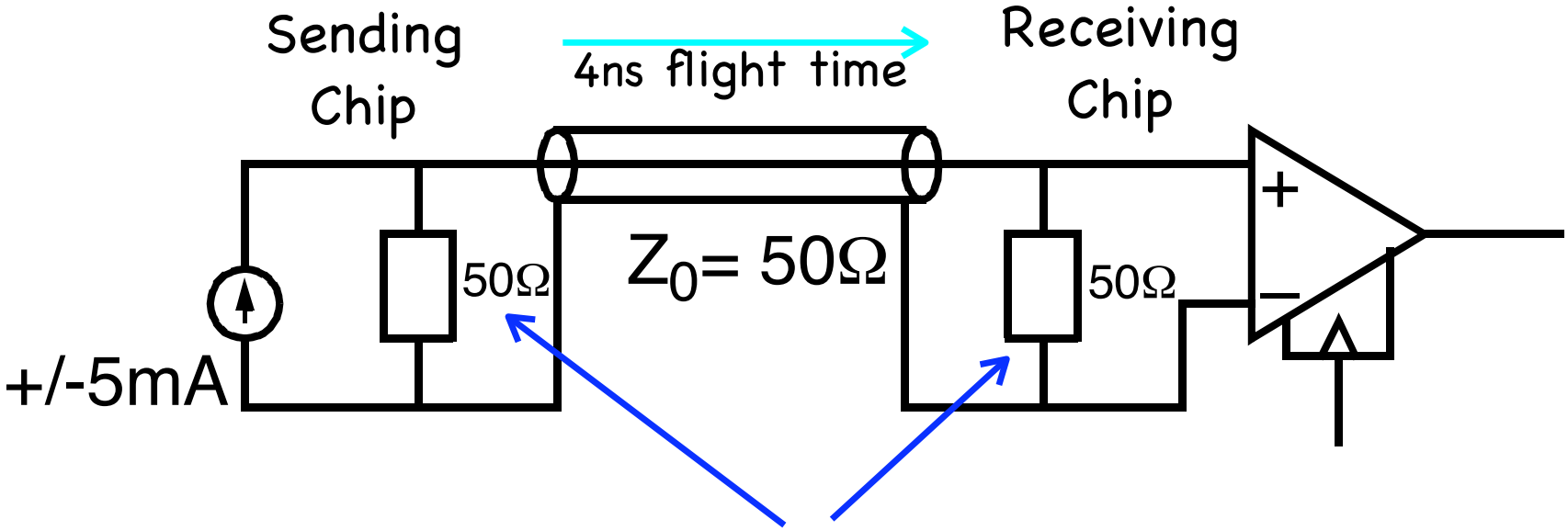
15 up-and-back traversals are needed to "ring-up" V_r to 90% of V_{dd} !



The impedance mismatches at each cable end reduce the pulse heights and launch the cyan reflection waves.

Incident-Wave Signaling

Incident-wave I/O

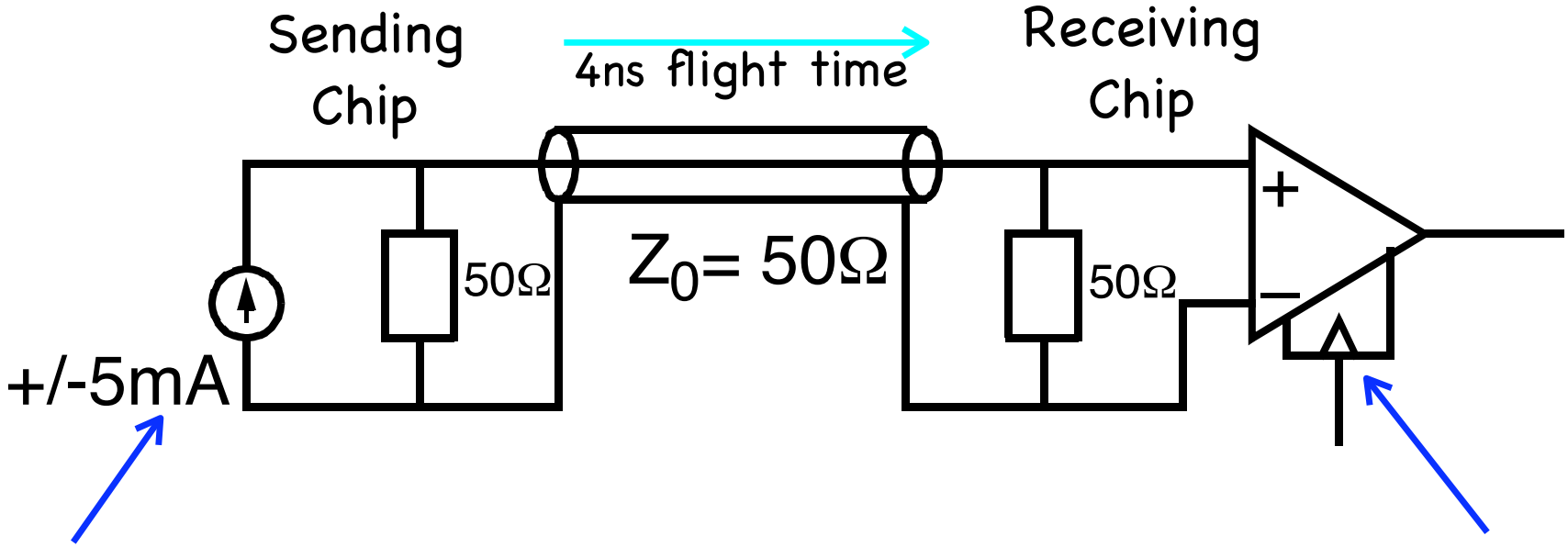


Kill reflections by making input and output impedances match the line impedance.

Each bit is communicated by the first arriving wave (the incident wave).

Several bits can be in-flight on the wire at once.

Differential, low-voltage.



Direction of current (+/-) codes one/zero.

Magnitude of current sets voltage (V=IR).

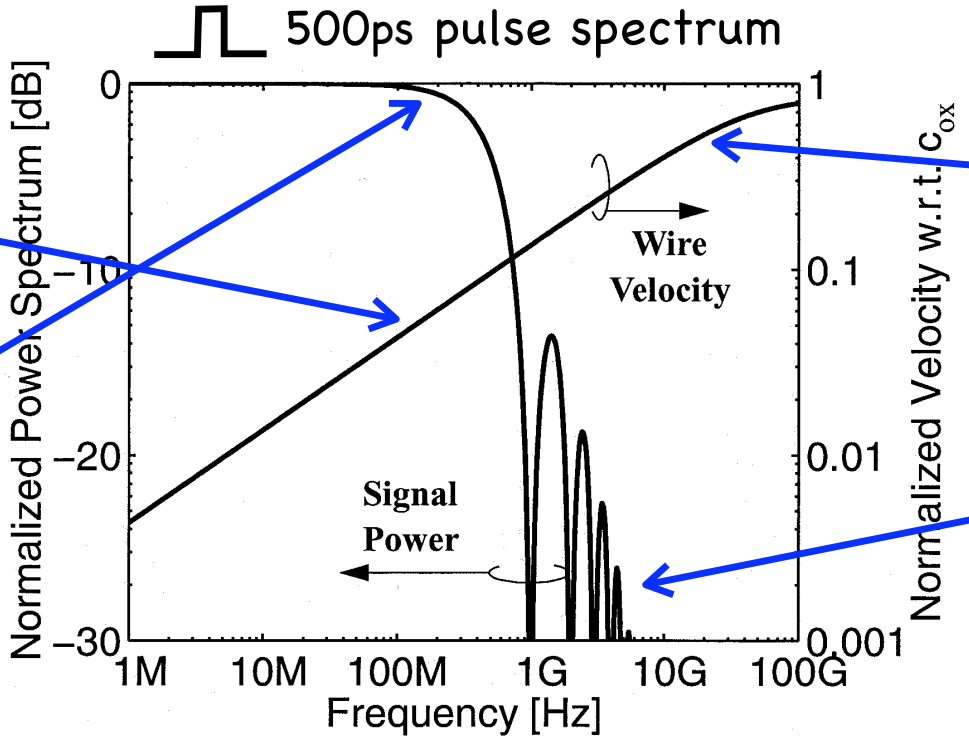
Like an SRAM sense amp. Differential, senses sign of voltage.

Energy dominated by DC power …

Line Equalization

Wire attenuation

Once we adopt the incident-wave approach, what limits our bandwidth?



Slow components of the pulse have high amplitude.

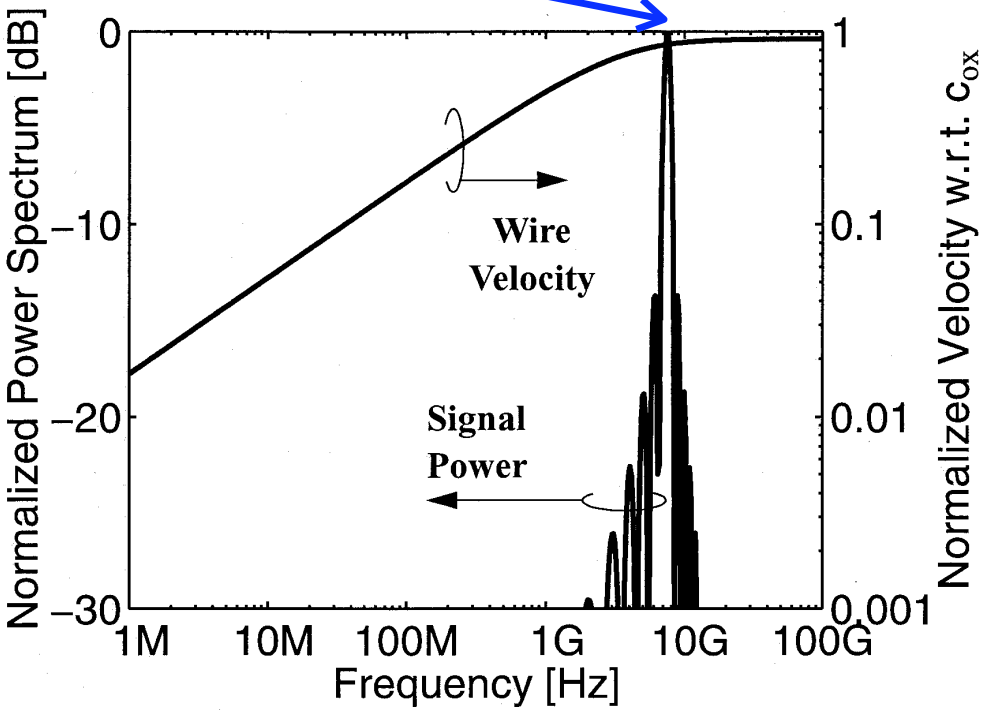
Fast components of the pulse have low amplitude.

The result is intersymbol interference. The slow-traveling low components of earlier bits swamp the fast edge of a new bit.

Equalization

How can we overcome intersymbol interference?

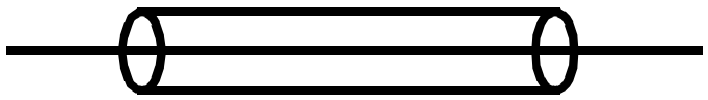
Ideally, we would send pulses whose frequency spectrum looks like this.



Or more generally, we want to send an ideal pulse that has been equalized to invert the wire frequency response.

Equalization

How can we overcome intersymbol interference?

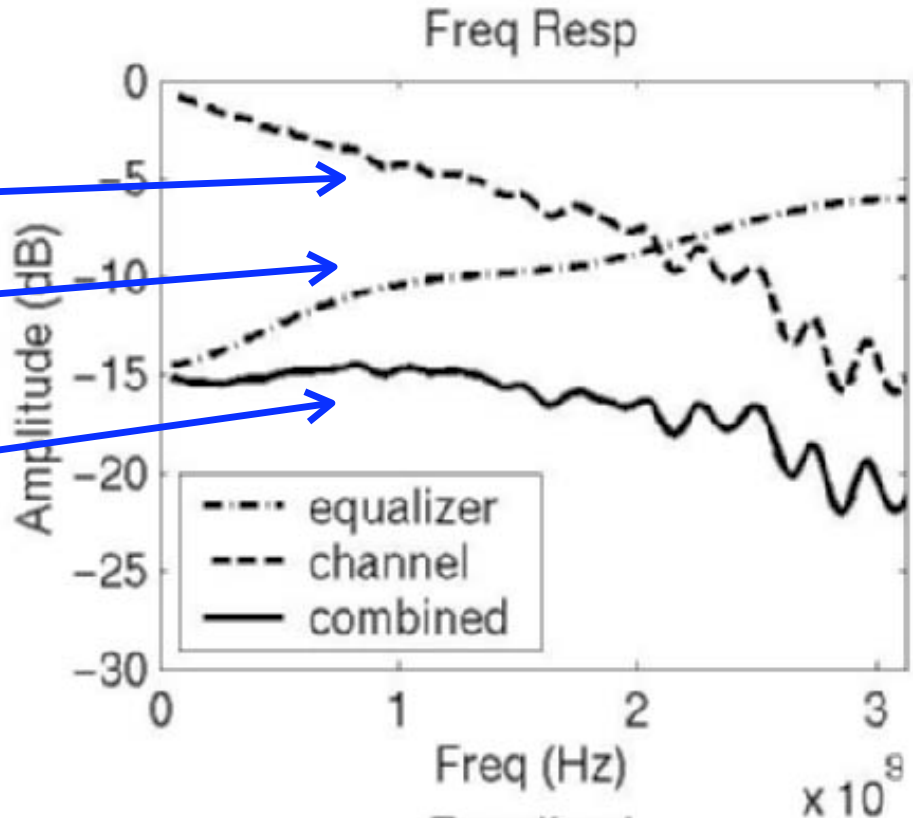


A simple 4-tap equalizer, sufficient for 4 Gb/s on a differential wire pair.

Original wire:

4-tap hi-pass EQ:

Combined, flatter response.



From: CMOS High-Speed I/Os — Present and Future

M.-J. Edward Lee[1], William J. Dally[1,2], Ramin Farjad-Rad[1], Hiok-Tiaq Ng[1], Ramesh Senthinathan[1], John Edmondson[1], and John Poulton[1]

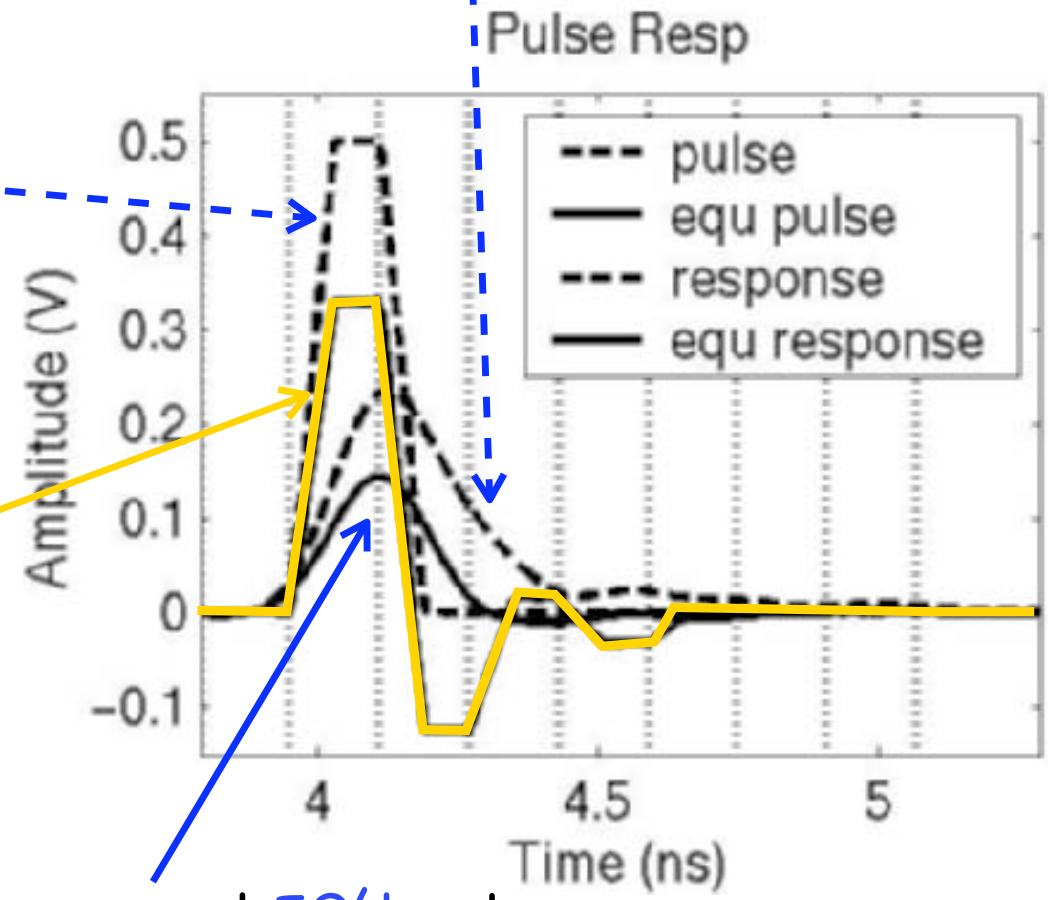
Equalization

The 4-tap equalizer, as seen in the time domain.

What receiver sees when we send non-EQ'd pulse.

Ideal pulse:

Ideal pulse after EQ:



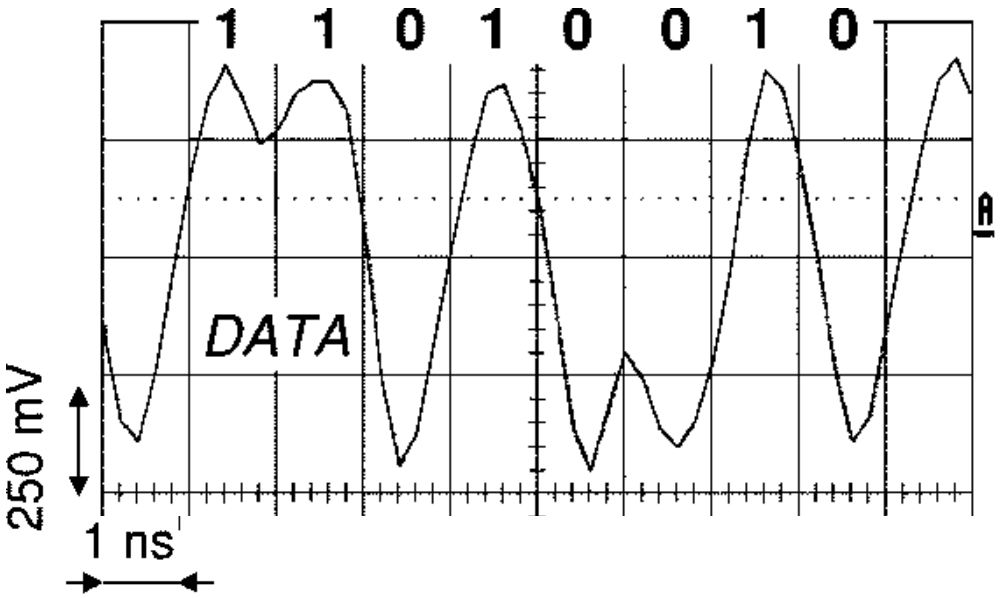
What receiver sees when we send EQ'd pulse.

Eye Diagrams

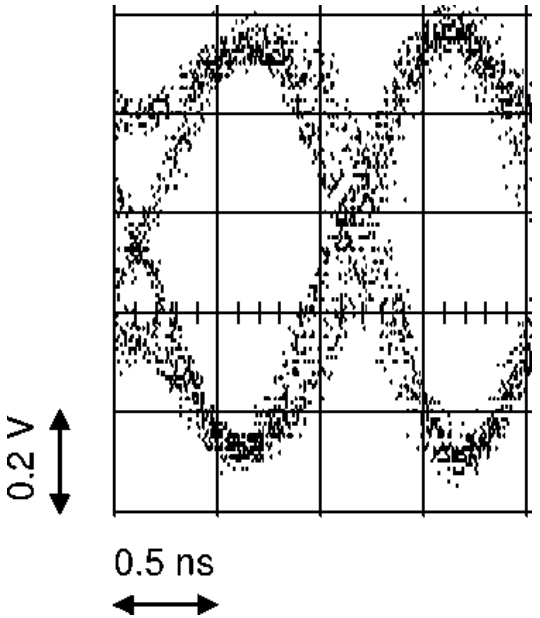
Eye diagrams

A way to visualize bits on a wire.

Oscilloscope trace of receive-end of wire.



Fold the trace at the clock period. If the received signal is clean, an open eye is seen.



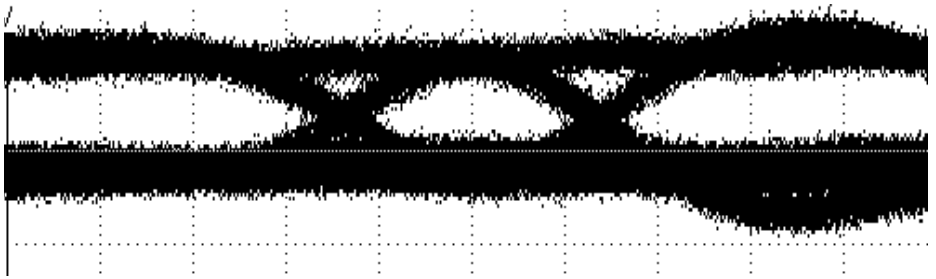
Eye diagrams

For 4 Gb/s, 4-tap equalizer example.

Receive waveform without EQ. Eye is closed, due to inter-symbol interference.

100ps/division

Receive waveform with EQ. Eye is open, due to boost of high-frequency pulse components.



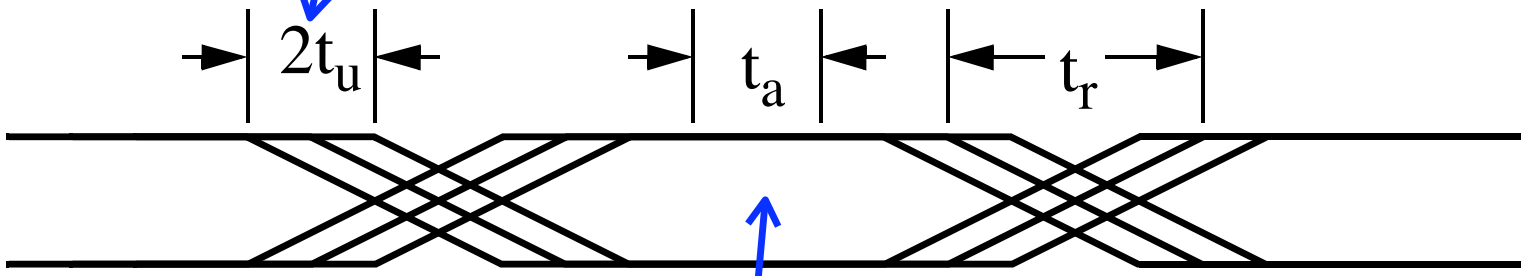
100ps/division

Eye diagrams

What limits bandwidth?

Uncertainty time.
All sources of temporal uncertainty (jitter, etc).

Rise time. Depends on drive current of output transistor.



Aperture time. How long it takes sense amp to make +/- decision.

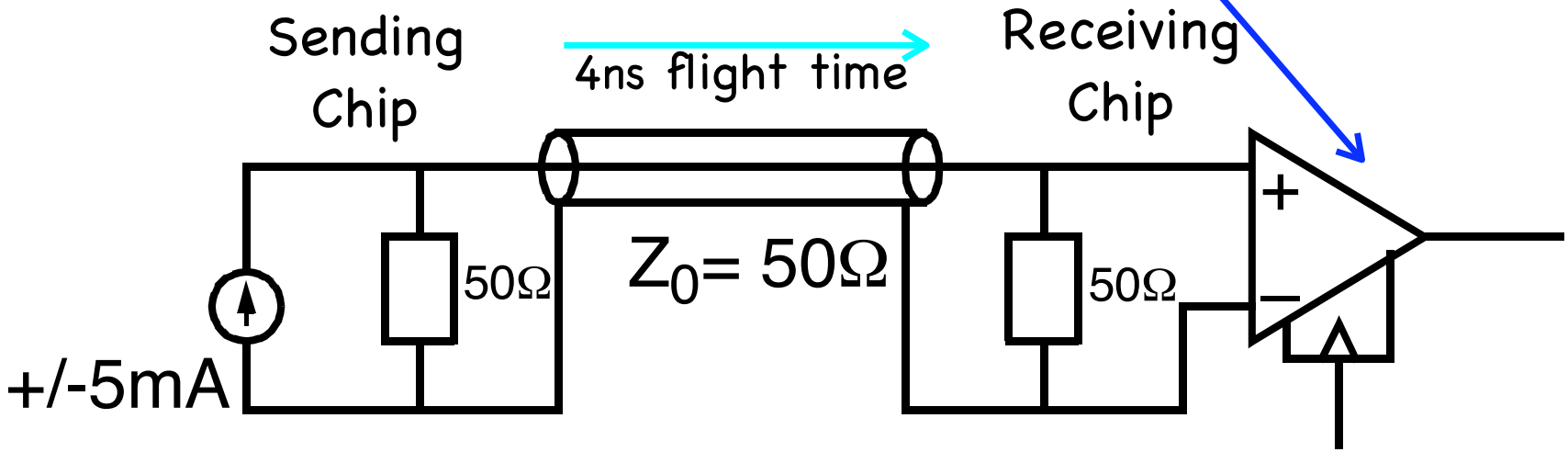
All should improve with process, but all have fundamental limits (thermal noise, non-perfect line equalization, etc).

Clock Recovery

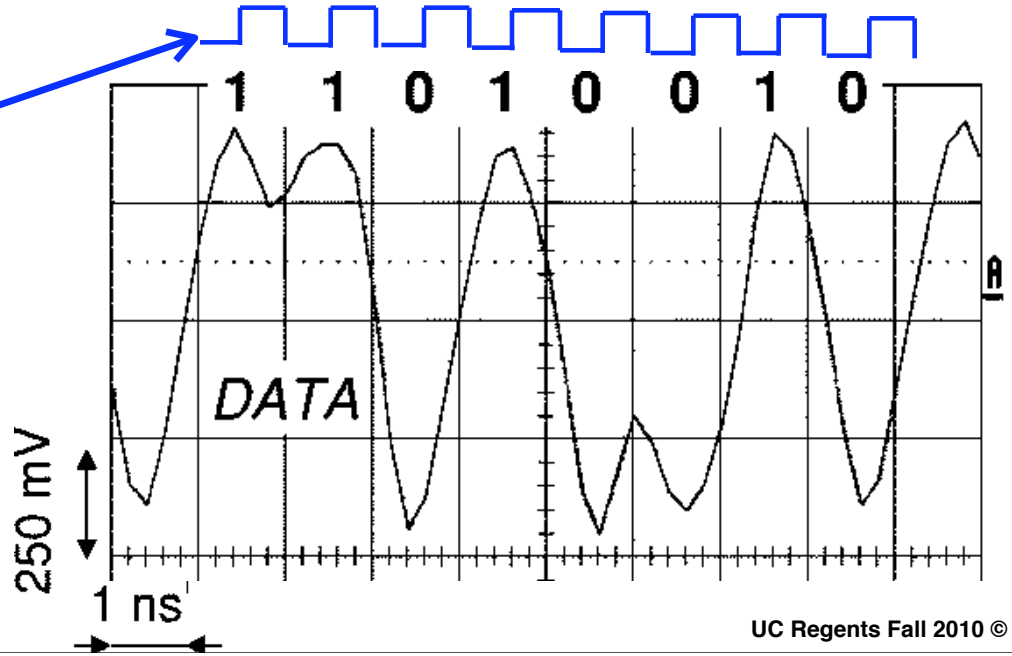
Incident-wave I/O

Sense amp is clocked.

How does the receiver place the clock edges?

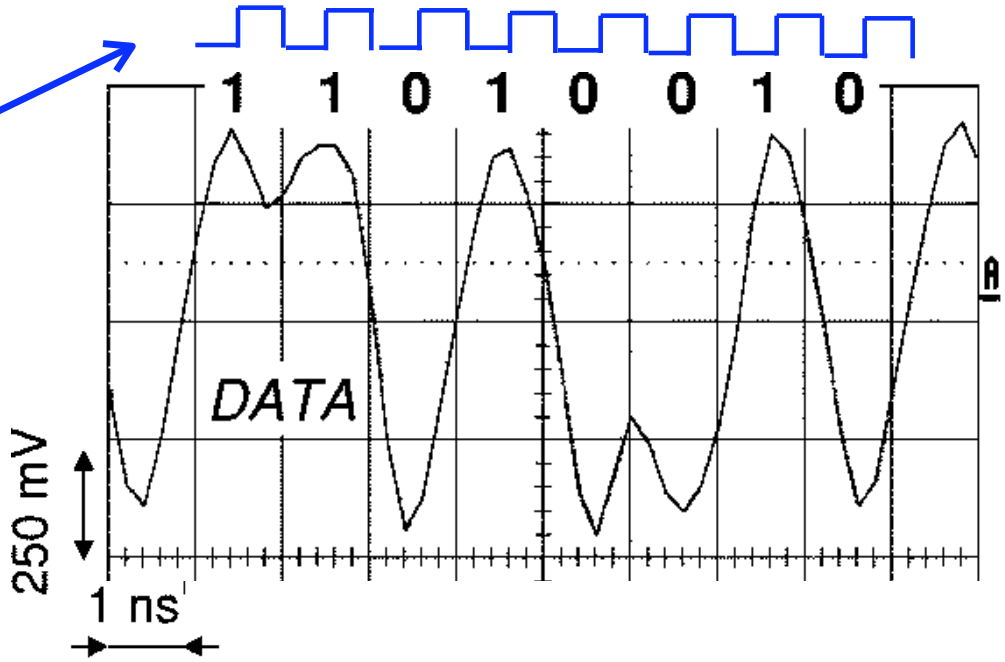


Sense amp should be clocked on positive edge of this clock. But receiver has to generate this clock from the data!

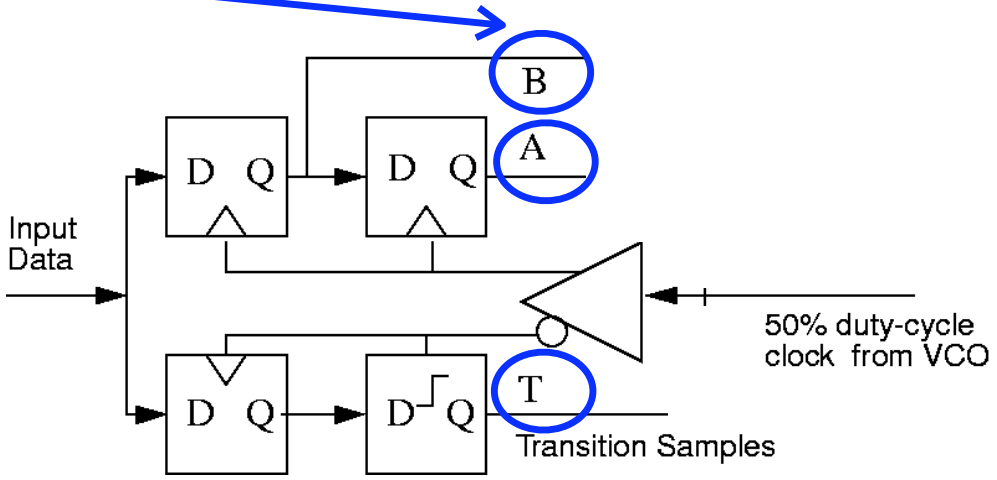


Alexander detector

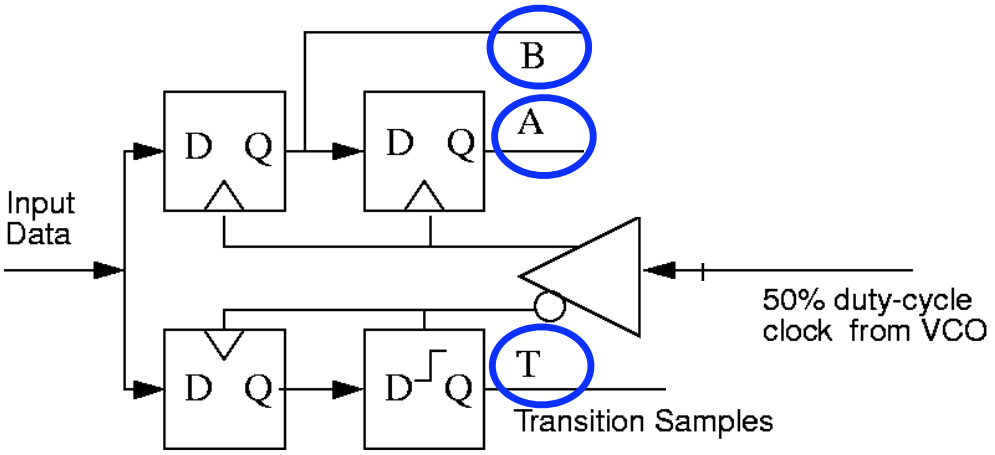
Make an initial guess for clock frequency, and clock in data on both edges.



If we guess clock frequency perfectly, A and B would be two adjacent bits, and T would be random …



Alexander detector

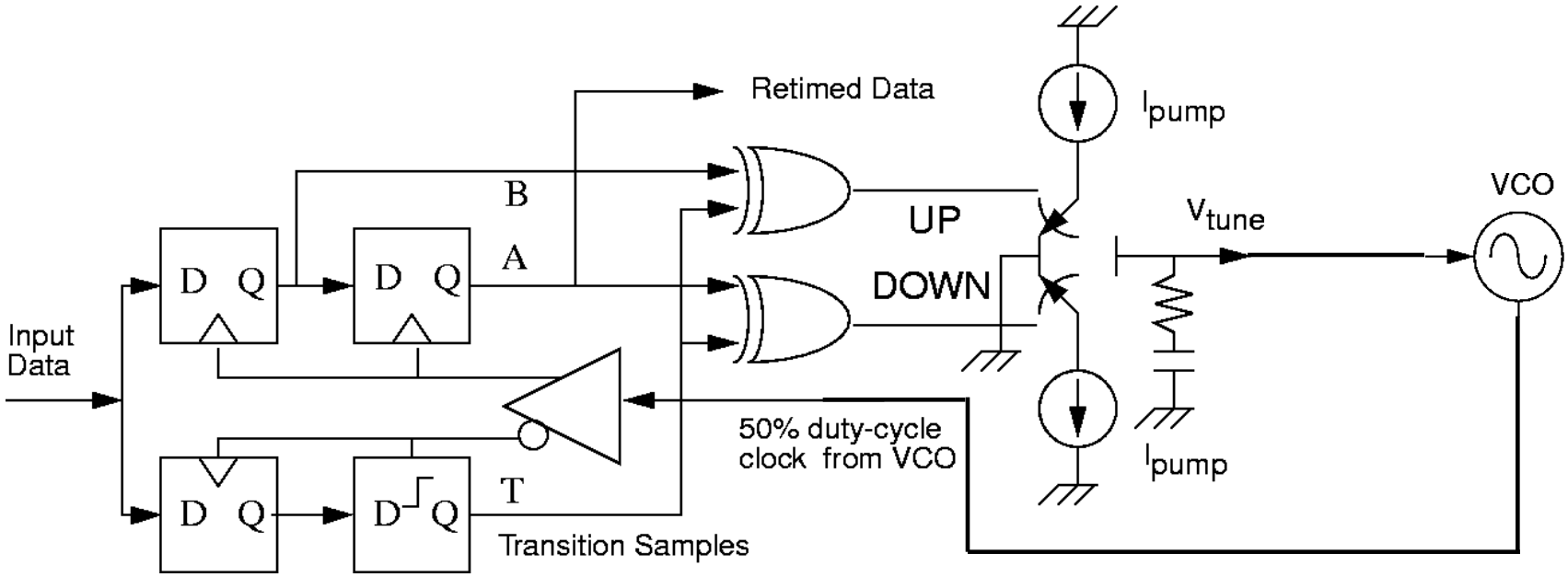


If our clock frequency guess is wrong, we can use this truth table on A, B, and T to see if the edges are arriving early or late.

State	A	T	B	UP	DOWN	Meaning
0	0	0	0	0	0	hold
1	0	0	1	0	1	early
2	0	1	0	1	1	error
3	0	1	1	1	0	late
4	1	0	0	1	0	late
5	1	0	1	1	1	error
6	1	1	0	0	1	early
7	1	1	1	0	0	hold

Alexander detector

We can embed this truth table as logic gates, and use it to tune a VCO's frequency to recover the transmitter's clock.

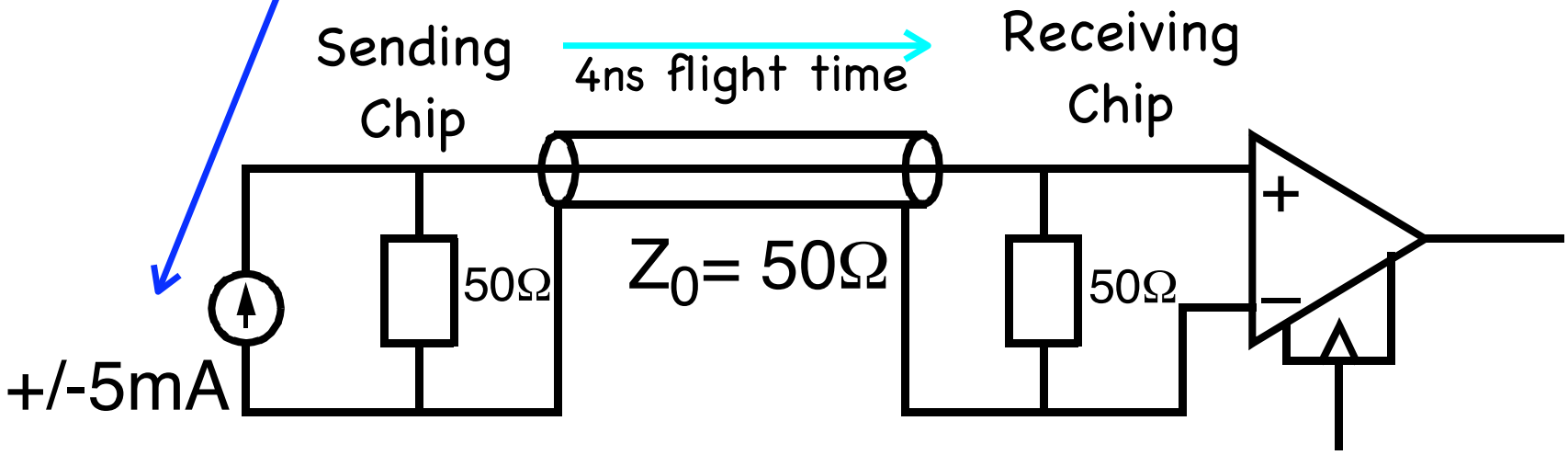


Real-world versions track duty-cycle changes, etc …

Coding and Framing

Input stream.

For clock recovery to work (and other reasons), we need to restrict the input data stream.



Input stream: a "river" of bits

0100101010010010101010101 ...

We restrict the characteristics of this river ...

A river of bits ...

0100101010010010101010101101010010 ...

M

- The first M bits will be receiver with very high error.
- Bits M+1 onward will be received correctly with high probability (but not 100% correct).
- Low error condition holds as long as bit river keeps flowing at a constant clock rate.

- At most, N consecutive 1's or 0's may appear in the river. N may be as low as 5.
- Over "long" stretches of bits, the number of 1's sent must equal the number of 0's sent.

8b/10b codes

Given an arbitrary bitstream, we can code it to have these desired properties.

Example: 8b/10b coding

Bits we want to send:

... 01001010100100101010101011010010 ...

Each 8 bit sequence recoded as 10 bits

... xxxxxxxxxx ...

Recoding algorithm guarantees 0/1 restrictions are met.

Code also offers "out-of-band" 10-bit codes that act as control characters, which higher-level protocols can use to frame the stream, etc.

Does not do error-correction on user data ...

Acknowledgment: Figures and data in this talk are excerpted from the papers below:

High-Performance Electrical Signaling

William J. Dally[1], Ming-Ju Edward Lee[1], Fu-Tai An[1], John Poulton[2], and Steve Tell[2]

CMOS High-Speed I/Os — Present and Future

M.-J. Edward Lee[1], William J. Dally[1,2], Ramin Farjad-Rad[1], Hiok-Tiaq Ng[1], Ramesh Senthinathan[1], John Edmondson[1], and John Poulton[1]

Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems

Richard C. Walker

Near Speed-of-Light Signaling Over On-Chip Electrical Interconnects

Richard T. Chang, *Student Member, IEEE*, Niranjan Talwalkar, *Student Member, IEEE*, C. Patrick Yue, *Member, IEEE*, and S. Simon Wong, *Fellow, IEEE*

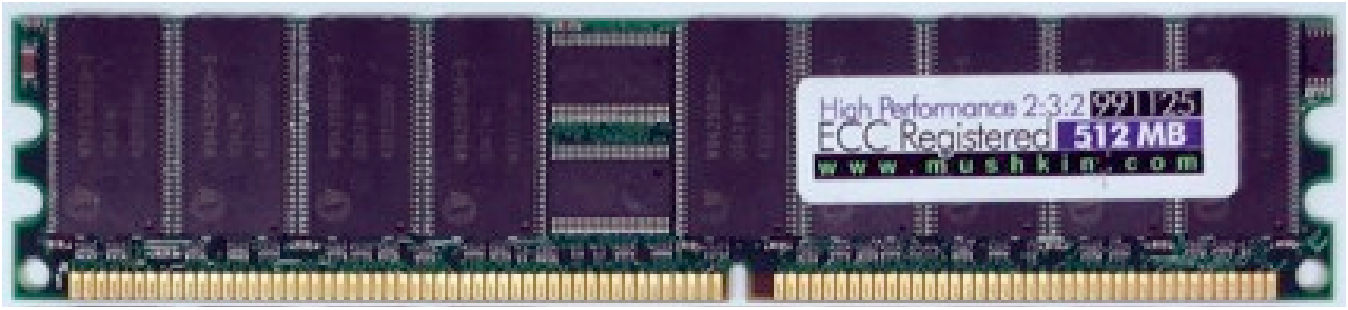
LVDS I/O Interface for Gb/s-per-Pin Operation in 0.35-μm CMOS

Andrea Boni, *Member, IEEE*, Andrea Pierazzi, and Davide Vecchi

Figures of Merit to Characterize the Importance of On-Chip Inductance

Yehea I. Ismail, Eby G. Friedman, and Jose L. Neves[1]

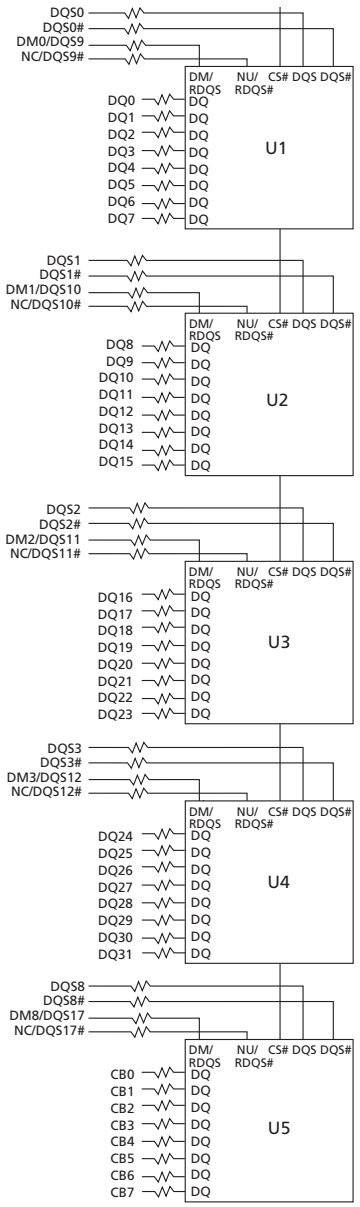
Part II: DRAM

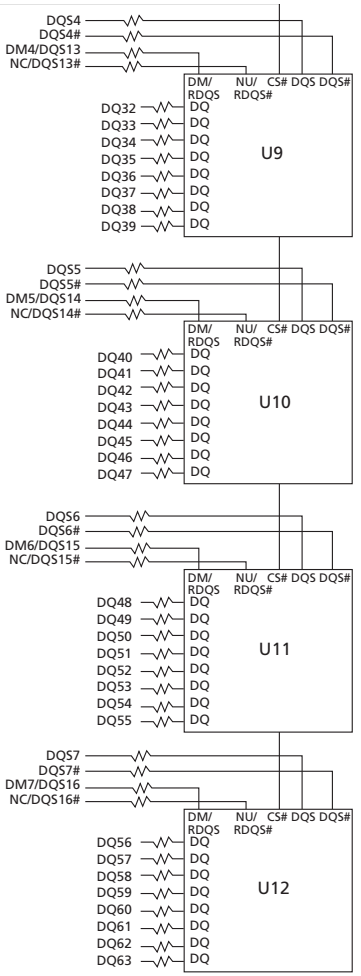


From DIMM module to DRAM chips ...

Each RAM chip responsible for 8 lines of the 64 bit data bus (U5 holds the check bits).

Commands sent to all 9 chips, qualified by per-chip select lines.





Lower levels of DRAM bus specification

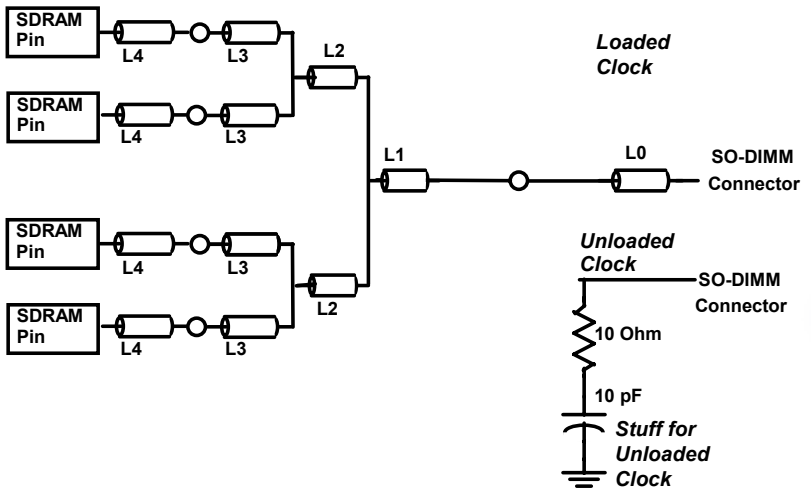


Table 9: Trace Length Table for Clock Topologies

Segment	L0	L1	L2	L3	L4	Total Min	Total Max
Length	0.10	1.00	0.80	0.50	0.10	2.45	2.55
Tolerance	± 0.05	± 0.02	± 0.02	± 0.02	± 0.05		
Layer	Outer	Inner	Inner	Inner	Outer		

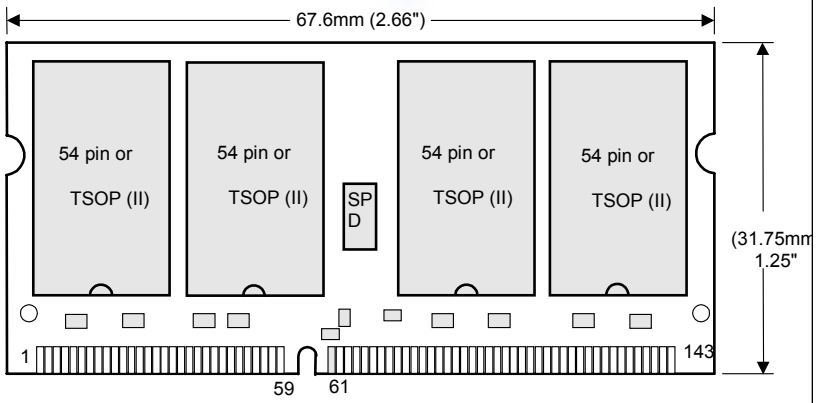
Transaction Protocols

Signal Timing on Wires

Wires

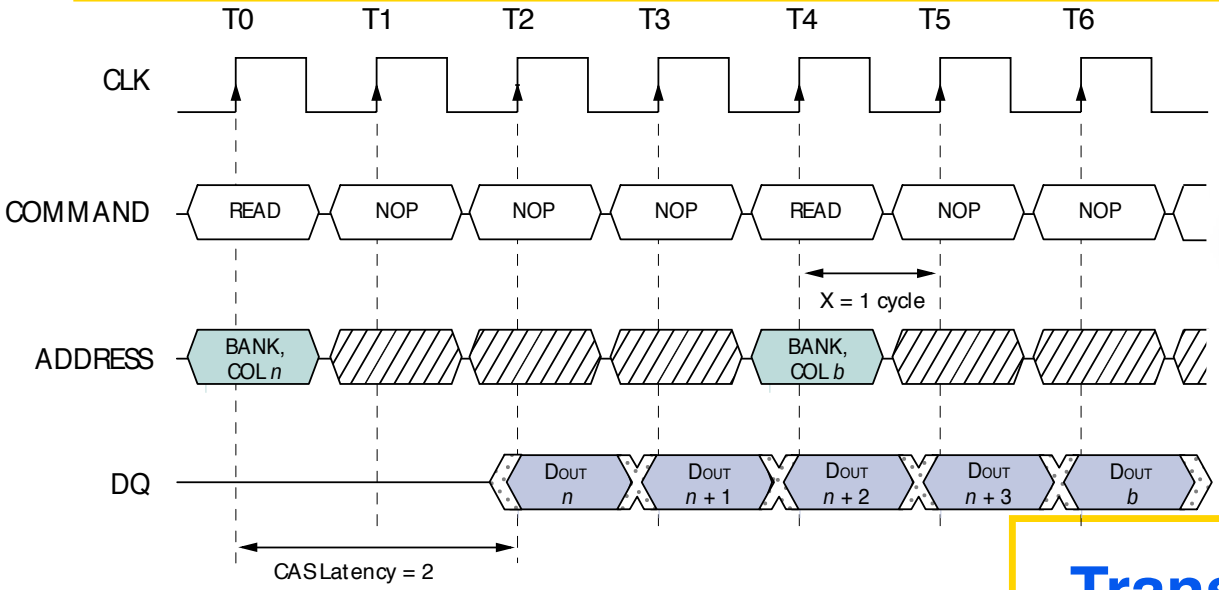
Electrical Properties

Mechanical Properties



Ideally, DIMMs made by any manufacturer should fit into any compliant socket, and work.

Upper levels of DRAM bus specification



Collaboration between DRAM manufacturers (Samsung, Micron) and DRAM users (Intel, Cisco, ...).

TIMING PARAMETERS

SYMBOL*	-7E MIN	-7E MAX	-75 MIN	-75 MAX	UNITS
tAC (3)		5.4		5.4	ns
tAC (2)		5.4		6	ns
tAH	0.8		0.8		ns
tAS	1.5		1.5		ns
tCH	2.5		2.5		ns
tCL	2.5		2.5		ns
tCK (3)	7		7.5		ns
tCK (2)	7.5		10		ns
tCKH	0.8		0.8		ns

Transaction Protocols

Signal Timing on Wires

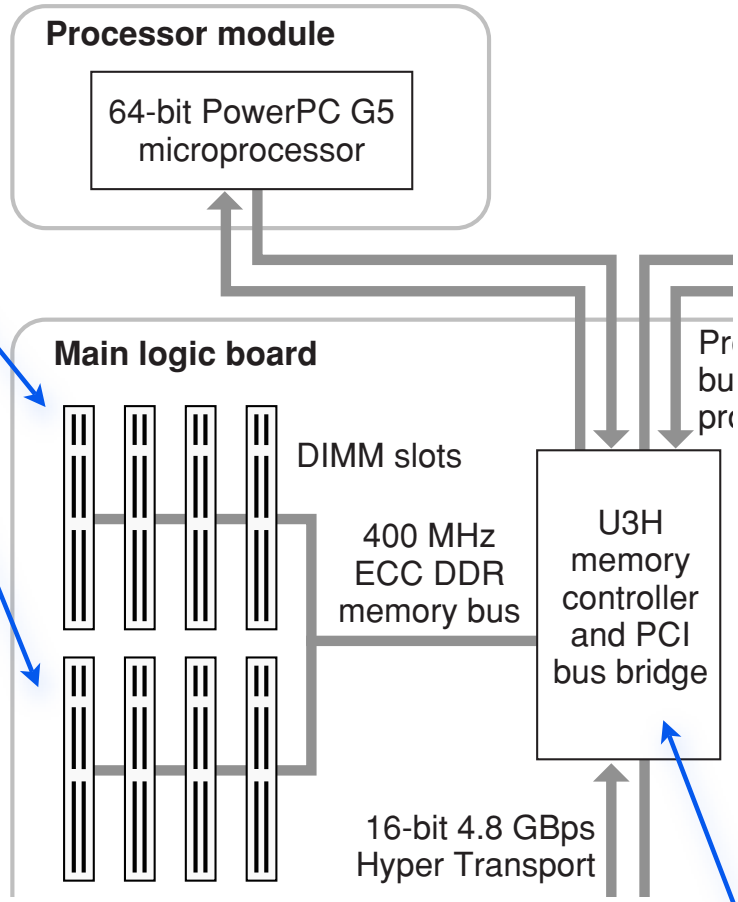
Wires

Electrical Properties

Mechanical Properties

Bus wires shared between many DIMMS

Apple Xserve G5 - has 8 DIMM slots, to support 8GB.



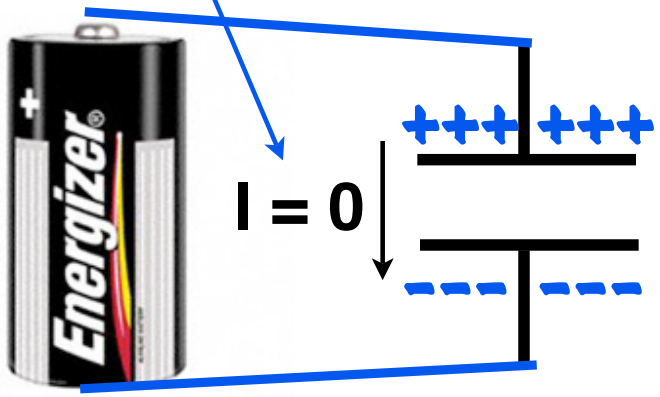
DIMMs respond to transaction requests. Since memory controller is the only bus master, and there are a small number of DIMM slots, bus sharing is easy: use dedicated wires to each slot.

Memory controller is the only "bus master" - it can start transactions on the bus, but the DIMMs cannot.

Dynamic Memory Cells

Recall: Capacitors in action

Because the dielectric is an insulator, and does not conduct.



After circuit "settles" ...

Q = C V = C * 1.5 Volts (D cell)

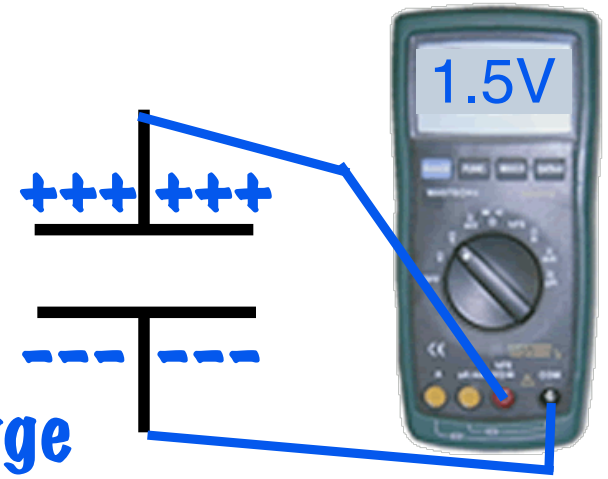
Q: Charge stored on capacitor

C: The capacitance of the device: function of device shape and type of dielectric.

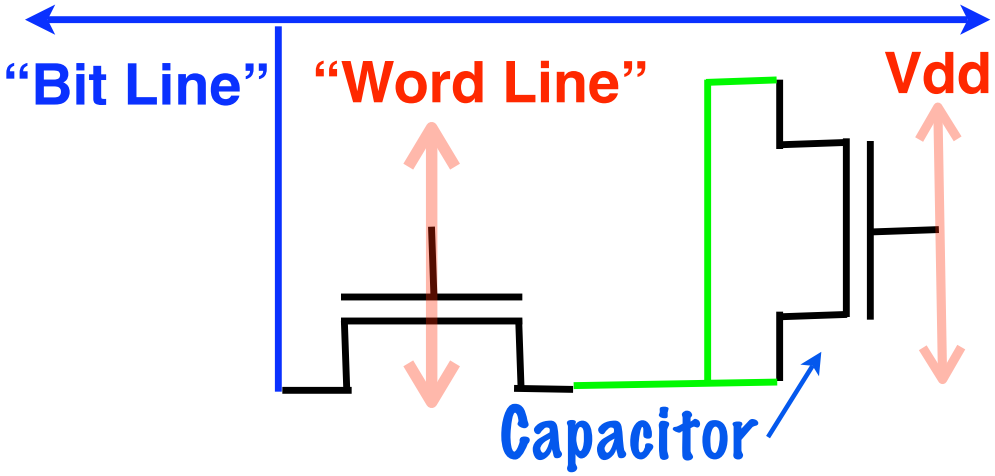
After battery is removed:

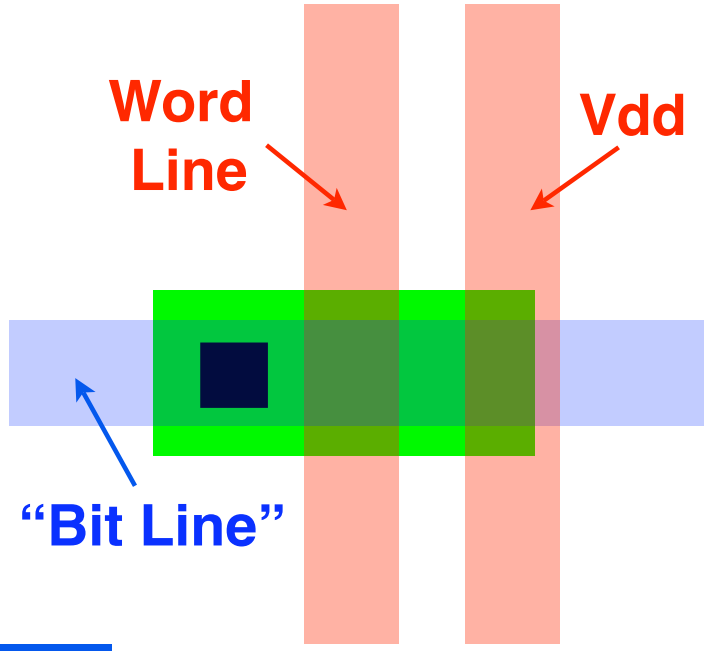
Still, Q = C * 1.5 Volts

Capacitor "remembers" charge

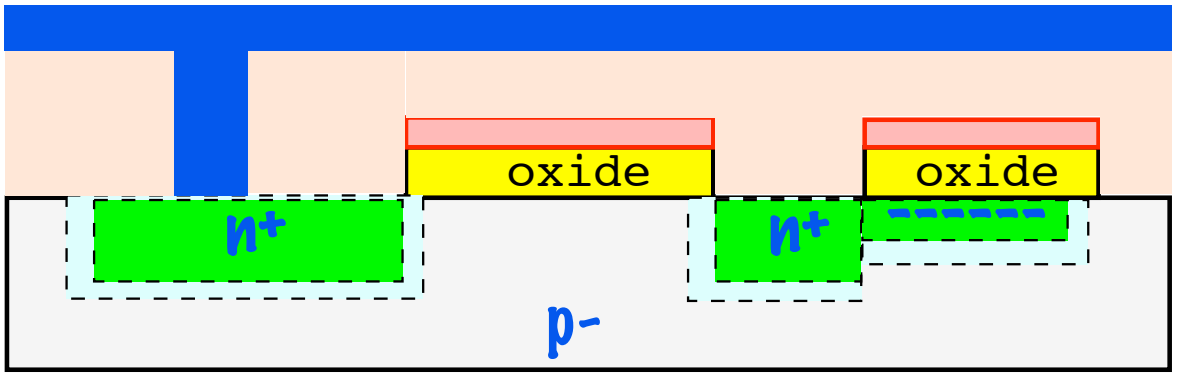


DRAM cell: 1 transistor, 1 capacitor



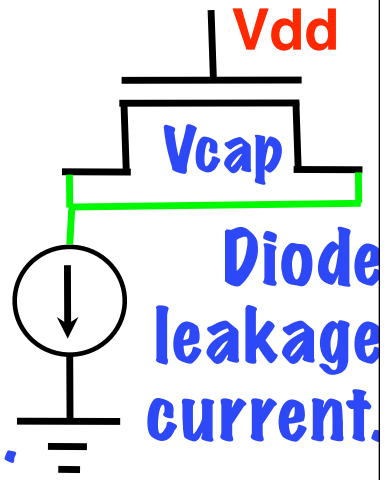


"Bit Line"

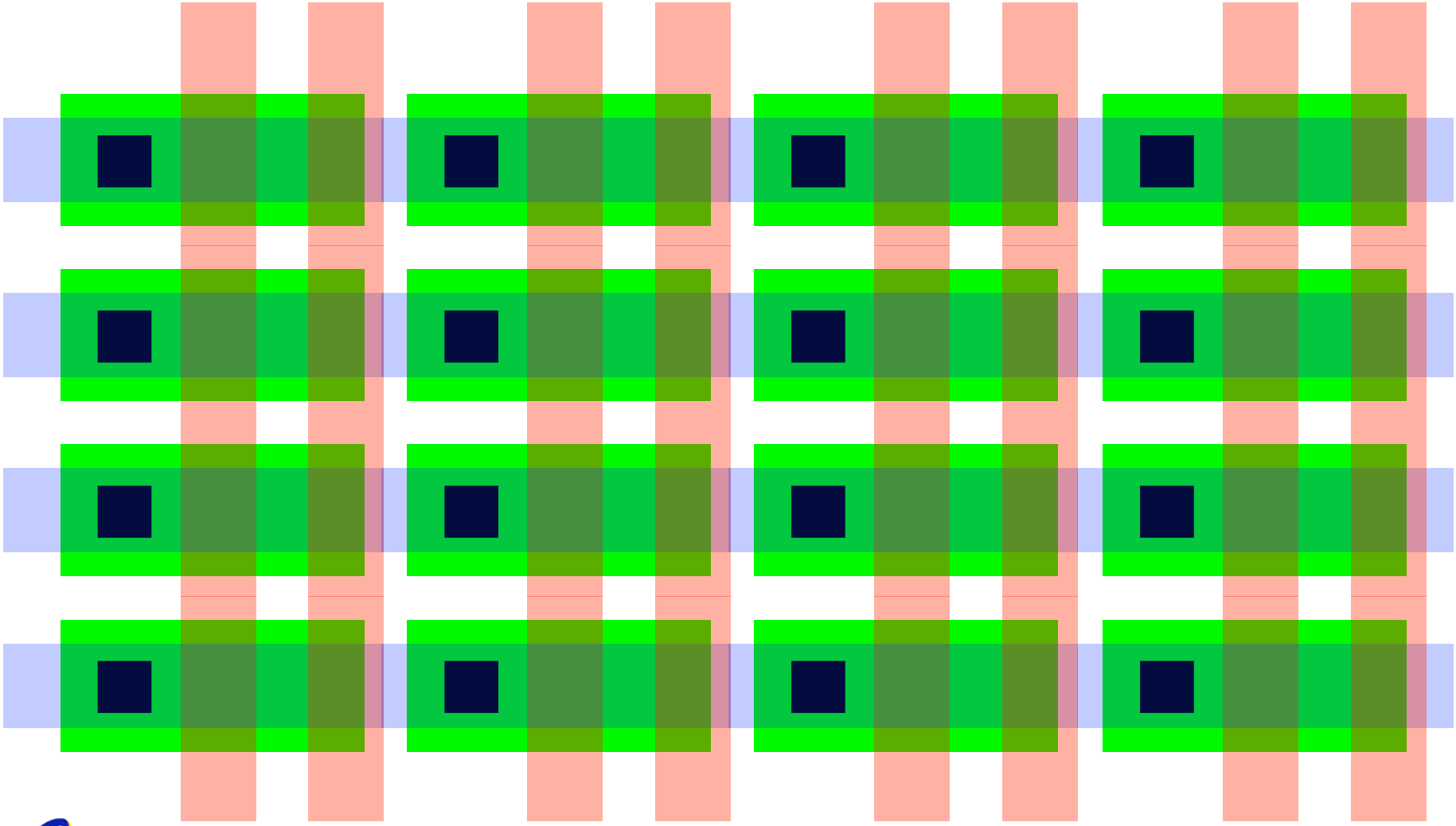


Word Line and Vdd run on "z-axis"

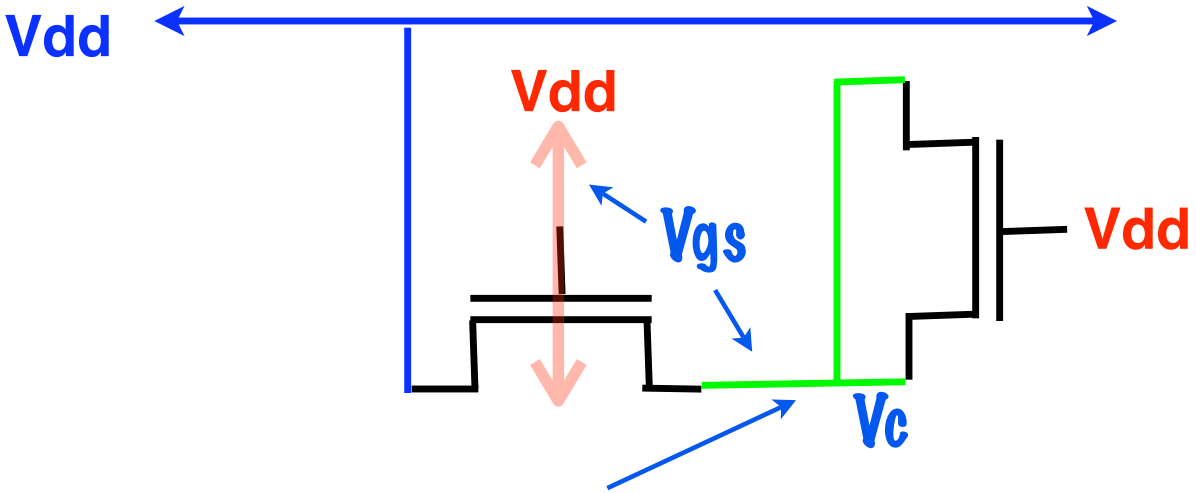
Why Vcap values start out at ground.



A 4 x 4 DRAM array (16 bits) ....



DRAM Circuit Challenge #1: Writing

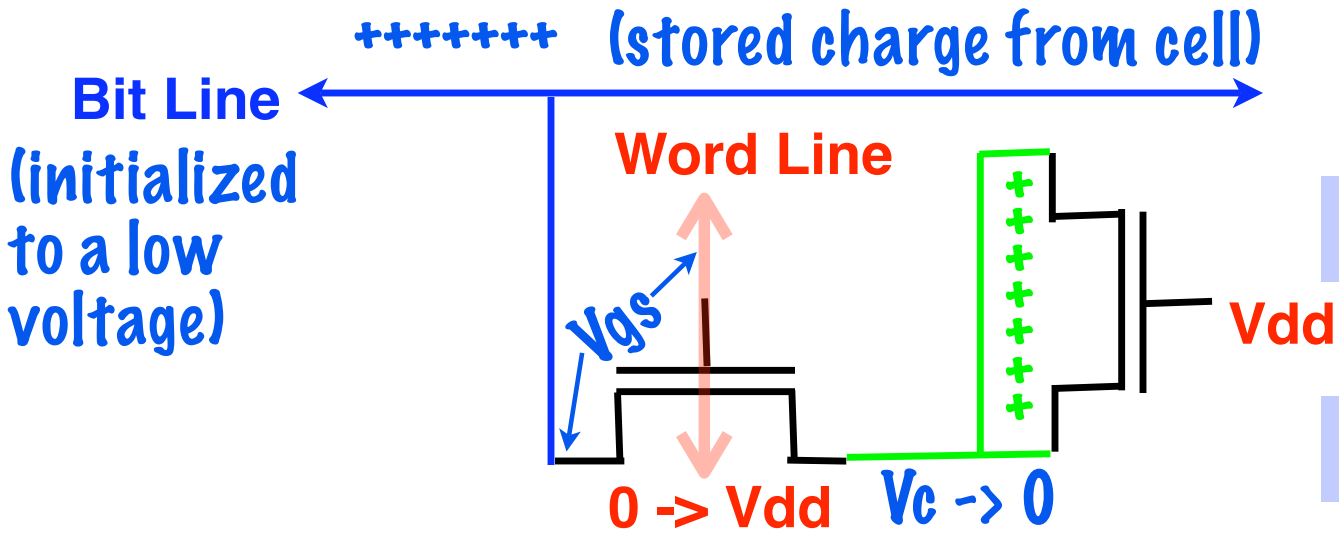


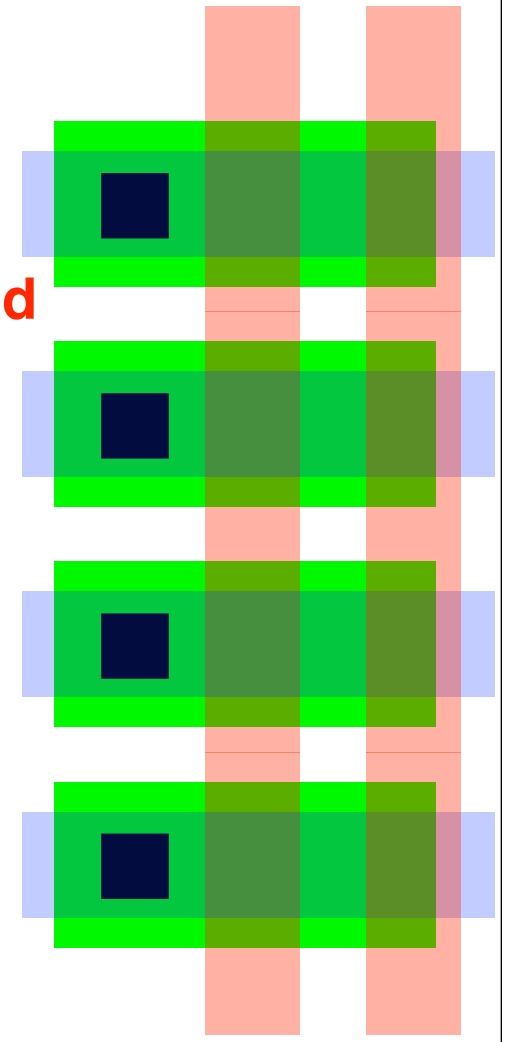
Vdd - Vth. Bad, we store less charge. Why do we not get Vdd?

Ids = [(μϵW)/(2LD)] [Vgs -Vth]^2 ,
but "turns off" when Vgs <= Vth!

Vgs = Vdd - Vc. When Vdd - Vc == Vth, charging effectively stops!

DRAM Challenge #2: Destructive Reads

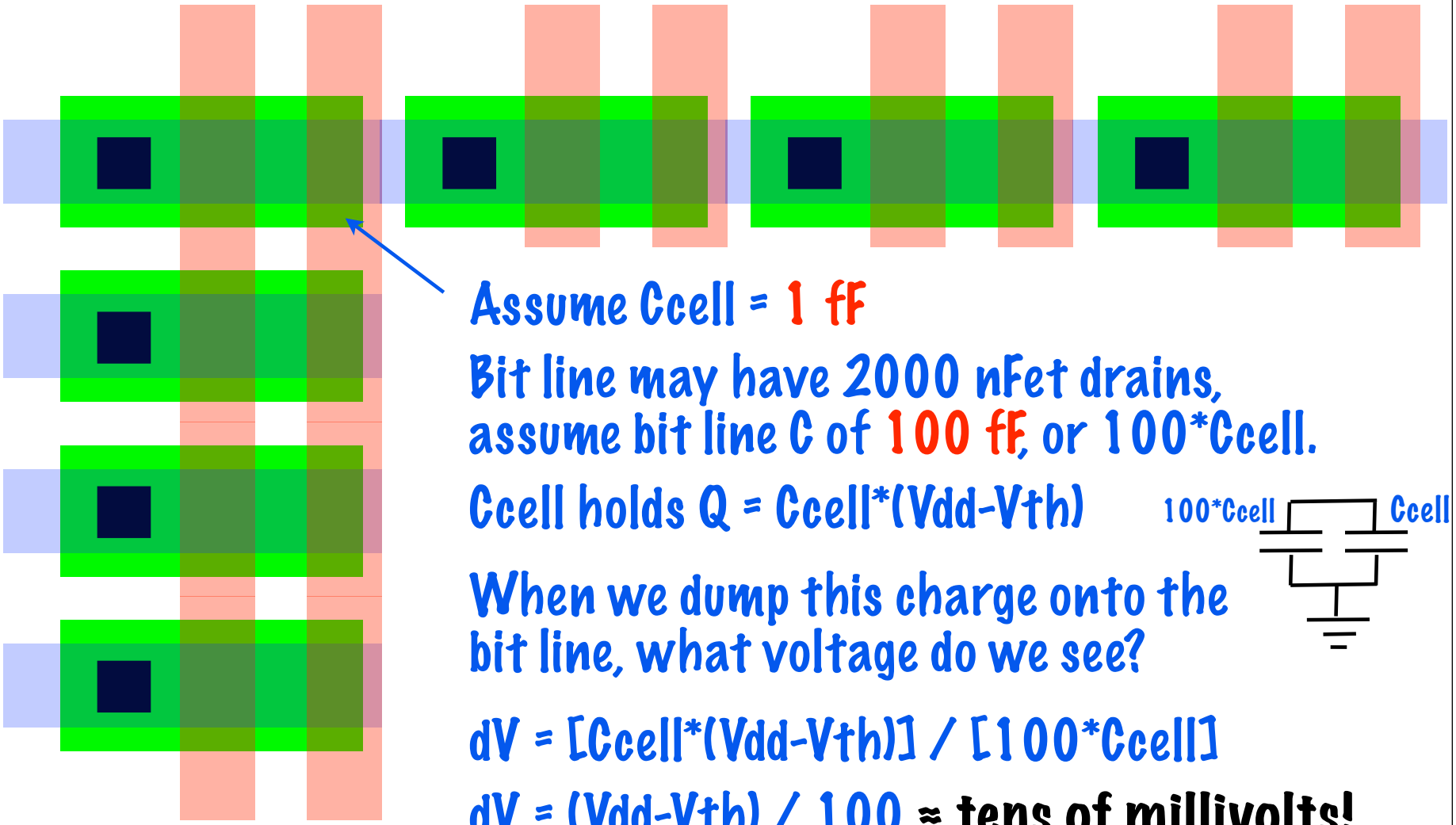




Raising the word line removes the charge from every cell it connects to!

Must write back after each read.

DRAM Circuit Challenge #3a: Sensing



Assume Ccell = 1 fF

Bit line may have 2000 nFet drains, assume bit line C of 100 fF, or 100*Ccell.

Ccell holds Q = Ccell*(Vdd-Vth)

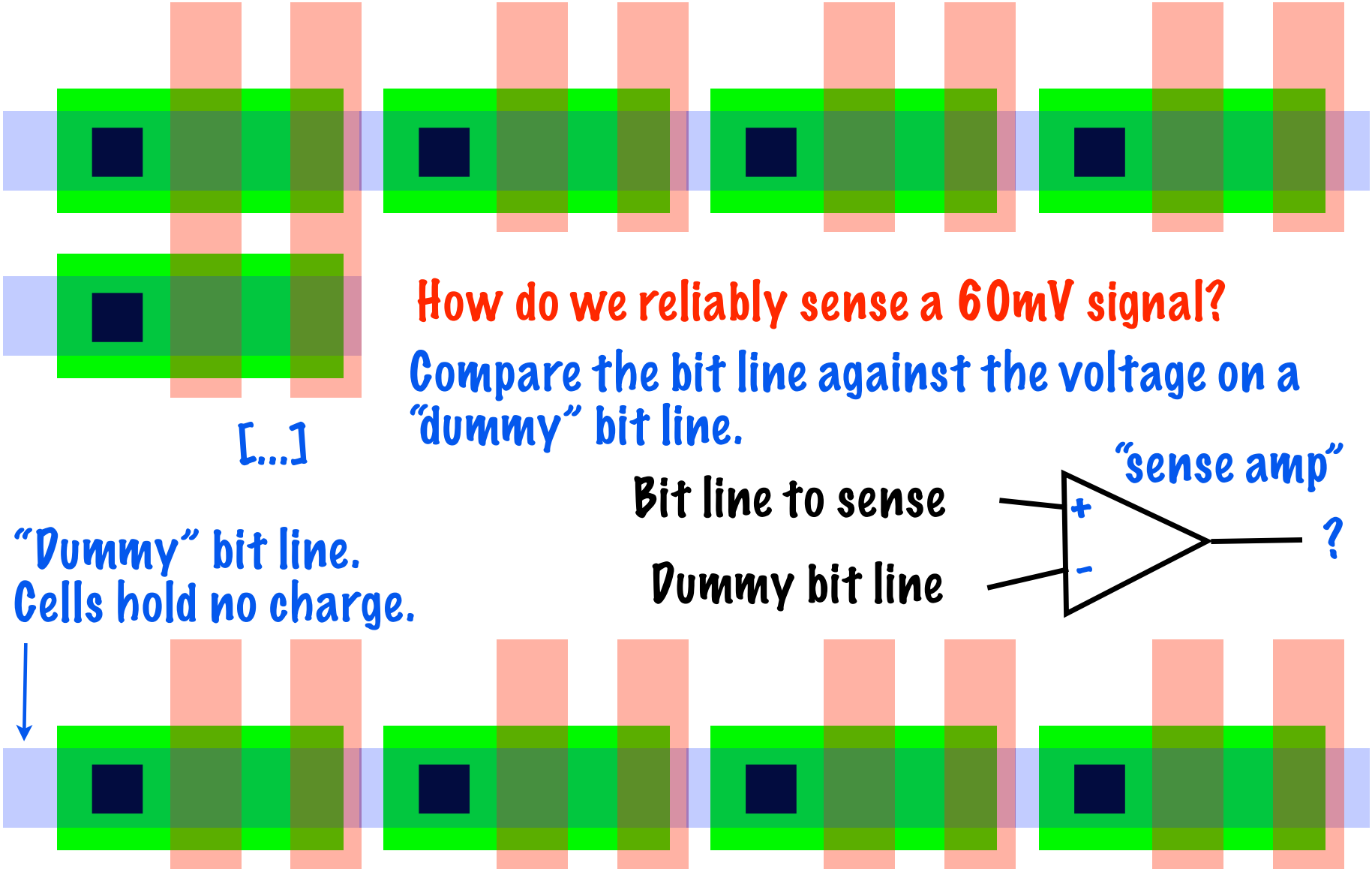
When we dump this charge onto the bit line, what voltage do we see?

dV = [Ccell*(Vdd-Vth)] / [100*Ccell]

dV = (Vdd-Vth) / 100 ≈ tens of millivolts!

In practice, scale array to get a 60mV signal.

DRAM Circuit Challenge #3b: Sensing



How do we reliably sense a 60mV signal?

Compare the bit line against the voltage on a "dummy" bit line.

[...]

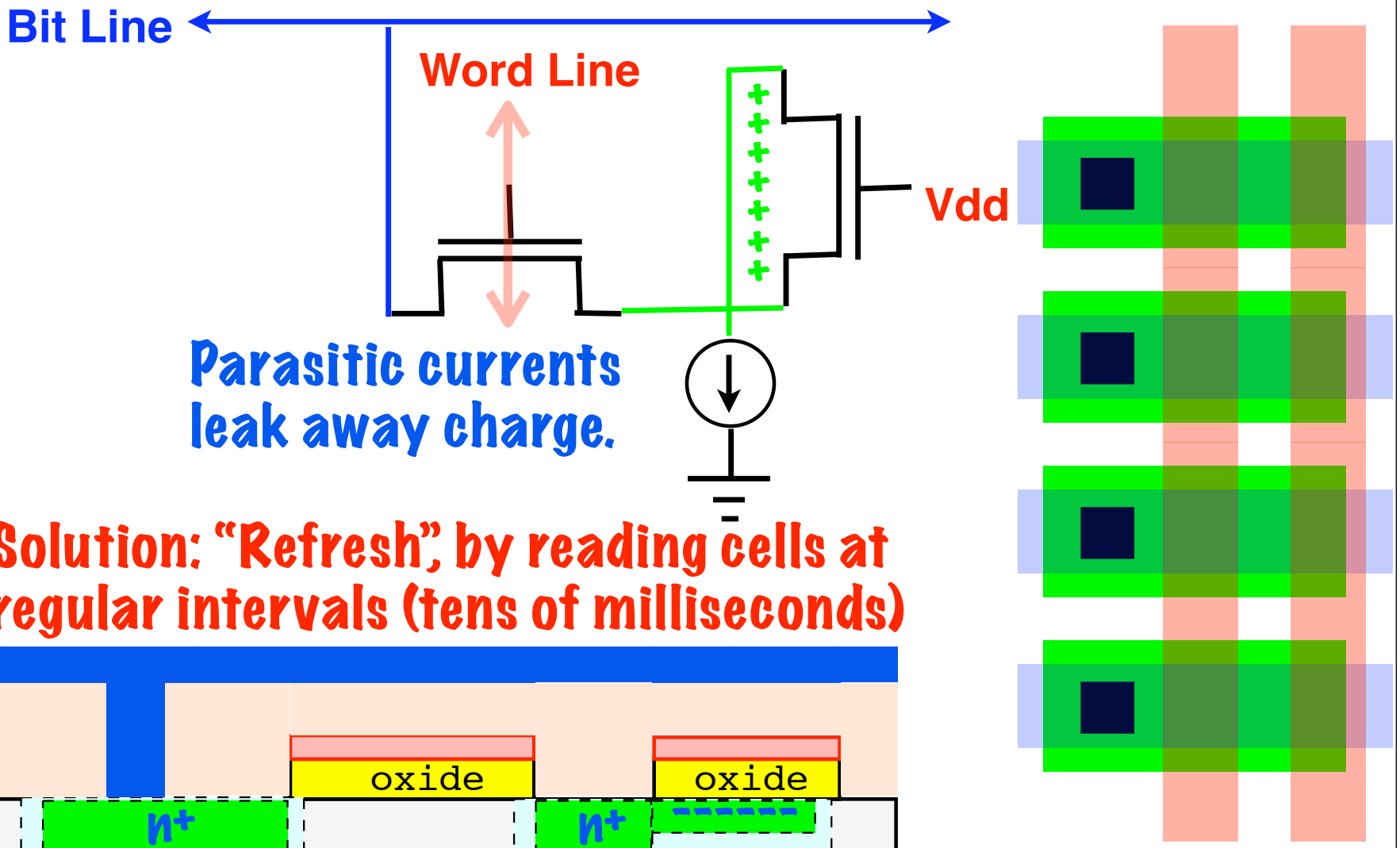
"Dummy" bit line.
Cells hold no charge.

Bit line to sense

Dummy bit line

"sense amp"

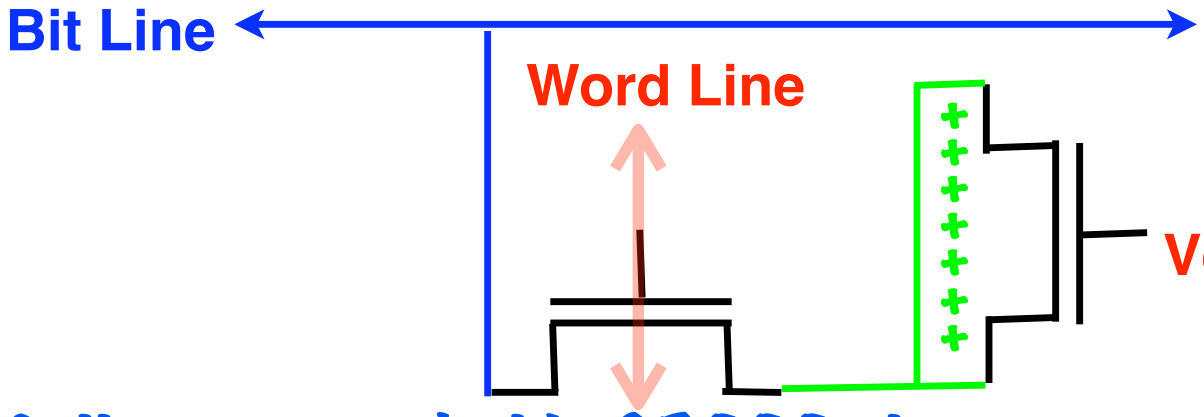
DRAM Challenge #4: Leakage ...

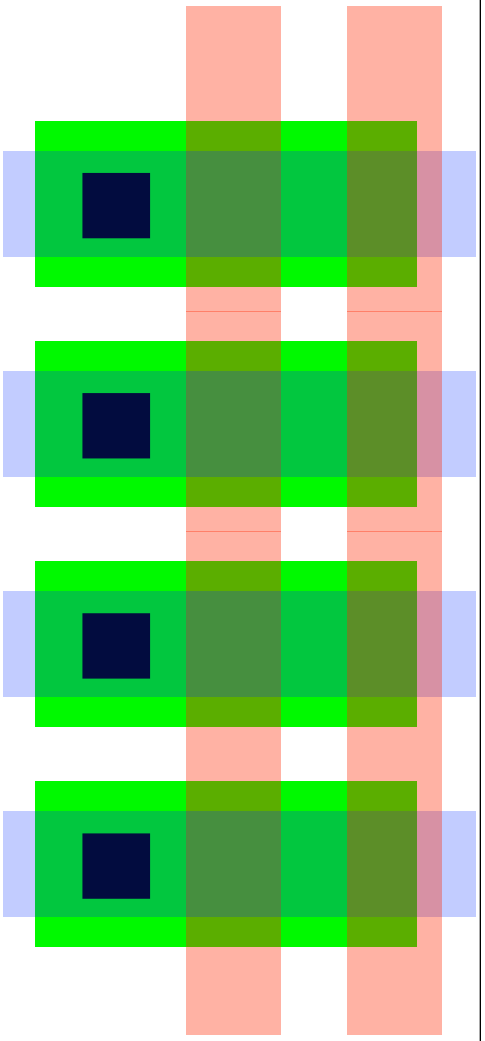


Solution: "Refresh", by reading cells at regular intervals (tens of milliseconds)

Diode leakage ...

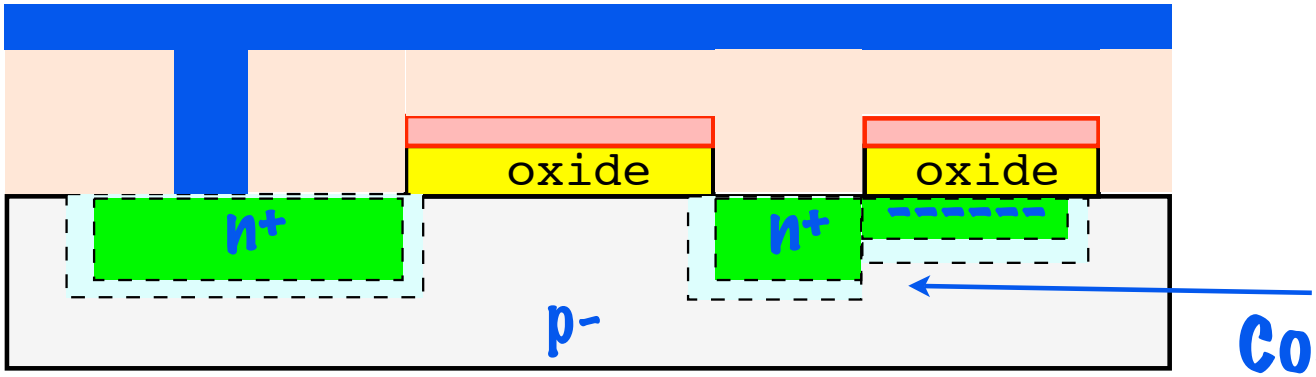
DRAM Challenge #5: Cosmic Rays ...





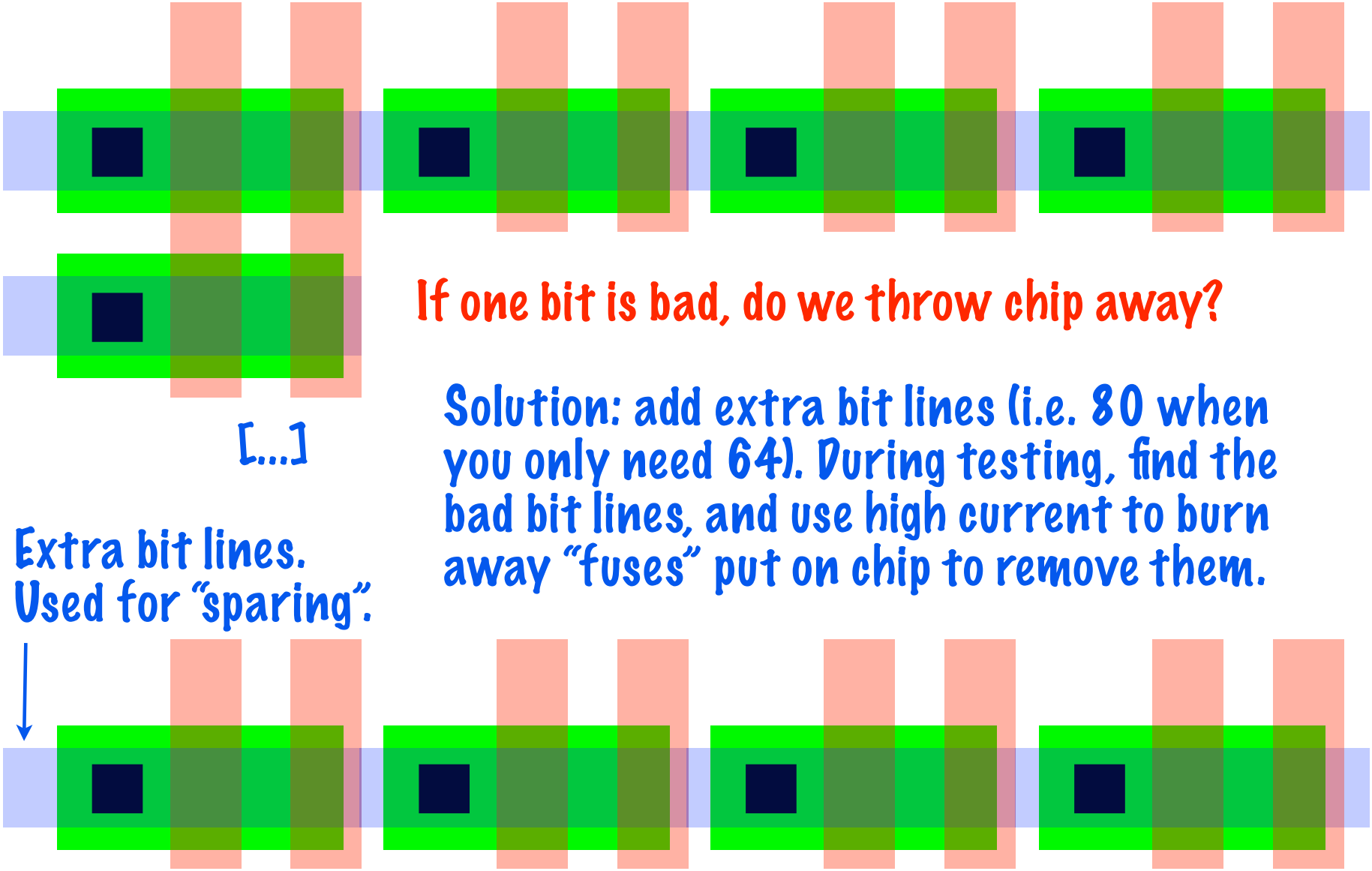
Cell capacitor holds 25,000 electrons (or less). Cosmic rays that constantly bombard us can release the charge!

Solution: Store extra bits to detect and correct random bit flips (ECC).



Cosmic ray hit.

DRAM Challenge 6: Yield



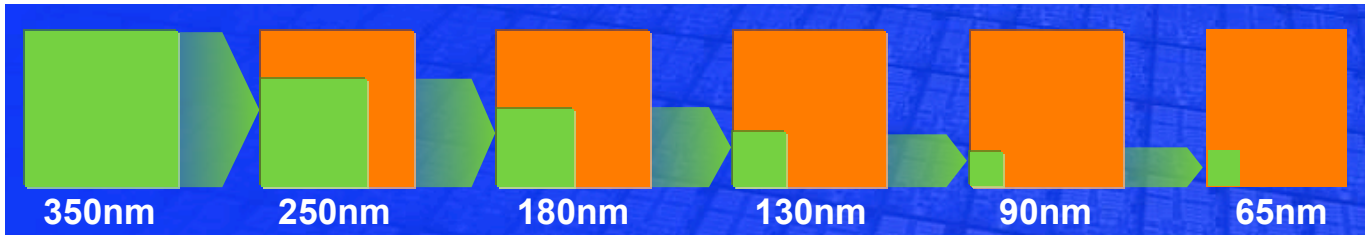
If one bit is bad, do we throw chip away?

Solution: add extra bit lines (i.e. 80 when you only need 64). During testing, find the bad bit lines, and use high current to burn away "fuses" put on chip to remove them.

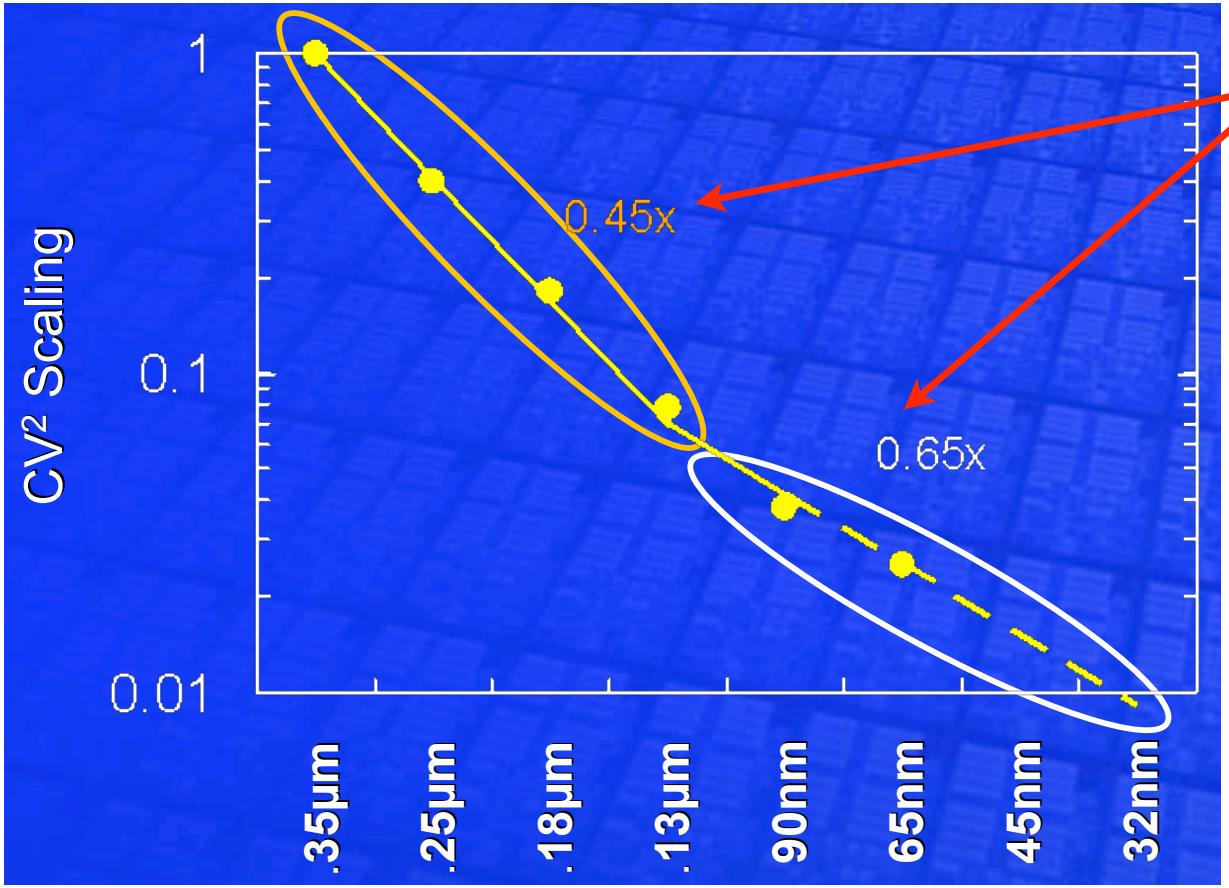
[...]

Extra bit lines.
Used for "sparing".

Recall: Process Scaling



Recall process scaling ("Moore's Law")

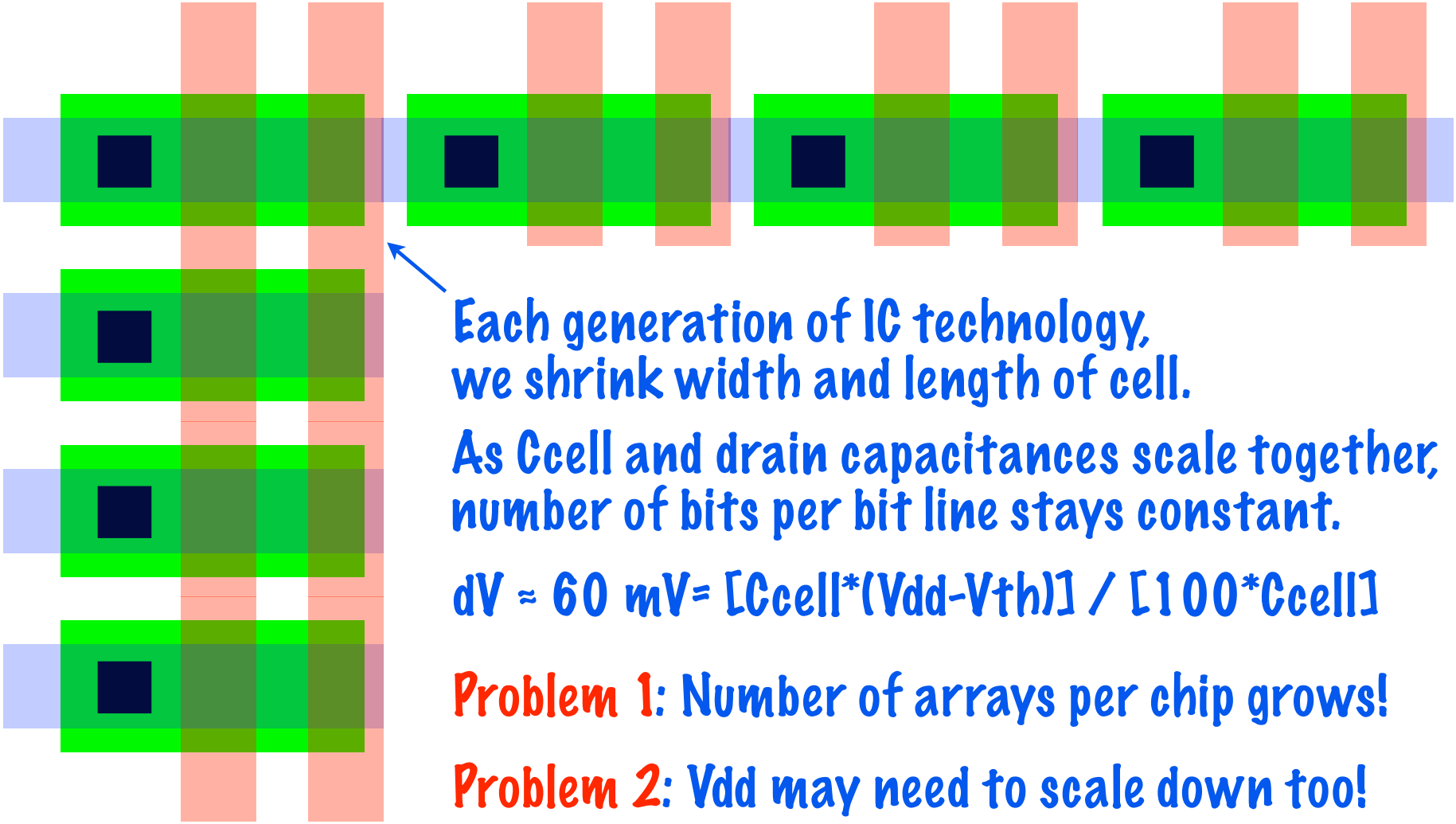


Due to reducing V and C (length and width of Cs decrease, but plate distance gets smaller).

Recent slope more shallow because V is being scaled less aggressively.

From: "Facing the Hot Chips Challenge Again", Bill Holt, Intel, presented at Hot Chips 17, 2005.

DRAM Challenge 7: Scaling



Each generation of IC technology, we shrink width and length of cell.

As Ccell and drain capacitances scale together, number of bits per bit line stays constant.

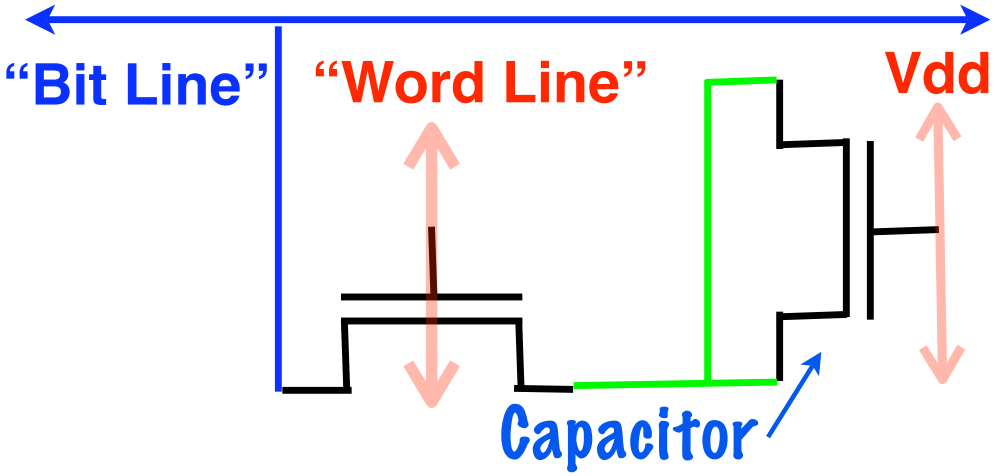
dV ≈ 60 mV= [Ccell*(Vdd-Vth)] / [100*Ccell]

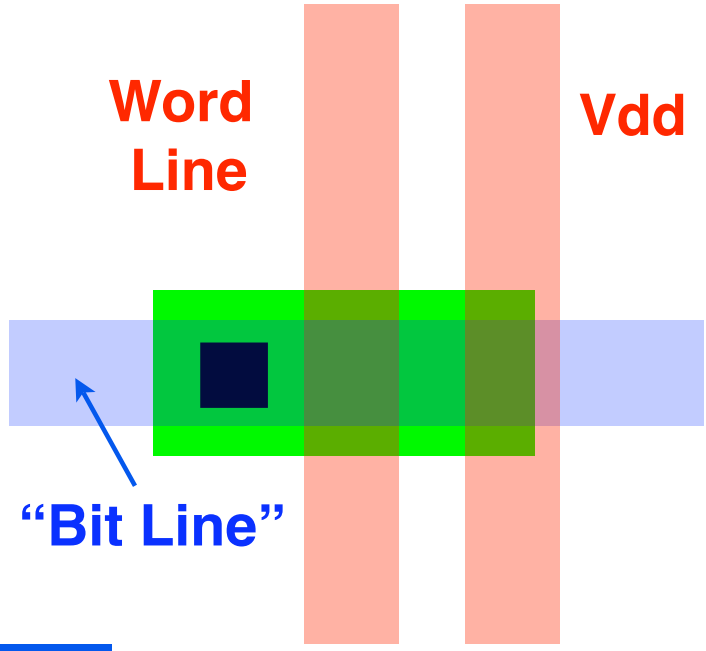
Problem 1: Number of arrays per chip grows!

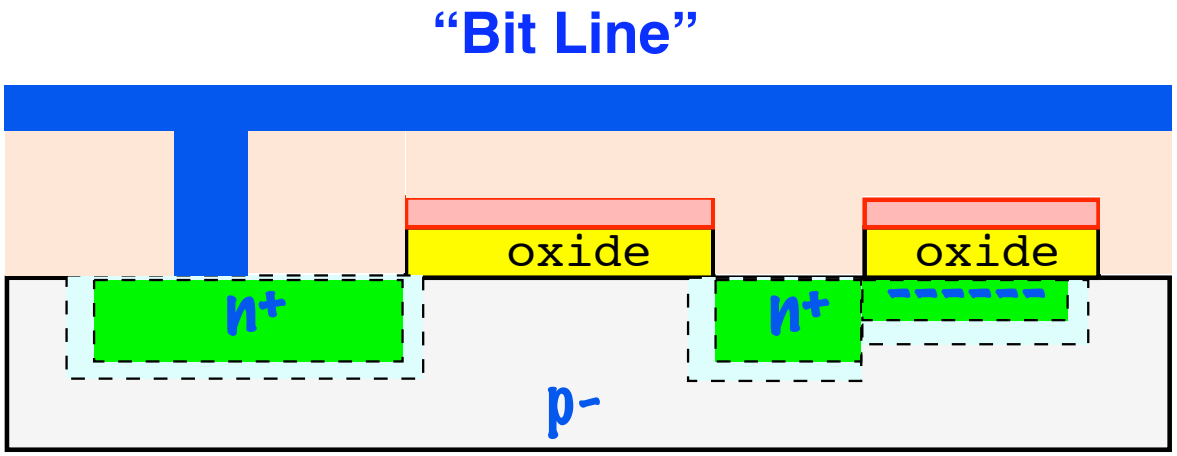
Problem 2: Vdd may need to scale down too!

Solution: Constant Innovation of Cell Capacitors!

Poly-diffusion Ccell is ancient history

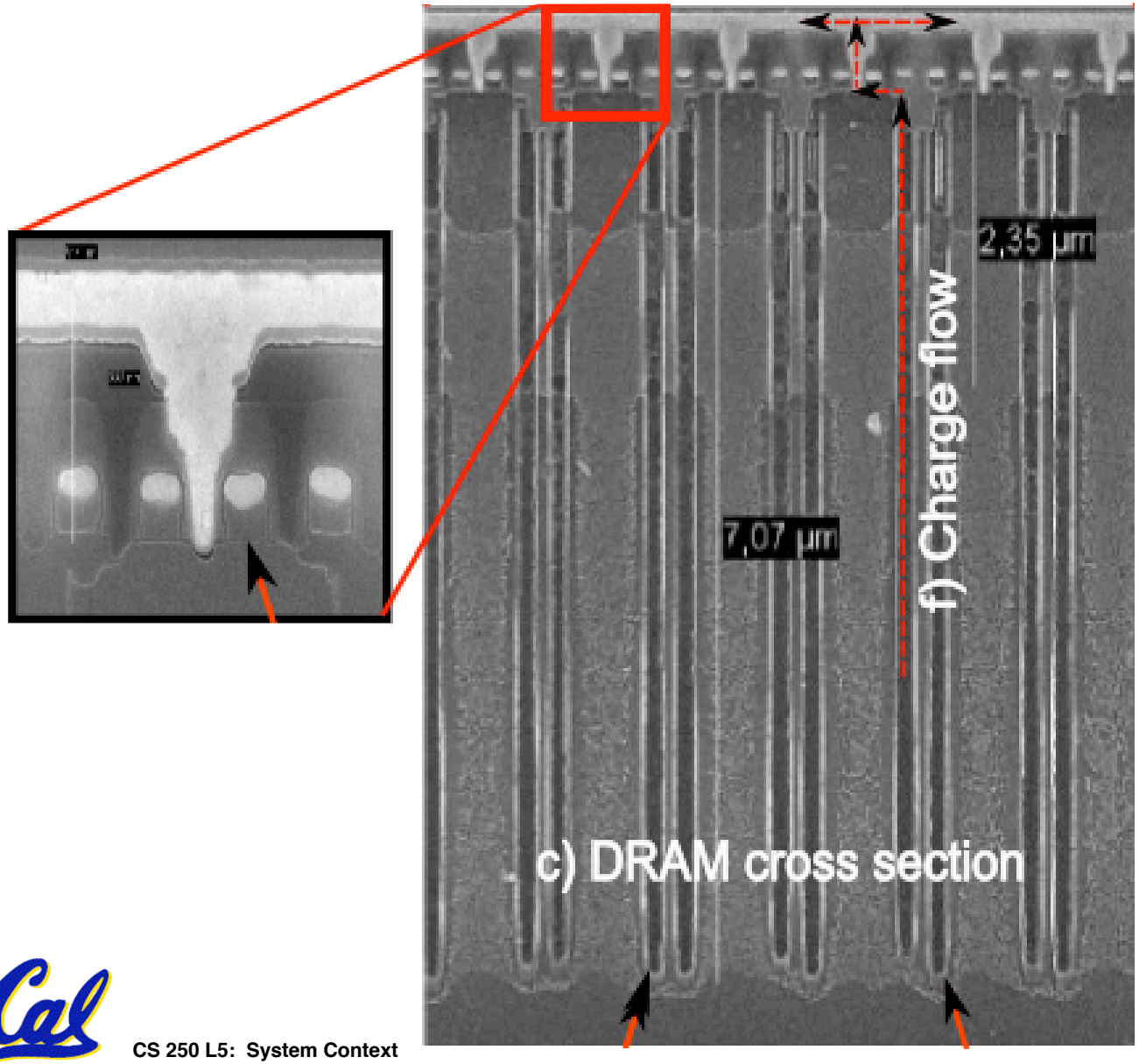






Word Line and Vdd run on "z-axis"

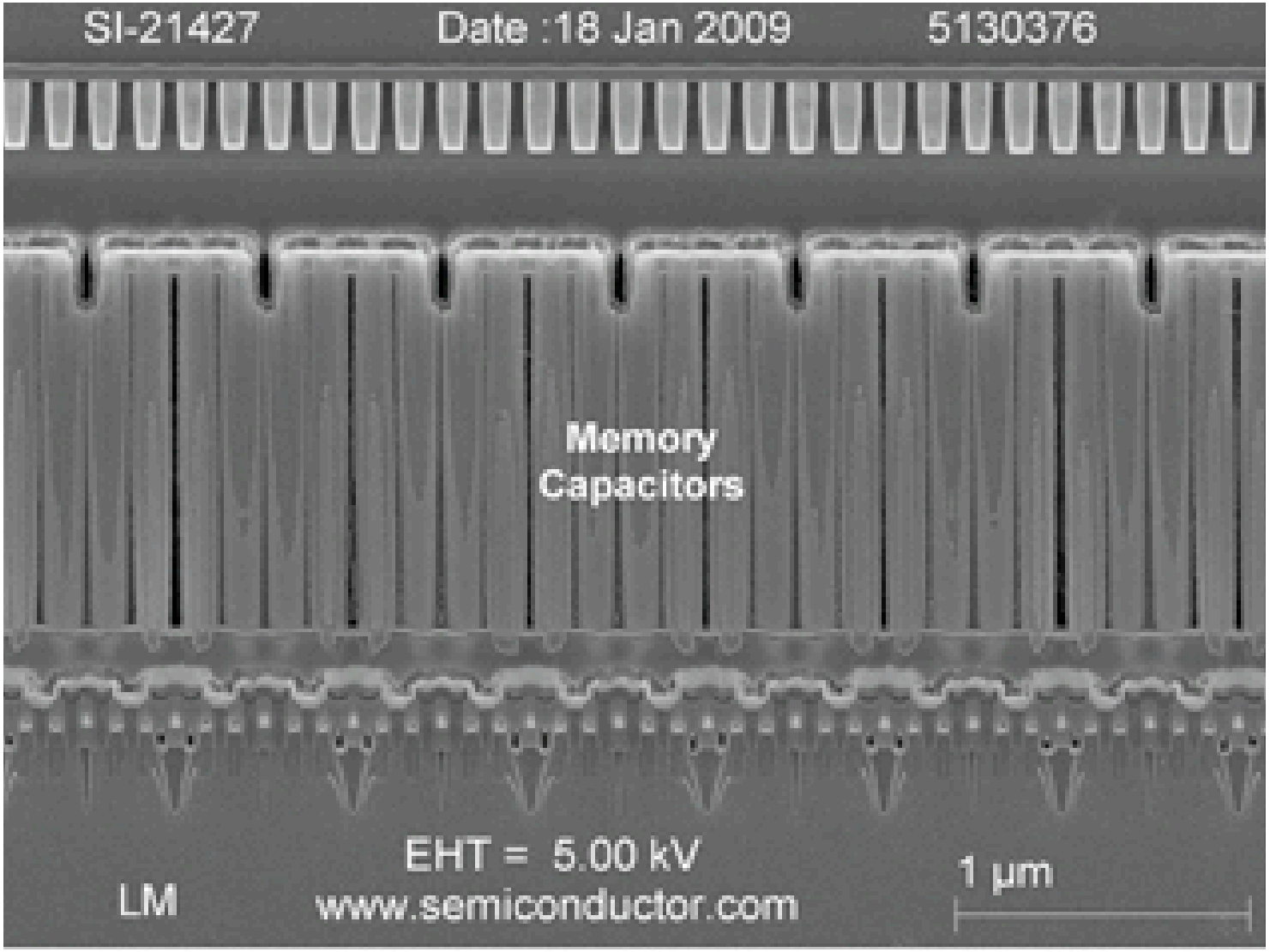
Early replacement: “Trench” capacitors



No longer competitive as commodity DRAM parts.

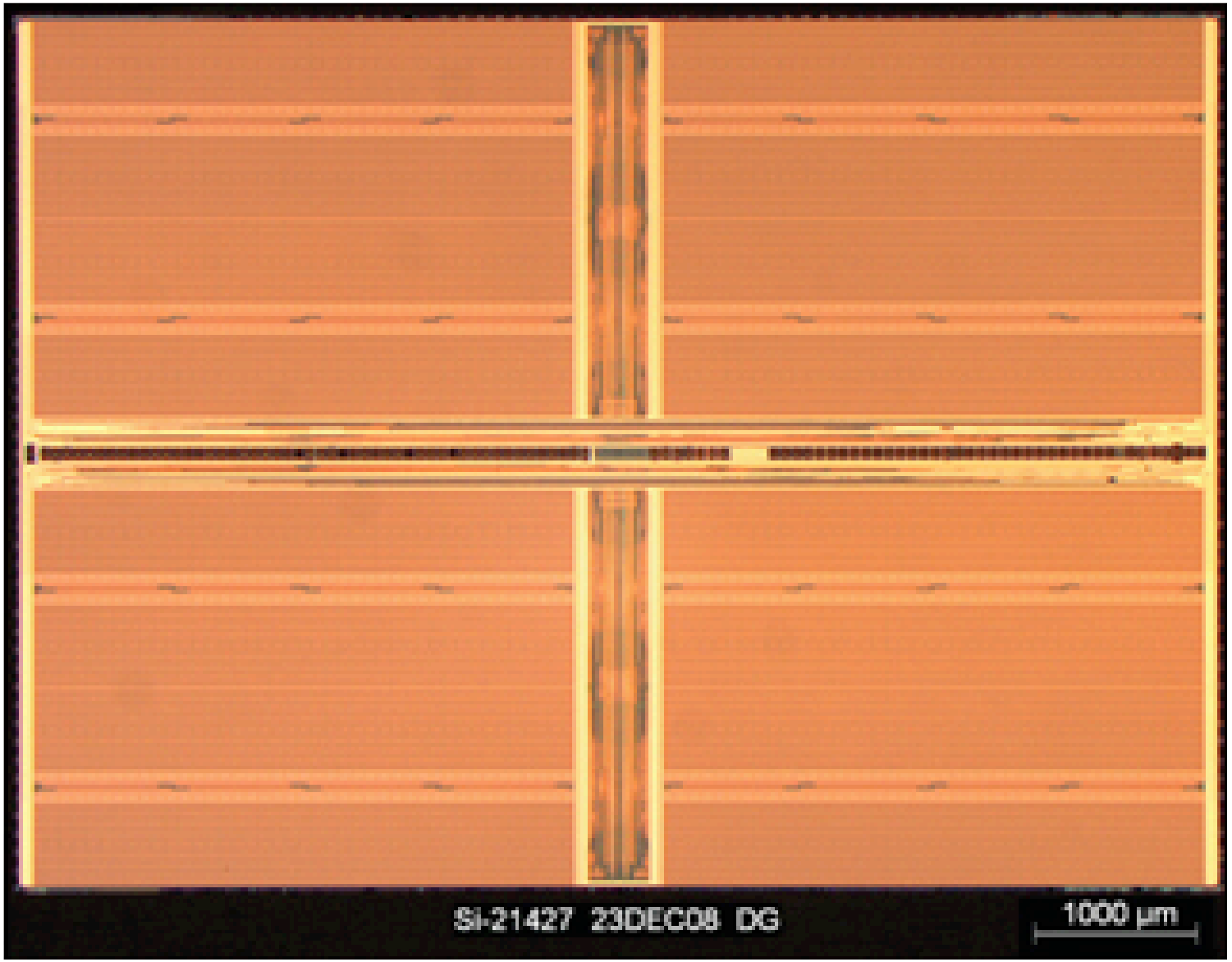
Still used as embedded DRAM in logic processes.

Modern cells: “stacked” capacitors



Micron 1-Gbit DDR2 50-nm SDRAM

Micron 50nm 1-Gbit DDR2 die photo



Memory Arrays

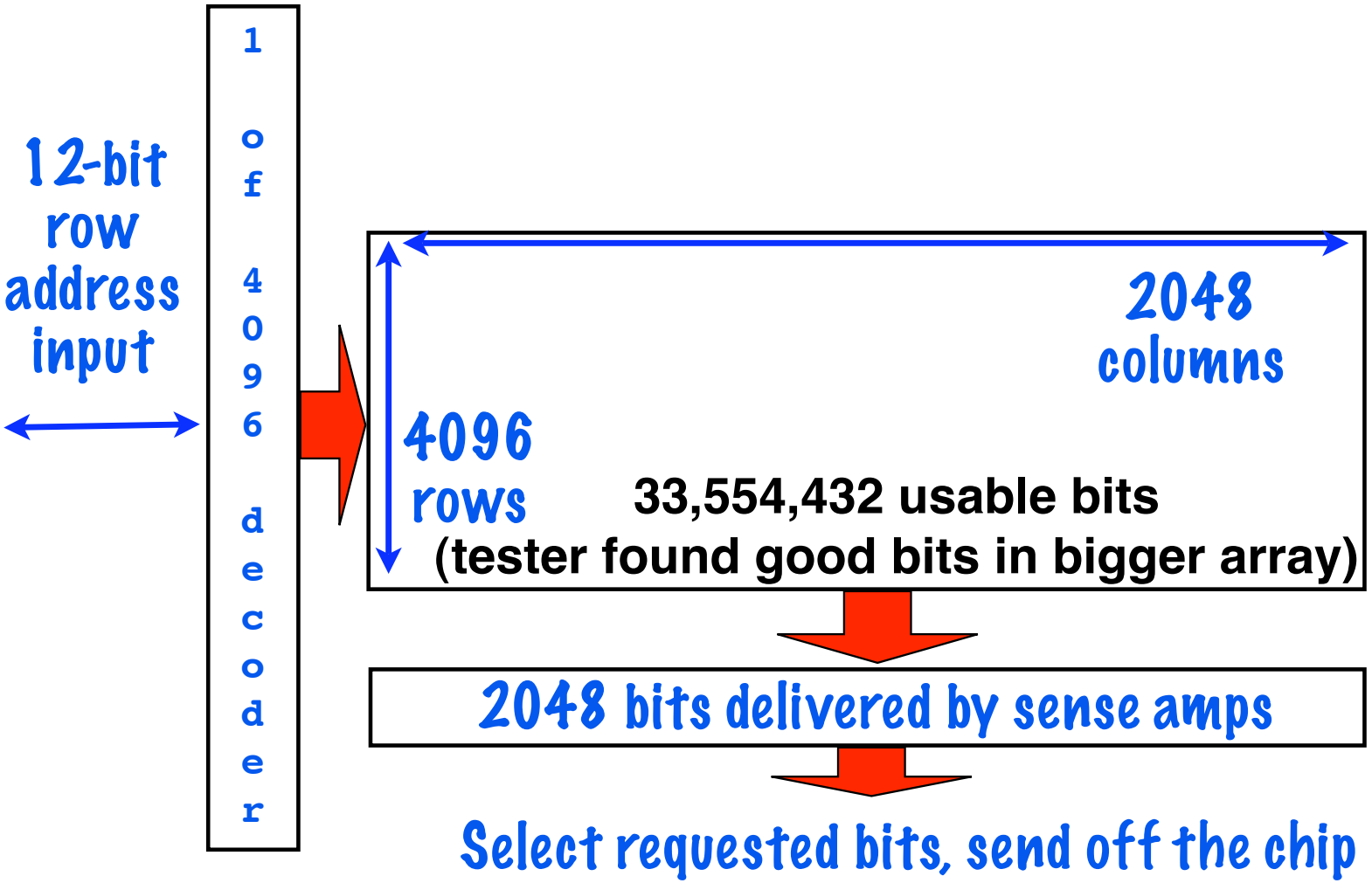
Older SDRAM part: 133 Mhz, 128 Mb

SYNCHRONOUS DRAM

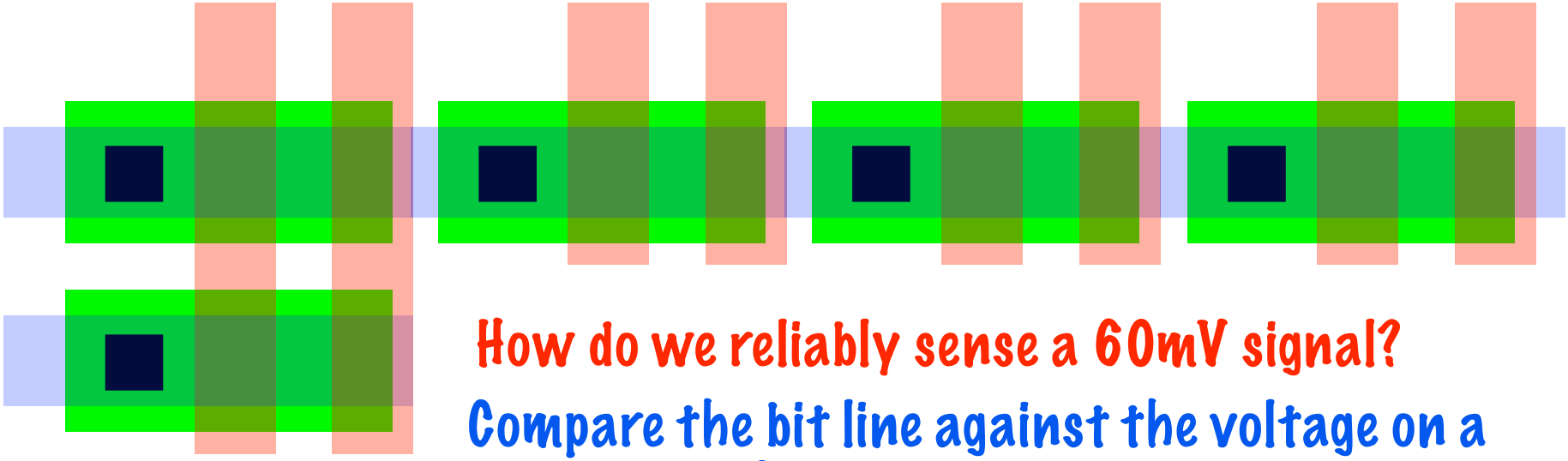
MT48LC32M4A2 – 8 Meg x 4 x 4 banks
MT48LC16M8A2 – 4 Meg x 8 x 4 banks
MT48LC8M16A2 – 2 Meg x 16 x 4 banks

For the latest data sheet, please refer to the Micron Web site: www.micron.com/dramds

A "bank" of 32 Mb (128Mb chip -> 4 banks)



Recall DRAM Challenge #3b: Sensing



How do we reliably sense a 60mV signal?

Compare the bit line against the voltage on a "dummy" bit line.

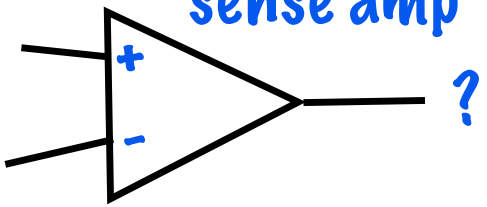
[...]

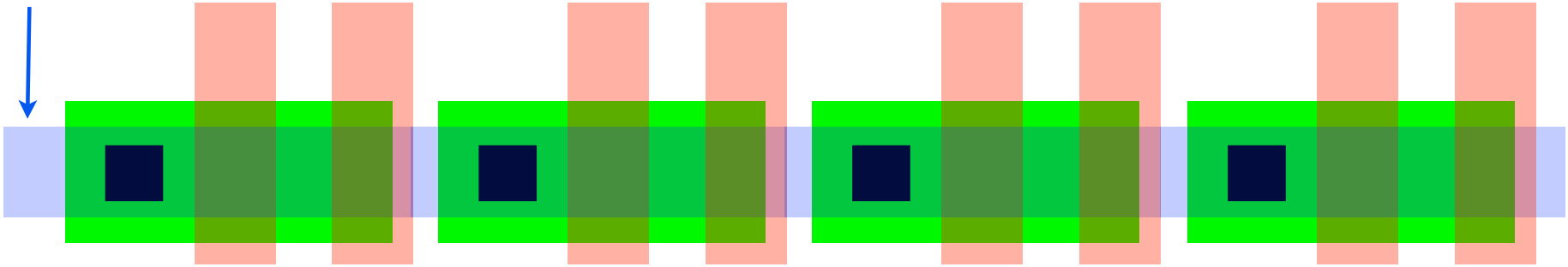
"Dummy" bit line.
Cells hold no charge.

Bit line to sense

Dummy bit line

"sense amp"

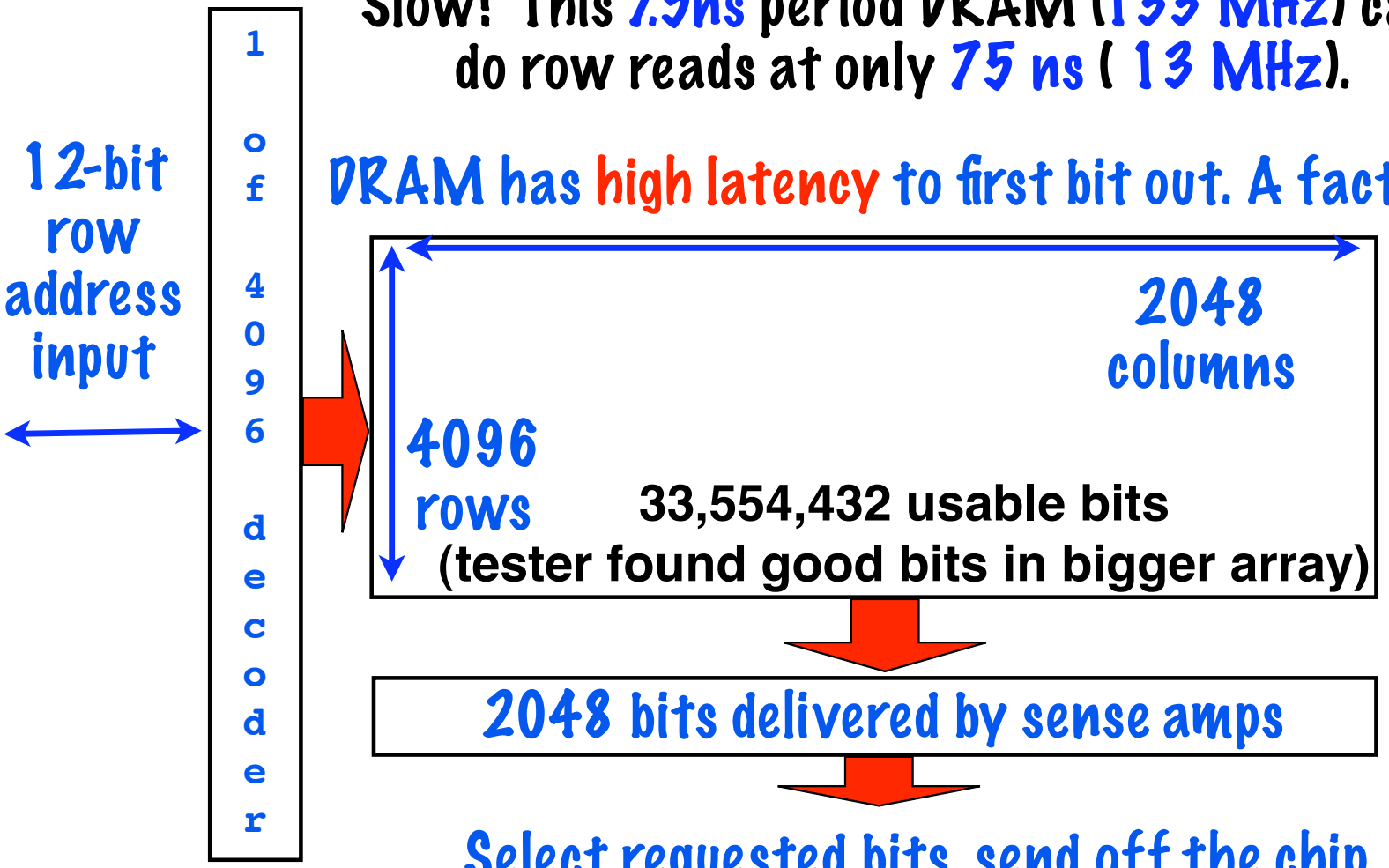




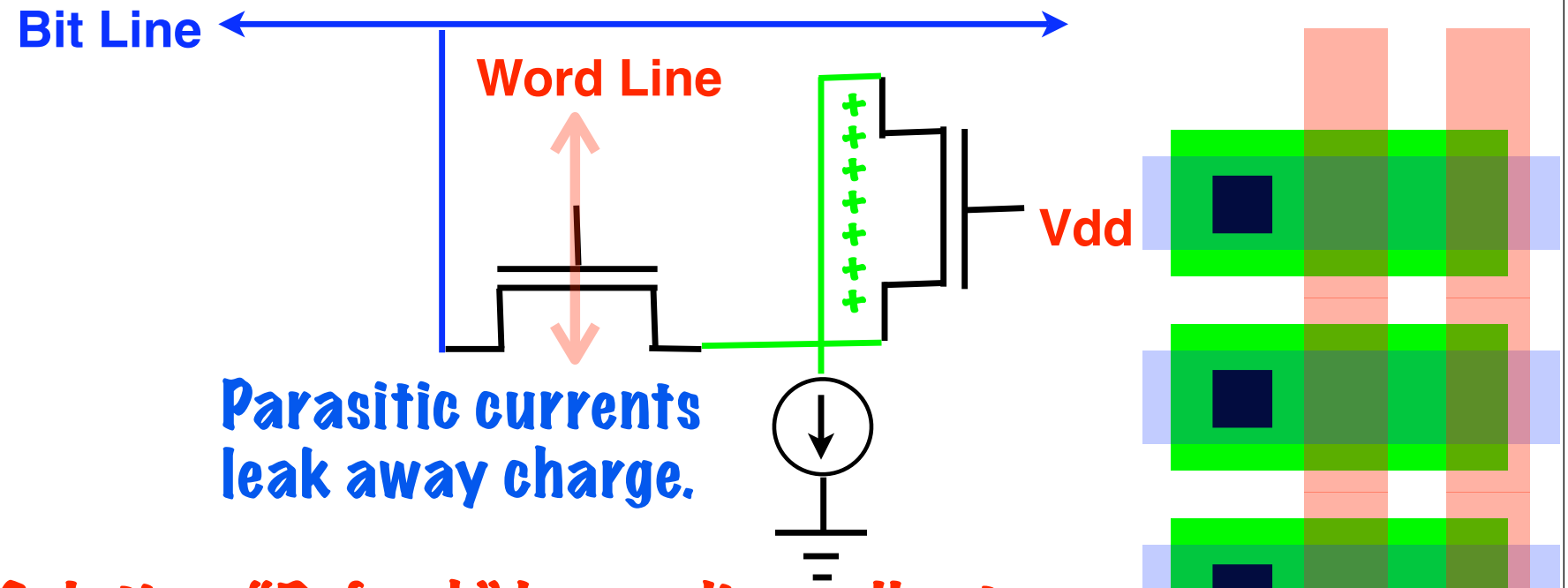
"Sensing" is row read into sense amps

Slow! This 7.5ns period DRAM (133 MHz) can do row reads at only 75 ns (13 MHz).

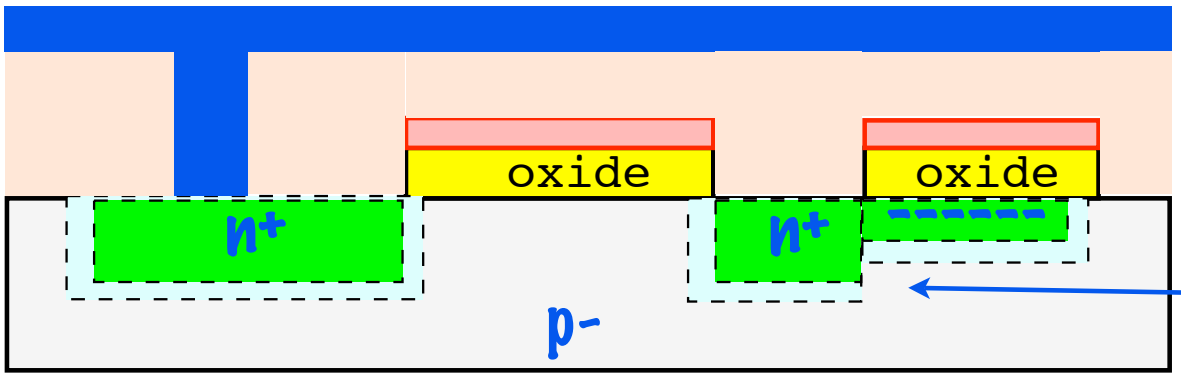
DRAM has high latency to first bit out. A fact of life.



An ill-timed refresh may add to latency



Solution: "Refresh", by reading cells at regular intervals (tens of milliseconds)



Diode leakage ...

Latency is not the same as bandwidth!

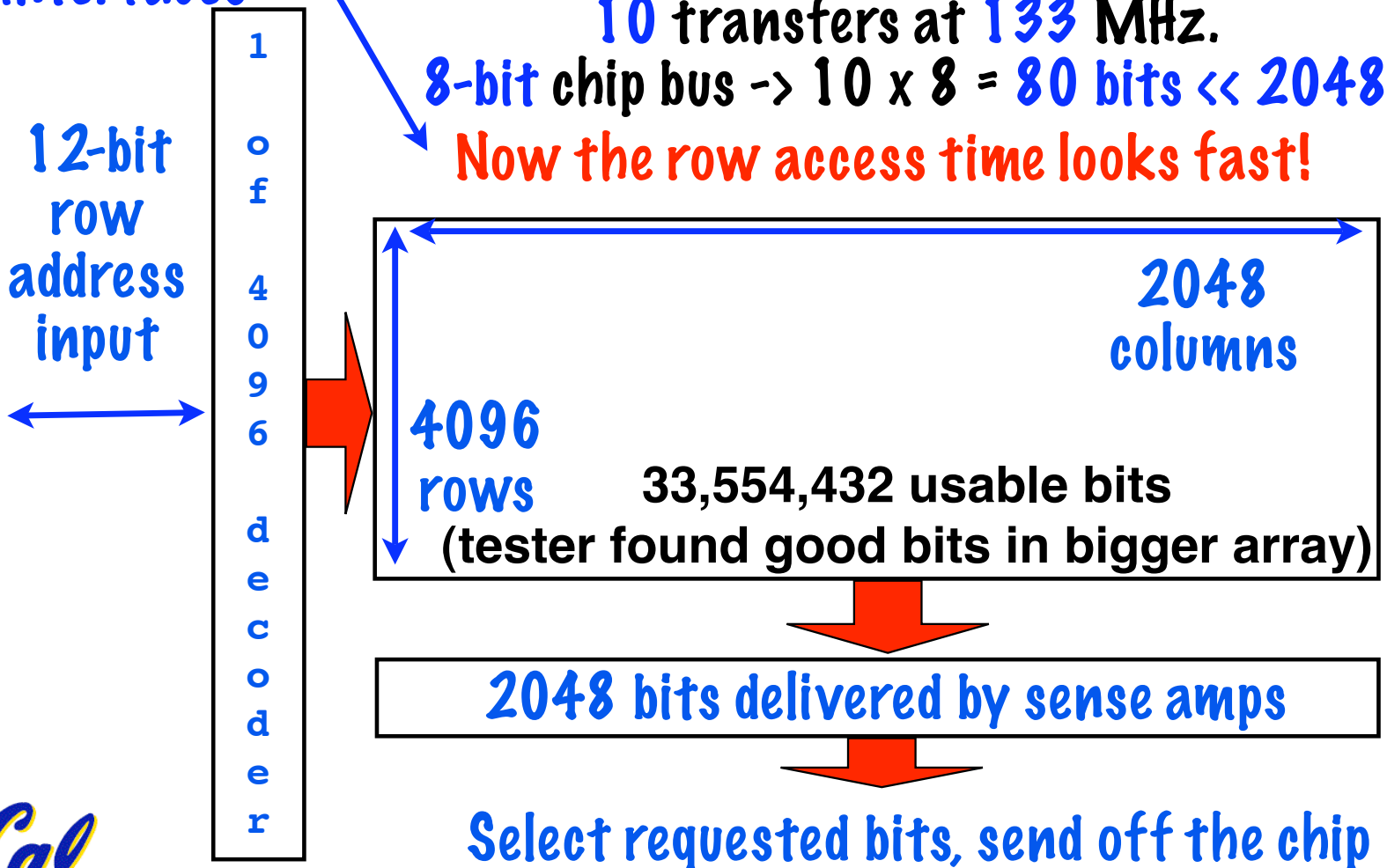
Thus, push to faster DRAM interfaces

What if we want all of the 2048 bits?

In row access time (75 ns) we can do 10 transfers at 133 MHz.

8-bit chip bus -> 10 x 8 = 80 bits << 2048

Now the row access time looks fast!

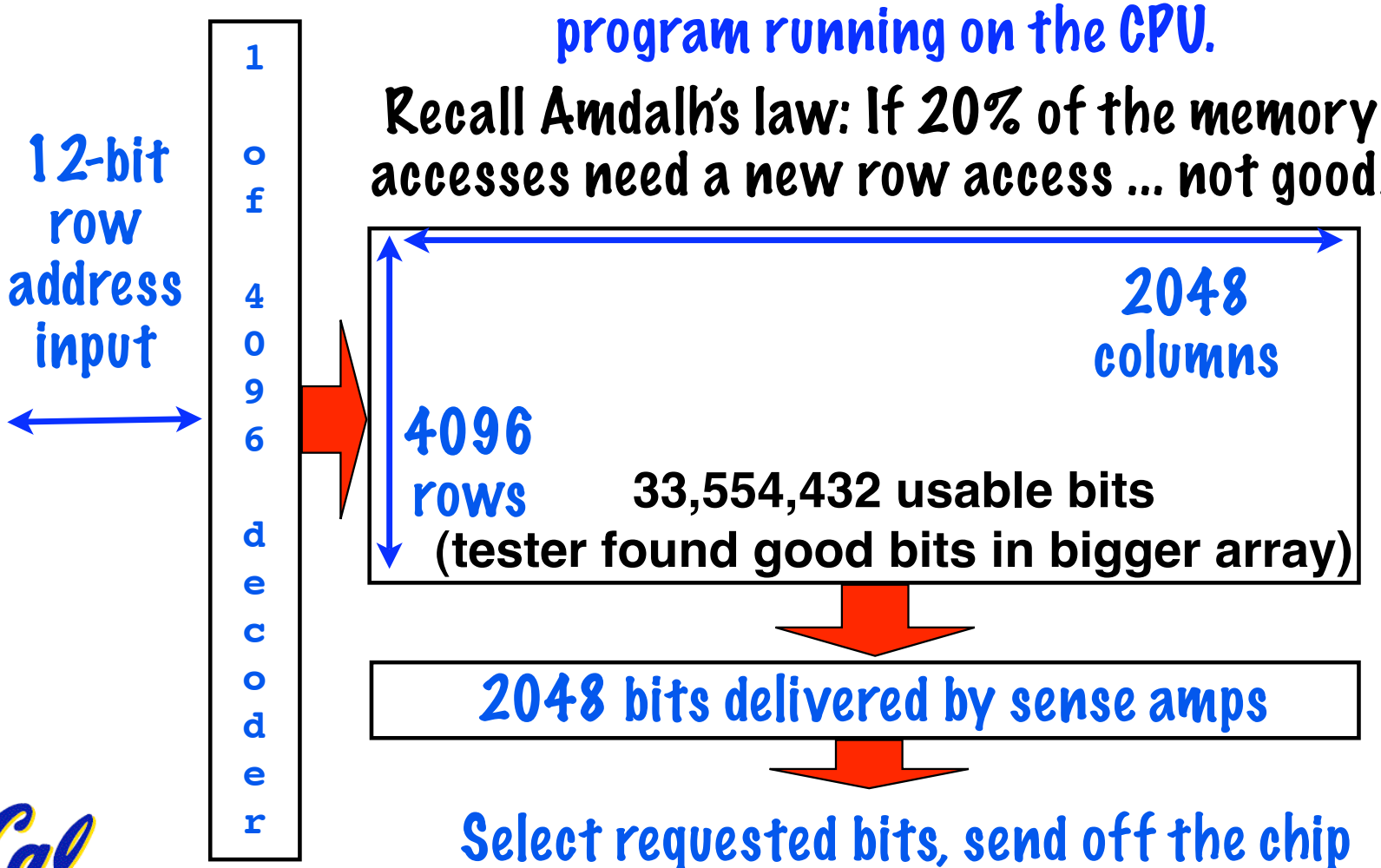


Sadly, it's rarely this good ...

What if we want all of the 2048 bits?

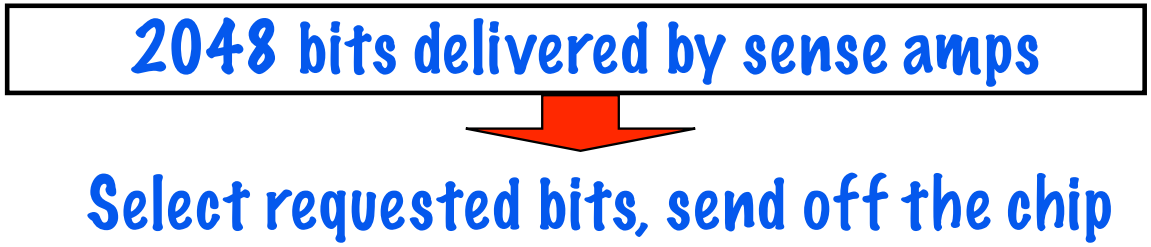
The "we" for a CPU would be the program running on the CPU.

Recall Amdalh's law: If 20% of the memory accesses need a new row access ... not good.



DRAM latency/bandwidth chip features

- **Columns:** Design the right interface for CPUs to request the subset of a column of data it wishes:

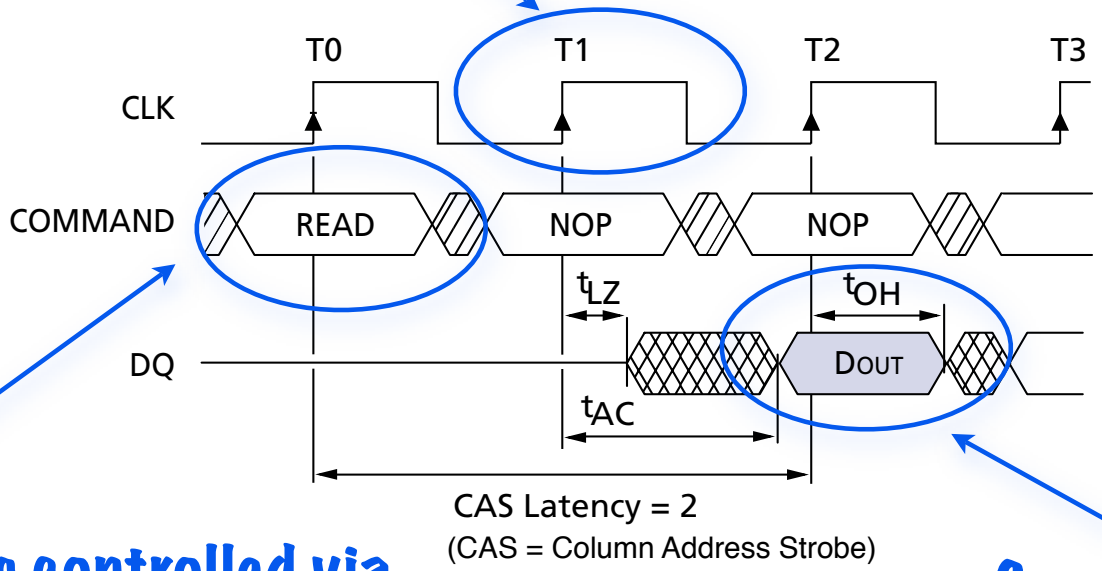


- **Interleaving:** Design the right interface to the 4 memory banks on the chip, so several row requests run in parallel.

Off-chip interface for the Micron part ...

A clocked bus protocol (133 MHz)

Note! This example is best-case! To access a new row, a slow ACTIVE command must run before the READ.

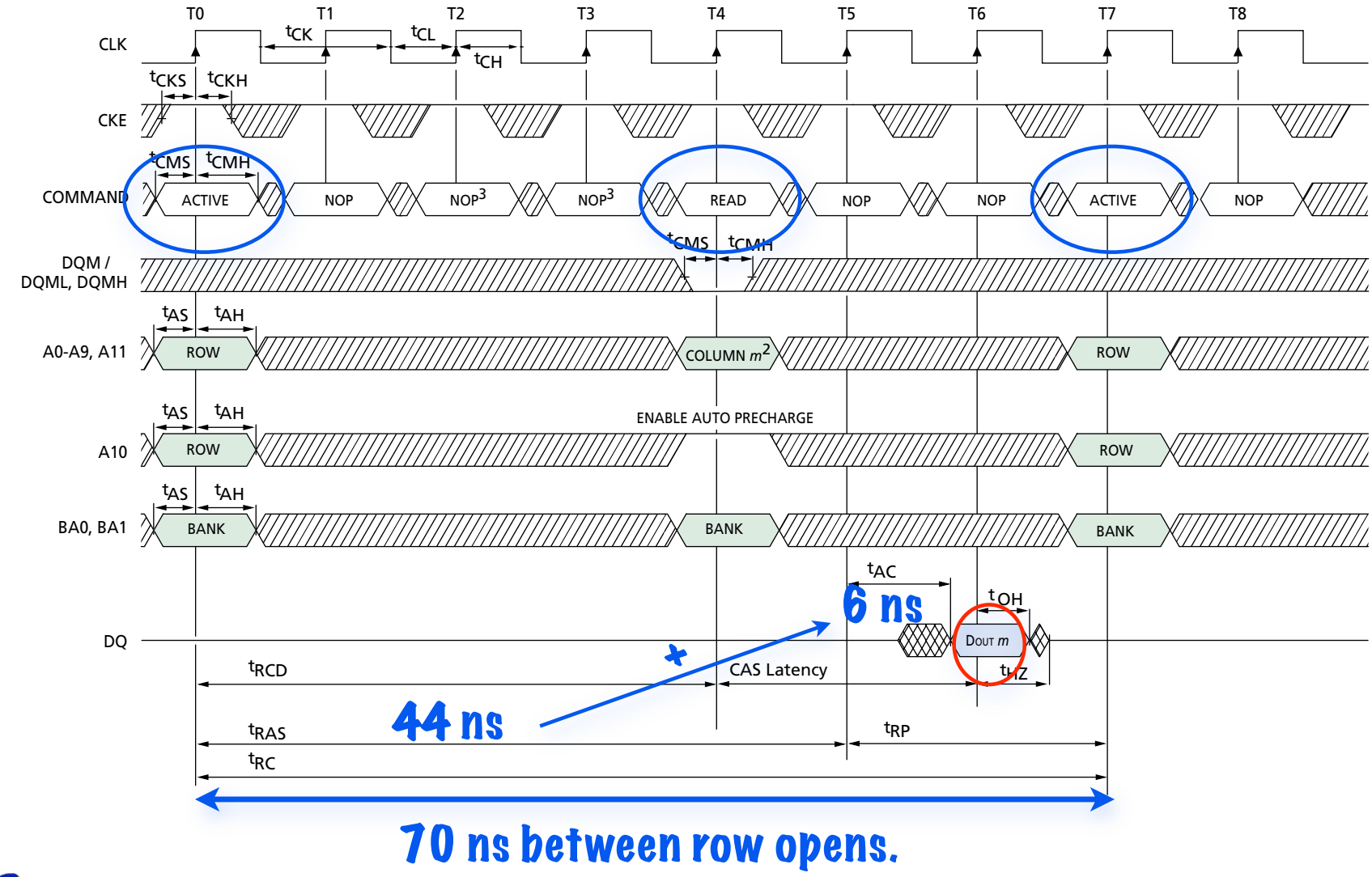


DRAM is controlled via commands (READ, WRITE, REFRESH, ...)

Synchronous data output.

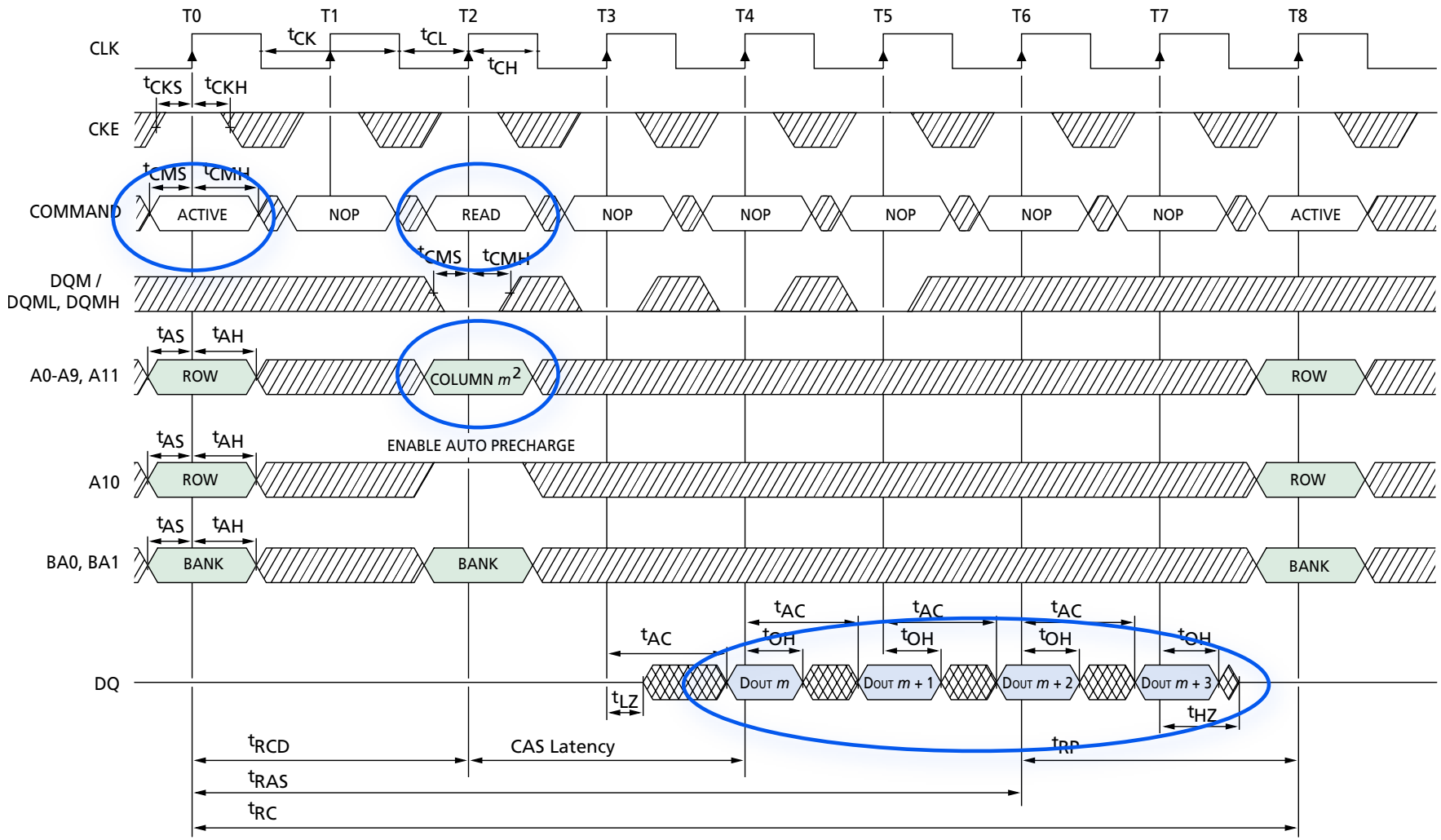
From Micron 128 Mb SDRAM data sheet (on "resources" web page)

Opening a row before reading ...

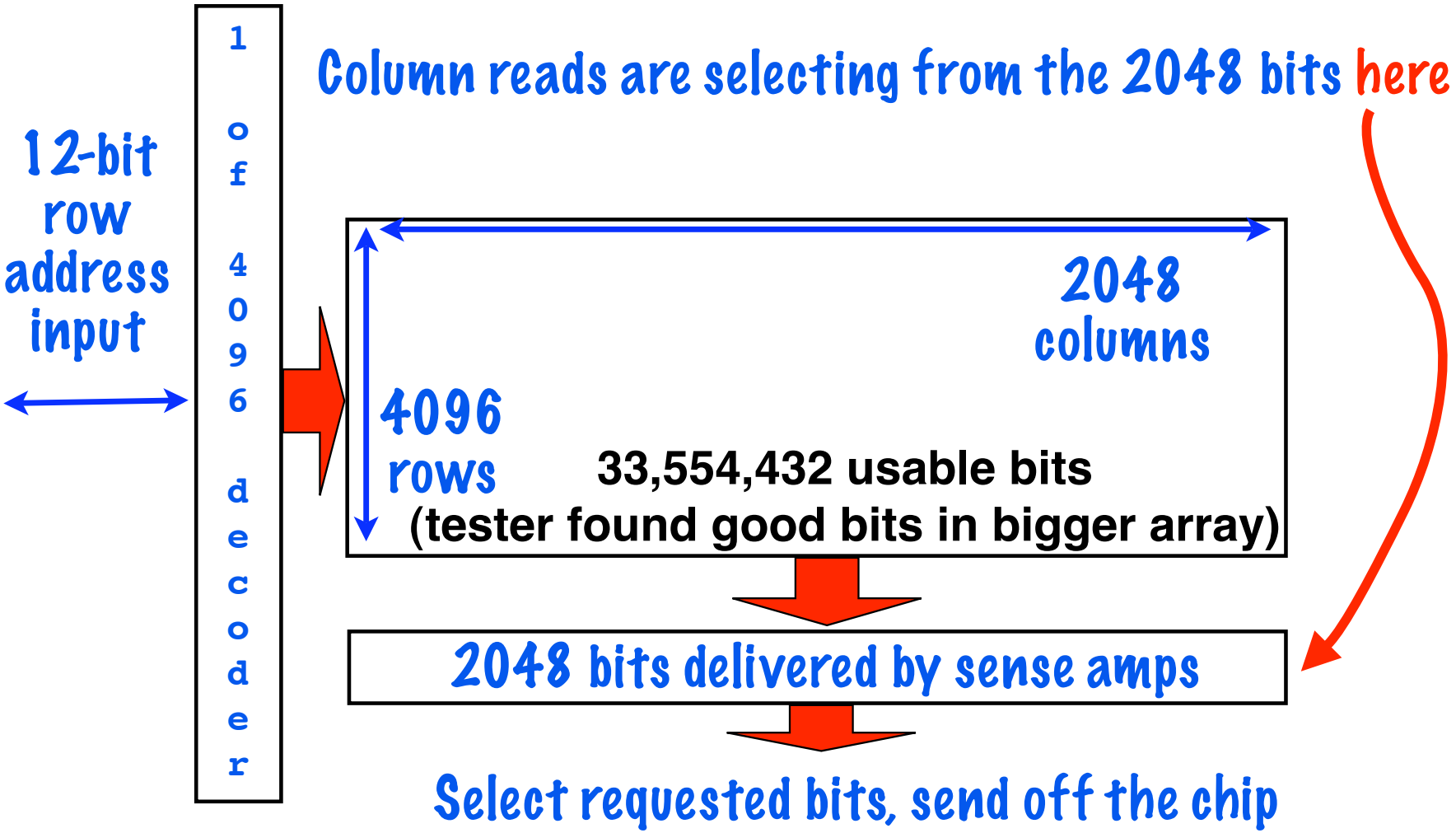


Note: Example uses Auto-Precharge, 100Mhz part

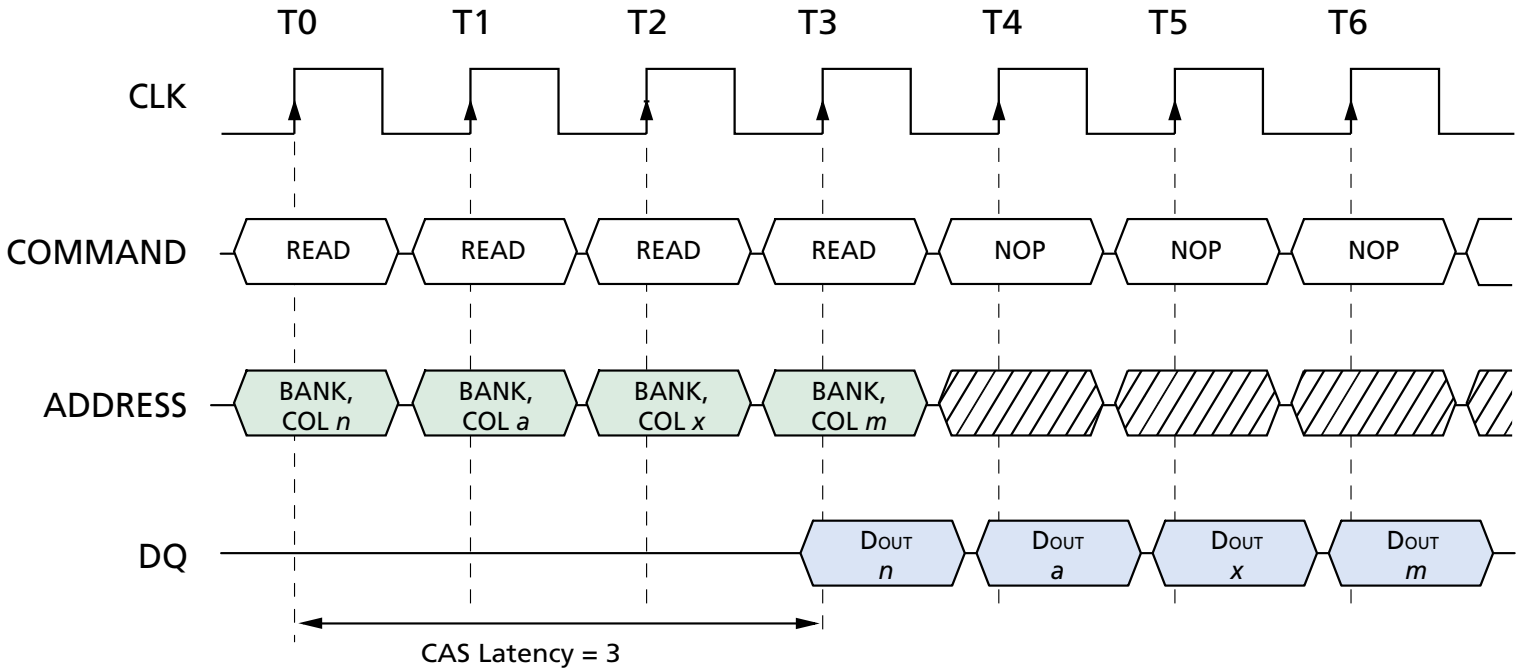
However, we can read columns quickly



Why? Reading "delivered bits" is fast.



Interleave: Access all 4 banks in parallel



NOTE: Each READ command may be to any bank. DQM is LOW.

Figure 8
Random READ Accesses

Next Class: Paper Readings

Implementing the Scale Vector-Thread Processor. Ronny Krashinsky, Christopher Batten, and Krste Asanovic. ACM Transactions on Design Automation of Electronic Systems (TODAES), 13(3), 41:1-41:24, July 2008. paper
Energy-Performance Tradeoffs in Processor Architecture and Circuit Design: A Marginal Cost Analysis. Omid Azizi, Aqeel Mahesri, Benjamin C. Lee, Sanjay J. Patel, Mark Horowitz. Proceedings of the 31th International Symposium on Computer Architecture (ISCA-37) 2010. paper