CS 250
VLSI System Design

Lecture 12 – System Context

2011-10-10
John Wawrzynek and Krste Asanović
with John Lazzaro

TA: Brian Zimmer

www-inst.eecs.berkeley.edu/~cs250/
How does the CPU interact with the system?
Main board: identify sub-systems ...
Main board: identify sub-systems...

- Touchpad controller (underside)
- FLASH drive module
Intel Core i5: Sandy Bridge

All on chip:

- x86 cores
- GPU
- North Bridge
- DRAM controller

On chip ring network
Today: serial I/O + DRAM

Thunderbolt: Four 10 Gb/s point-to-point links.

DRAM: From device physics to bus protocols.
"Active Cables"

Each pair is a 10 Gb/s point-to-point link
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

Sending Chip
CMOS Inverter

PC Board Trace or Cable

Receiving Chip
CMOS Inverter

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.
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.

\[ Z_0 = 50\Omega \]

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 $L/c_w = 4$ns.
Inverter models.

- Typical output impedance is 400 ohms.

Sending Chip

CMOS

Inverter

Typical input impedance of pad is 1000 ohms.

Receiving Chip

CMOS

Inverter

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 reflection waves.
Incident-Wave Signaling
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).

Energy dominated by DC power

Like an SRAM sense amp. Differential, senses sign of voltage.
Line Equalization
Once we adopt the incident-wave approach, what limits our bandwidth?

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

Slow components of the pulse have high amplitude.

Fast components of the pulse have low amplitude.

Wire attenuation
Equalization

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.
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.

How can we overcome intersymbol interference?

---

CMOS High-Speed I/Os — Present and Future

M.-J. Edward Lee\textsuperscript{1}, William J. Dally\textsuperscript{1,2}, Ramin Farjad-Rad\textsuperscript{1}, Hiok-Tiaq Ng\textsuperscript{1}, Ramesh Senthinathan\textsuperscript{1}, John Edmondson\textsuperscript{1}, and John Poulton\textsuperscript{1}
Equalization

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

Pulse: 

Pulse after EQ:

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

What receiver sees when we send EQ’d pulse.
Eye Diagrams
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

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

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

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

- $2t_u$
- $t_a$
- $t_r$

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

Input stream: a “river” of bits

0100101010010010101010101010101 ...

We restrict the characteristics of this river ...
A river of bits ...

01001010100100101010101101010010 ...

\[ M \]

The first \( M \) bits will be received 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.
Given an arbitrary bitstream, we can code it to have these desired properties.

Example: 8b/10b coding

Bits we want to send:

... 0100101010010010101010101101010010 ...

Each 8 bit sequence recoded as 10 bits

... xxxxxxxxxxxx ...

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. Dally¹, Ming-Ju Edward Lee¹, Fu-Tai An¹, John Poulton², and Steve Tell²

CMOS High-Speed I/Os — Present and Future
M.-J. Edward Lee¹, William J. Dally¹,², Ramin Farjad-Rad¹, Hiok-Tiaq Ng¹, Ramesh Senthinathan¹, John Edmondson¹, and John Poulton¹

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. Neves¹
Part II: DRAM
Recall: Capacitors in action

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

After circuit “settles” ...

\[ Q = C \cdot V = C \times 1.5 \text{ Volts (D cell)} \]

\(Q\): Charge stored on capacitor

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

After battery is removed:

Still, \(Q = C \times 1.5\) Volts

Capacitor “remembers” charge
DRAM cell: 1 transistor, 1 capacitor

“Bit Line”

“Word Line”

Vdd

Capacitor

Word Line and Vdd run on “z-axis”

Why Vcap values start out at ground.

Diode leakage current.
A 4 x 4 DRAM array (16 bits) ....
DRAM Circuit Challenge #1: Writing

\[ V_{dd} - V_{th}. \text{ Bad, we store less charge. Why do we not get } V_{dd}? \]

\[ I_{ds} = k \ (V_{gs} - V_{th})^2. \]

"Turns off" when \( V_{gs} \leq V_{th}! \)

\[ V_{gs} = V_{dd} - V_{c}. \text{ When } V_{dd} - V_{c} = V_{th}, \text{ charging effectively stops!} \]
Raising the word line removes the charge from every cell it connects to!

Must write back after each read.
Assume $C_{cell} = 1 \text{ fF}$

Bit line may have 2000 nFet drains, assume bit line $C$ of 100 fF, or 100*$C_{cell}$.

$C_{cell}$ holds $Q = C_{cell}*(V_{dd}-V_{th})$

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

$$dV = \frac{C_{cell}*(V_{dd}-V_{th})}{100*C_{cell}}$$

$$dV = \frac{(V_{dd}-V_{th})}{100} \approx \text{tens of millivolts!}$$

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.

“sense amp”

Bit line to sense
Dummy bit line

[...]

UC Regents Fall 2011 © UCB
DRAM Challenge #4: Leakage ...

Parasitic currents leak away charge.

Solution: “Refresh”, by reading cells at regular intervals (tens of milliseconds)
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).
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”.

Extra bit lines.
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.

DRAM Challenge 7: Scaling

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

As $C_{cell}$ and drain capacitances scale together, number of bits per bit line stays constant.

\[
dV = 60 \text{ mV} = \frac{C_{cell} \cdot (V_{dd} - V_{th})}{100 \cdot C_{cell}}
\]

Problem 1: Number of arrays per chip grows!

Problem 2: $V_{dd}$ 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

Memory 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**

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT48LC32M4A2</td>
<td>8 Meg x 4 x 4 banks</td>
</tr>
<tr>
<td>MT48LC16M8A2</td>
<td>4 Meg x 8 x 4 banks</td>
</tr>
<tr>
<td>MT48LC8M16A2</td>
<td>2 Meg x 16 x 4 banks</td>
</tr>
</tbody>
</table>

For the latest data sheet, please refer to the Micron Web site: [www.micron.com/dramds](http://www.micron.com/dramds)
A “bank” of 32 Mb (128Mb chip => 4 banks)

- 12-bit row address input
- 4096 decoder
- 4096 rows
- 2048 columns
- 33,554,432 usable bits (tester found good bits in bigger array)
- 2048 bits delivered by sense amps

Select requested bits, send off the chip
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.

12-bit row address input

1 of 4096 decoder

4096 rows

33,554,432 usable bits (tester found good bits in bigger array)

2048 columns

2048 bits delivered by sense amps

Select requested bits, send off the chip
An ill-timed refresh may add to latency

Parasitic currents leak away charge.

Solution: “Refresh”, by reading cells at regular intervals (tens of milliseconds)
Latency is not the same as bandwidth!

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!

Thus, push to faster DRAM interfaces

12-bit row address input

1 of 4096 decoder

4096 rows
33,554,432 usable bits
(tester found good bits in bigger array)

2048 columns

2048 bits delivered by sense amps

Select requested bits, send off the chip
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 Amdahl’s law: If 20% of the memory accesses need a new row access ... not good.

33,554,432 usable bits
(tester found good bits in bigger array)

2048 bits delivered by sense amps

Select requested bits, send off the chip
**DRAM latency/bandwidth chip features**

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

- **2048 bits delivered by sense amps**
- Select requested bits, send off the chip

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

- Bank 1
- Bank 2
- Bank 3
- Bank 4
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.

CAS Latency = 2
(CAS = Column Address Strobe)

From Micron 128 Mb SDRAM data sheet (on “resources” web page)
Opening a row before reading ...

Note: Example uses Auto-Precharge, 100Mhz part

70 ns between row opens.
However, we can read columns quickly
Why? Reading “delivered bits” is fast.

Column reads are selecting from the 2048 bits here.

33,554,432 usable bits
(tester found good bits in bigger array)

2048 bits delivered by sense amps

Select requested bits, send off the chip.
Interleave: Access all 4 banks in parallel

Figure 8
Random READ Accesses

NOTE: Each READ command may be to any bank. DQM is LOW.
Next Class: Project Proposals

Initial project proposal presentations.