EE273 Lecture 12
Clock Distribution
November 4, 1998

William J. Dally
Computer Systems Laboratory
Stanford University
billd@csl.stanford.edu

Today's Assignment

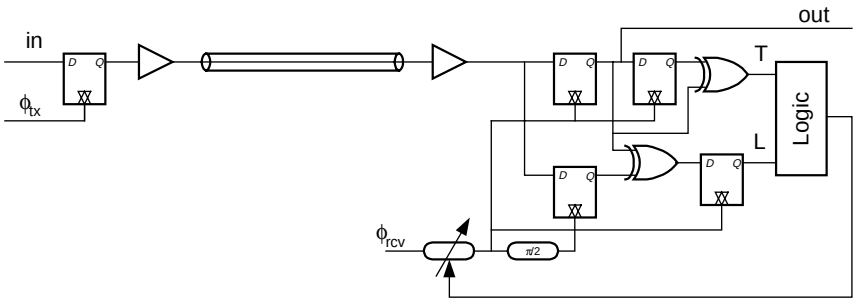
- Problem set 6 (the last one!)
  - SPICE a clock-distribution network
  - Exercise 9-6 - Spice your phase comparator and demonstrate that it has no *dead zone*
- Reading
  - Sections 10.1 and 10.2
  - Complete before class on Monday 11/9

A Quick Overview

- Per-line closed loop timing
  - timing loop closed individually around each data line
  - most sources of skew compensated
- The clock-distribution problem
  - want lots (10^5) of clock loads to see the clock rise at precisely the same picosecond
  - noise and variations in delay of distribution network make this difficult
- Off-chip Clock Distribution
  - wires are good LC transmission lines
  - typically use a tree of clock buffers
- On-chip Clock Distribution
  - wires are slow RC transmission lines
  - buffer delay is modulated by power-supply noise
  - still use a clock tree, but concerns are different

EE273, L12, Nov 4, 1998 3

Link with per-line Closed-Loop Timing

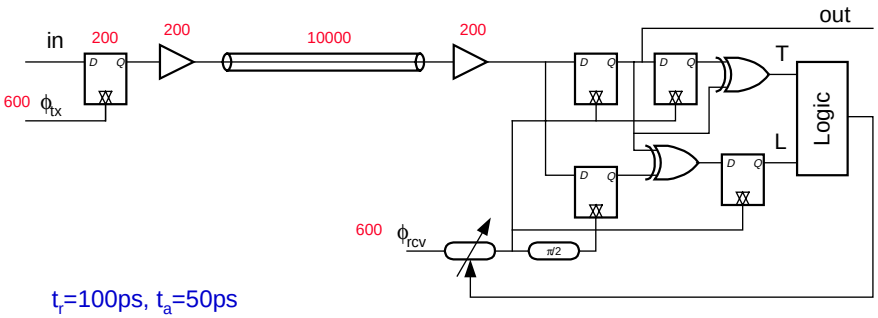


EE273, L12, Nov 4, 1998 4

Now which of these elements of timing uncertainty are cancelled?

- Skew
  - between clock line and data line
  - fixed differences in flip-flop, transmitter, and receiver delays
  - in transmit clock between flip-flops
  - aperture offset in receive flip flop
  - offset in 90 degree delay line
- Jitter
  - in transmit clock
  - in delay of flip-flops, transmitters, and receivers

Per-line Closed-Loop Timing Example



t_r=100ps, t_a=50ps
t_j=10% of active delay (p-p)
t_s=±10% of active, ± 1% of wire delay
t_{dCQ}=200ps, t_{buf}=200ps, t_{clk_buf}=600ps

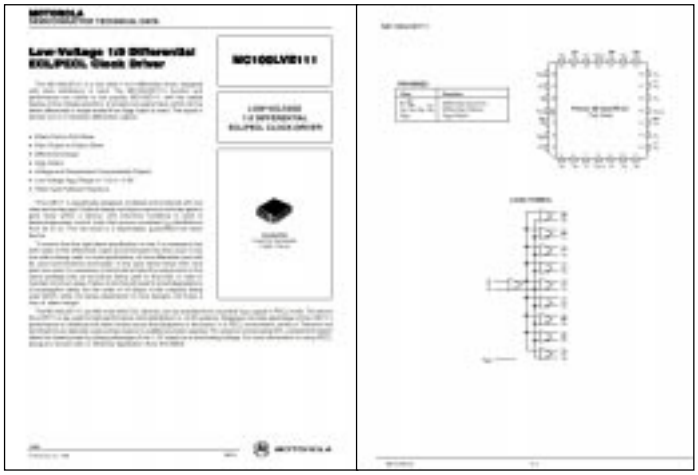
The Clock Distribution Problem

- Some systems have large synchronous clock domains
  - 10s - 100s of chips
  - 10^3 - 10^5 clock loads per chip
  - need to distribute the clock to within 10% of t_{ck}
    - 200ps for a 500MHz clock
- Two step process
  - get the clock to each chip with low skew
  - distribute the clock on each chip with low skew
- Other issues
  - may not want a single point of failure
    - can't have just one clock oscillator or master buffer
  - need to adjust for manufacturing variations with a minimum of labor

Off-Chip Clock Distribution

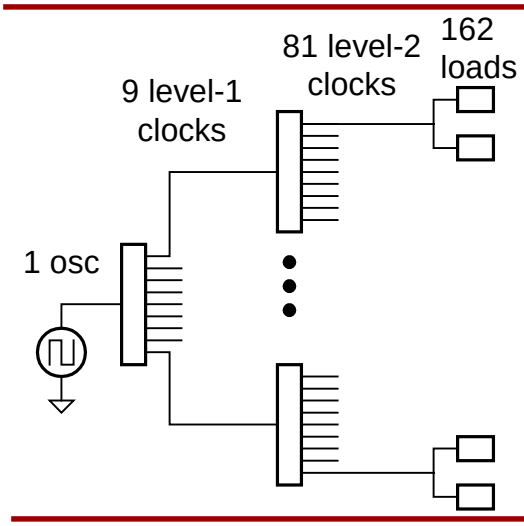
- Relatively easy because wire delays are
  - stable
  - easily characterized
- Most common strategy is a clock tree
  - single oscillator
  - buffer tree with a fanout of N
  - low-skew fanout buffers used
    - e.g., Motorola 100LVE111
  - tuning needed to get very low skews
    - self-tuning series terminated drivers
- For fault tolerance can use an array of oscillators
  - phase lock multiple clock generators to one another
- Traveling waves and standing waves can also be exploited
  - round-trip clock distribution
  - salphasic clock distribution

A Typical Clock-Tree Driver
The 100LVE111



Skew:
50ps within a part
200ps across parts

An Example Clock Tree



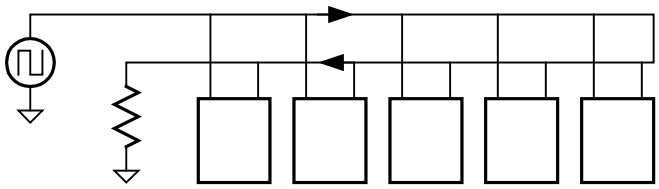
All lines are differential and terminated into 50Ω

Clock Trimming

- If open-loop skew is not good enough, clocks can be *trimmed*
  - adjust the *length* of each fanout to equalize delay
  - trombones
  - RC padding
  - modulation of buffer delay
- This can be done manually or automatically

Round-Trip Distribution

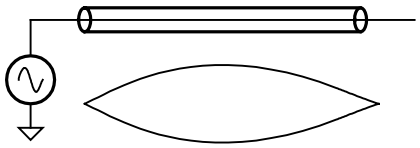
- Send clock down an array of modules (e.g., a bus) and back
- Each module sees forward and reverse traveling clock
- The average time of arrival is the same for all modules
- Modules can interpolate the forward and reverse traveling clocks *or*
- Data can be sent forward with the forward clock and backward with the reverse clock



Salphasic Distribution

- Drive an unterminated (or shorted) line with a sine-wave
- Standing wave has the same phase everywhere
  - just different amplitude
- Amplify and limit this wave to generate a clock with negligible skew!

$$V_f(x,t) = \sin(\omega t + \theta - x/v)$$
$$V_r(x,t) = \sin(\omega t + \theta + x/v - 2l/v)$$
$$V = V_f + V_r = 2\sin(\omega t)\cos\left(\frac{x-l}{v}\right)$$

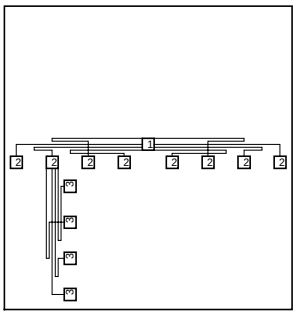


On-Chip Clock Distribution

- A large ASIC may have 10^4 to 10^5 clock loads
- Each is about 20fF
  - total load is 200pF to 2nF!
  - this load switches every cycle
- Need to
  - amplify a small clock signal to drive a 2nF load
  - distribute the clock over resistive wires to the amplifiers and loads
    - repeaters needed on long runs
  - avoid jitter
- Jitter comes from 3 main sources
  - differential supply variation modulates the delay of the clock buffers
  - single supply variation modulates the thresholds of the clock buffers
  - crosstalk to adjacent lines changes the delay of the clock fanout lines
- Each of these sources can be dealt with

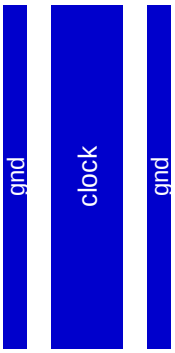
An On-Chip Clock Tree

- Distribute clock from center of chip
- Fanout first in X, then in Y
  - H-tree layout also possible
- Line lengths balanced at each stage of fanout
- Repeaters needed in long lines
- Wider than minimum width lines usually used
- Adjacent ground shields to eliminated crosstalk
- Differential distribution to avoid threshold shift
- Buffer sizing adjusted to match delay to varying spatial load



Reducing Jitter

- Address each of the three sources
- Differential Supply Noise
  - use a separate 'clock' power supply
  - adequately *bypass* this supply on chip
  - use insensitive buffers
- Single Supply Noise
  - use differential signaling
- Crosstalk
  - surround clock lines by parallel *shields*



Next Time

- Synchronization