Clock Distribution

Clocking

- Synchronous systems use a clock to keep operations in sequence
  - Distinguish *this* from *previous* or *next*
  - Determine speed at which machine operates

- Clock must be distributed to all the sequencing elements
  - Flip-flops and latches

- Also distribute clock to other elements
  - Domino circuits and memories
Clock Distribution

- On a small chip, the clock distribution network is just a wire
  - And possibly an inverter for clkb

- On practical chips, the RC delay of the wire resistance and gate load is very long
  - Variations in this delay cause clock to get to different elements at different times
  - This is called clock skew

- Most chips use repeaters to buffer the clock and equalize the delay
  - Reduces but doesn’t eliminate skew

Cycle Time Trends

- Much of CPU performance comes from higher $f$
  - $f$ is improving faster than simple process shrinks
  - Sequencing overhead is bigger part of cycle
  - Things just got different recently after multi-core came along

![Graphs showing cycle time trends over years](image)
Solutions

- **Reduce clock skew**
  - Careful clock distribution network design
  - Plenty of metal wiring resources

- **Analyze clock skew**
  - Only budget actual, not worst case skews
  - Local vs. global skew budgets

- **Tolerate clock skew**
  - Choose circuit structures insensitive to skew

Clock Dist. Networks

- **Ad hoc**
- **Grids**
- **H-tree**
- **Hybrid**
- **Spines**
- Can consume +30% total chip dynamic power
Clock Grids

- Use grid on two or more levels to carry clock
- Make wires wide to reduce RC delay
- Ensures low skew between nearby points
- But possibly large skew across die

Alpha Clock Grids
H-Trees

- Fractal structure
  - Gets clock arbitrarily close to any point
  - Matched delay along all paths
- Delay variations cause skew
- A and B might see big skew

Itanium 2 H-Tree

- Four levels of buffering:
  - Primary driver in the center
  - Four Repeater on the leaves of H
  - Second-level clock buffers
  - Local gaters
- Route around obstructions
## Hybrid Networks

- Use H-tree to distribute clock to many points
- Tie these points together with a grid
- Ex: IBM Power4, PowerPC
  - H-tree drives 16-64 sector buffers
  - Buffers drive total of 1024 points
  - All points shorted together with grid

## Skew Tolerance

- Flip-flops are sensitive to skew because of *hard edges*
  - Data launches at latest rising edge of clock
  - Must setup before earliest next rising edge of clock
  - Overhead would shrink if we can soften edge

- Latches tolerate moderate amounts of skew
  - Data can arrive anytime latch is transparent
## Summary

- Clock skew effectively increases setup and hold times in systems with hard edges

- Managing skew
  - Reduce: good clock distribution network
  - Analyze: local vs. global skew
  - Tolerate: use systems with soft edges

- Flip-flops and traditional domino are costly

- Latches and skew-tolerant domino perform at full speed even with moderate clock skews