21. Skews, Scaling

Jacob Abraham

Department of Electrical and Computer Engineering
The University of Texas at Austin

VLSI Design
Fall 2020

November 10, 2020

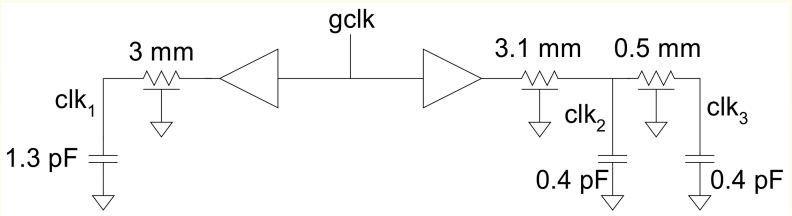
Clock Distribution

- 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

ECE Department, University of Texas at Austin Lecture 21. Skews, Scaling Jacob Abraham, November 10, 2020 1 / 48

Example

- Skew comes from differences in gate and wire delay
  - With right buffer sizing, clk_1 and clk_2 could ideally arrive at the same time
  - But power supply noise changes buffer delays
  - clk_2 and clk_3 will always see RC skew

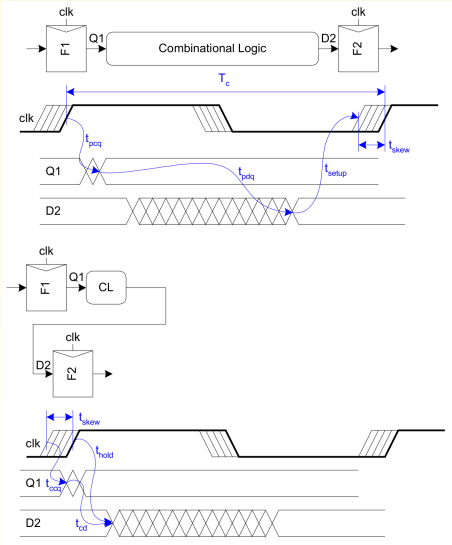


Review: Skew Impact

- Ideally full cycle is available for work
- Skew adds sequencing overhead
- Increases hold time too

$$t_{pd} \leq T_c - \underbrace{(t_{setup} + t_{pcq} + T_{skew})}_{\text{sequencing overhead}}$$

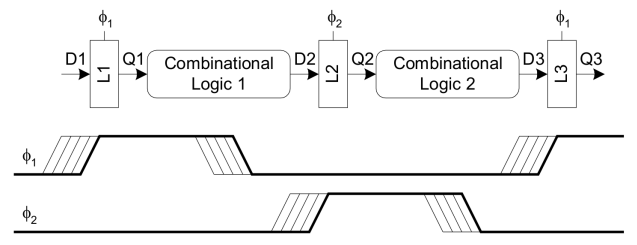
$$t_{cd} \geq t_{hold} - t_{ccq} + t_{skew}$$



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 any time latch is transparent

Skew: Latches



- 2-Phase Latches

$t_{pd} \leq T_c - \underbrace{(2t_{pdq})}_{\text{sequencing overhead}}$

$t_{cd1}, t_{cd2} \geq t_{hold} - t_{ccq} - t_{nonoverlap} + t_{skew}$

$t_{borrow} \leq T_c/2 - (t_{setup} + t_{nonoverlap} + t_{skew})$

- Pulsed Latches

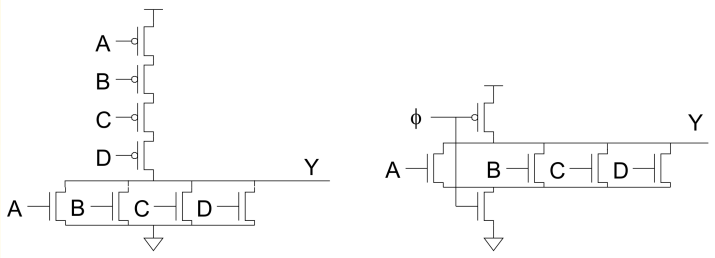
$t_{pd} \leq T_c - \underbrace{\max(t_{pdq}, t_{pcq} + t_{setup} - t_{pw} + t_{skew})}_{\text{sequencing overhead}}$

$t_{cd} \geq t_{hold} + t_{pw} - t_{ccq} + t_{skew}$

$t_{borrow} \leq t_{pw} - (t_{setup} + t_{skew})$

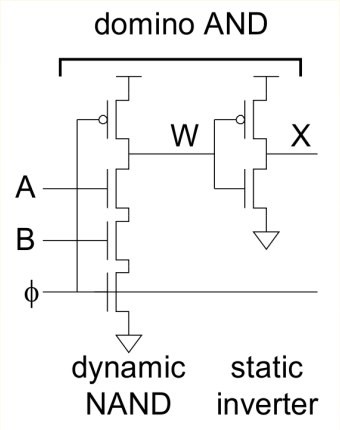
Dynamic Circuit Review

- Static circuits are slow because fat pMOS load input
- Dynamic gates use precharge to remove pMOS transistors from the inputs
  - Precharge: $\phi = 0$, output forced high
  - Evaluate: $\phi = 1$, output may pull low



Domino Circuits

- Dynamic inputs must monotonically rise during evaluation
  - Place inverting stage between each dynamic gate
  - Dynamic/static pair called domino gate
- Domino gates can be safely cascaded

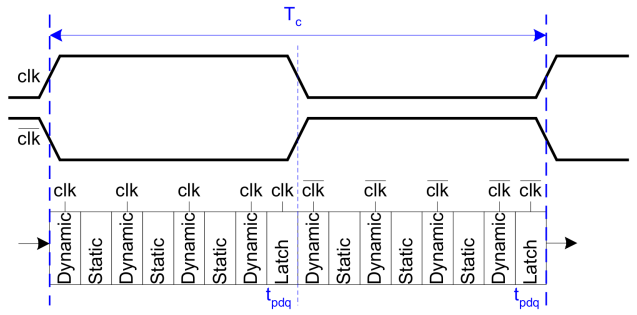


Domino Timing

- Domino gates are 1.5 – 2x faster than static CMOS
  - Lower logical effort because of reduced C_{in}
- Challenge is to keep precharge off critical path
- Look at clocking schemes for precharge and evaluate
  - Traditional schemes have severe overhead
  - Skew-tolerant domino hides this overhead

Traditional Domino Circuits

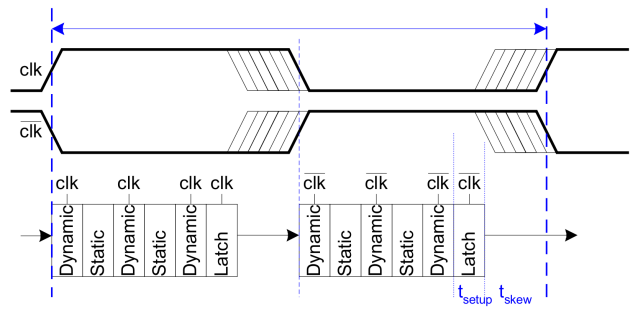
- Hide precharge time by ping-ponging between half-cycles
  - One evaluates while other precharges
  - Latches hold results during precharge



$$t_{pd} = T_c - 2t_{pdq}$$

Clock Skew

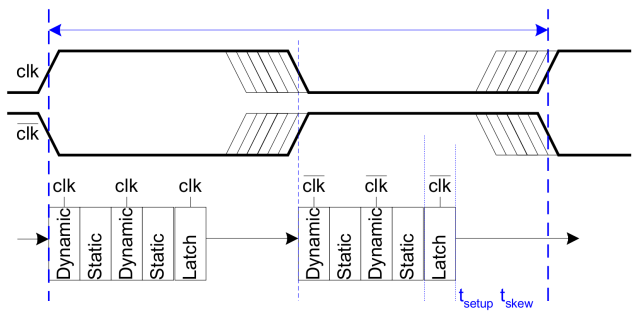
- Skew increases sequencing overhead
  - Traditional domino has hard edges
  - Evaluate at latest rising edge
  - Setup at latch by earliest falling edge



$$t_{pd} = T_c - 2t_{pdq} - 2t_{skew}$$

Time Borrowing

- Logic may not exactly fit half-cycle
  - No flexibility to borrow time to balance logic between half cycles
  - Traditional domino sequencing overhead is about 25% of cycle time in fast systems!

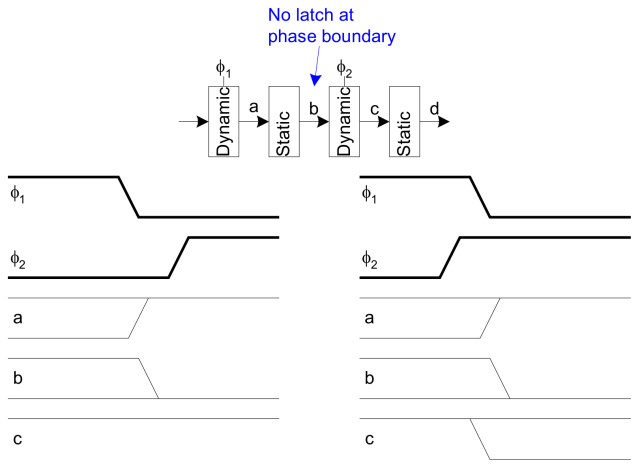


Relaxing the Timing

- Sequencing overhead caused by hard edges
  - Data departs dynamic gate on late rising edge
  - Must setup at latch on early falling edge
- Latch functions
  - Prevent glitches on inputs of domino gates
  - Holds results during precharge
- Is the latch really necessary?
  - No glitches if inputs come from other domino
  - Can we hold the results in another way?

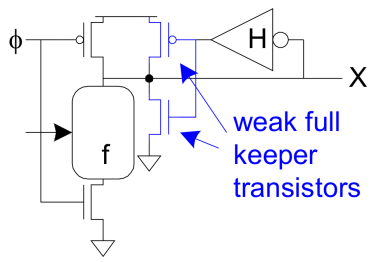
Skew-Tolerant Domino

- Use overlapping clocks to eliminate latches at phase boundaries
  - Second phase evaluates using results of first



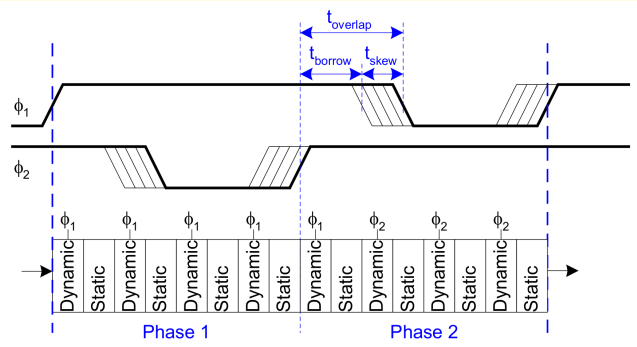
Full Keeper

- After second phase evaluates, first phase precharges
- Input to second phase falls
  - Violates monotonicity?
- But we no longer need the value
- Now the second gate has a floating output
  - Need full keeper to hold it either high or low



Time Borrowing

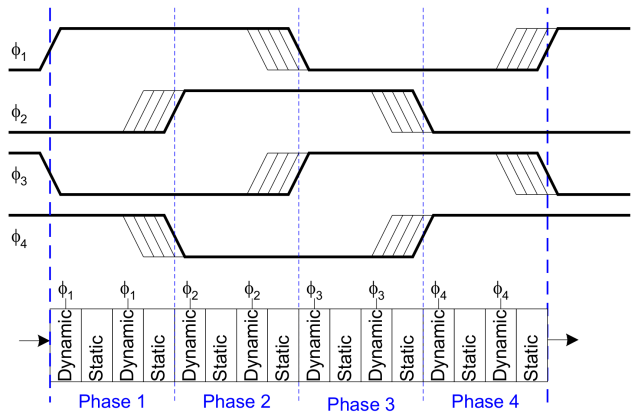
- Overlap can be used to
  - Tolerate clock skew
  - Permit time borrowing
- No sequencing overhead



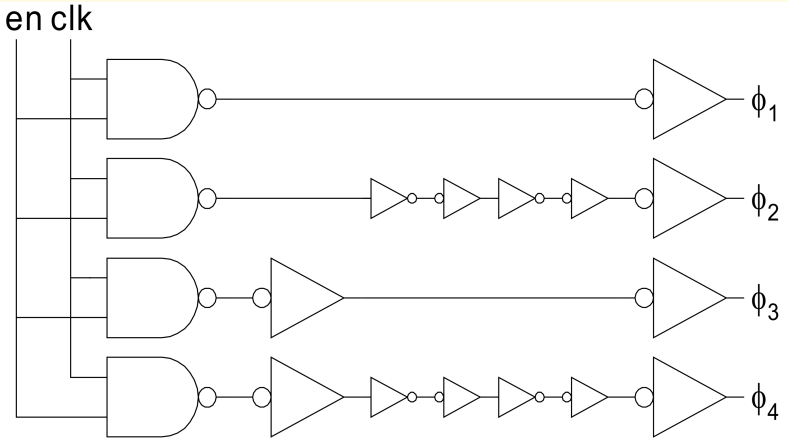
$$t_{pd} = T_c$$

Multiple Phases

- With more clock phases, each phase overlaps more
  - Permits more skew tolerance and time borrowing

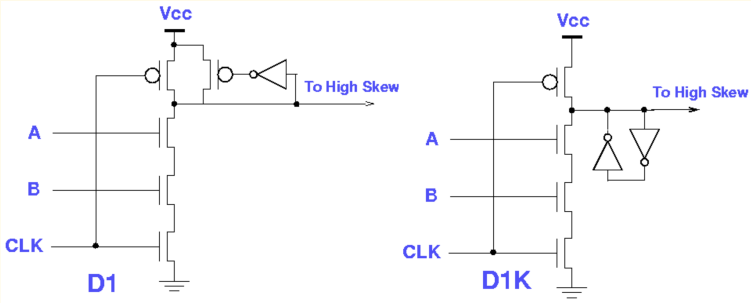


Clock Generation



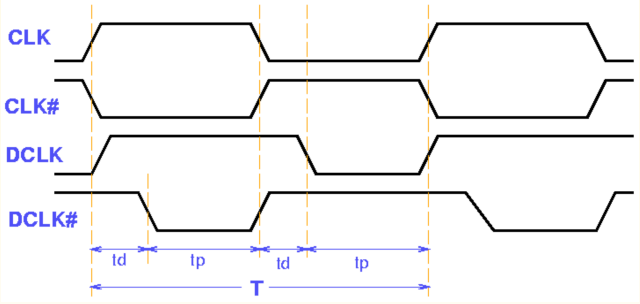
Opportunistic Time Borrowing

U. S. Patent no. 5517136 (Harris et al., May 14, 1996, assigned to Intel Corporation)
Pipelined domino logic allowing a slow stage to "borrow" from the time normally allocated to a faster stage

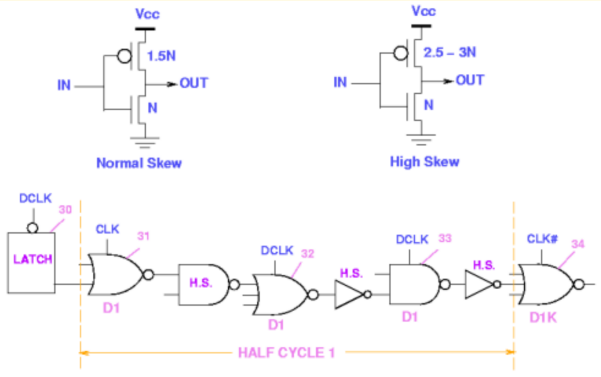


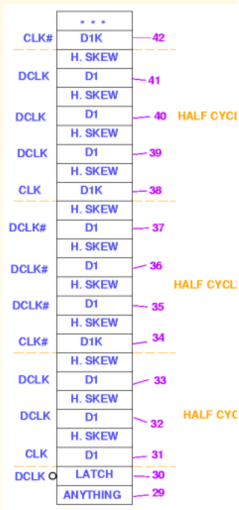
Clocking of Time-Borrowing Pipeline

- Delayed falling edges on clocks allow evaluation to continue into subsequent half cycle
  - Time delay t_d should be greater than of equal to the hold time of the domino logic gate plus any global clock skew
- Can generate the clocks by a local reference driven by the chip's global reference clock signal



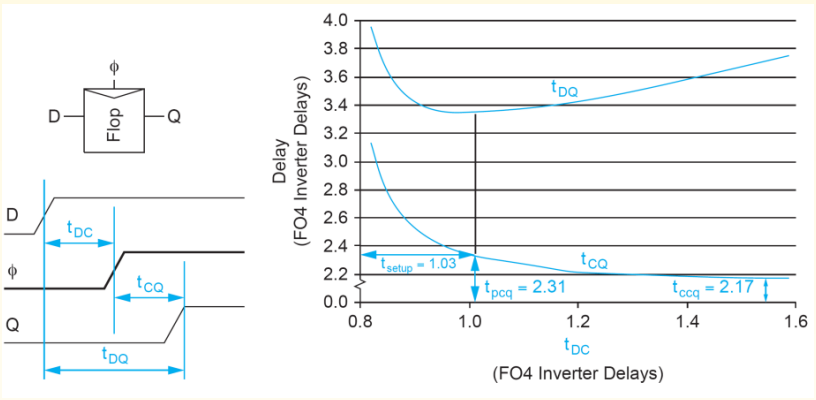
Example of an OTB Pipeline





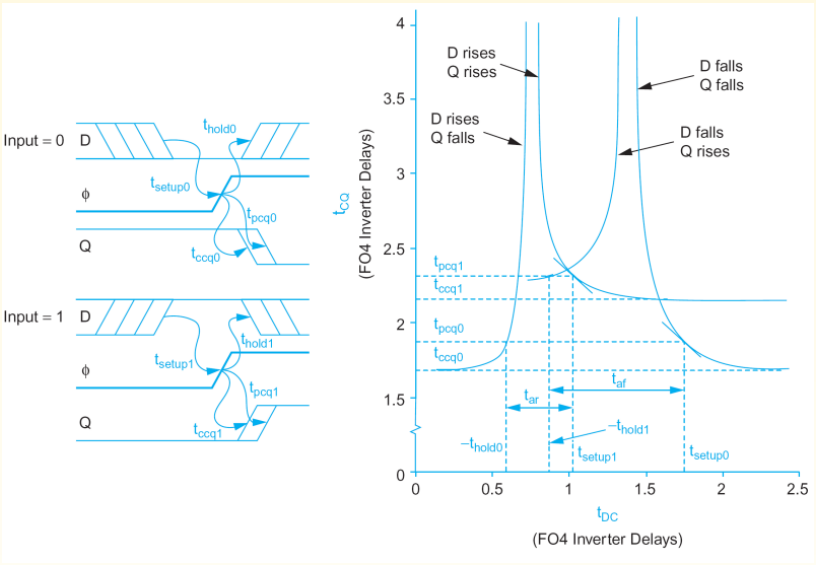
Half-cycles 1 and 3 evaluate when CLK is high, half-cycle 2 when CLK is low

Another Look into Flip-Flops and Clocking Delays

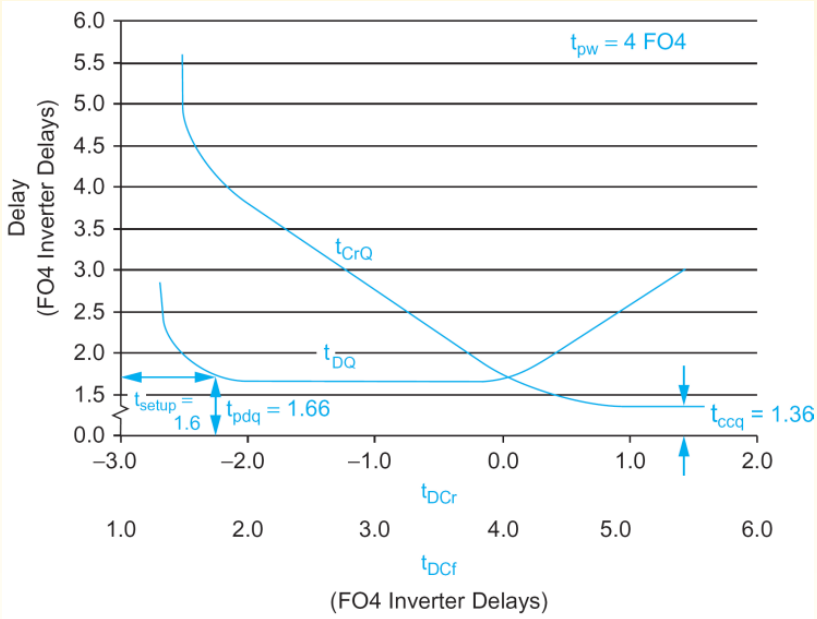


Flip-flop delay versus data arrival time

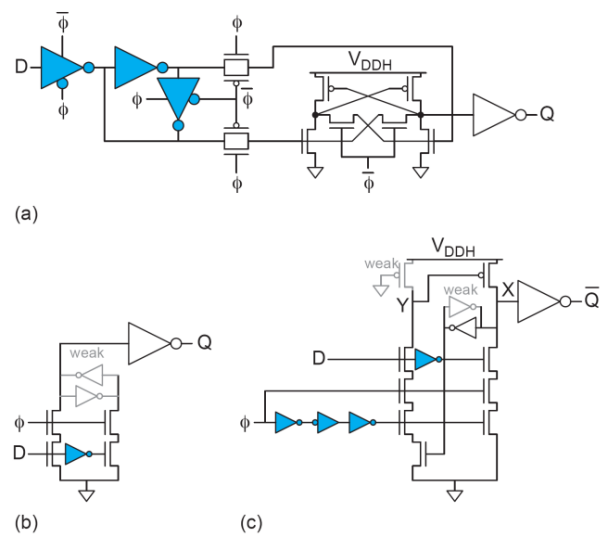
Flip-Flop Setup and Hold Times – Different Data Values



Latch Delay Versus Data Arrival Time

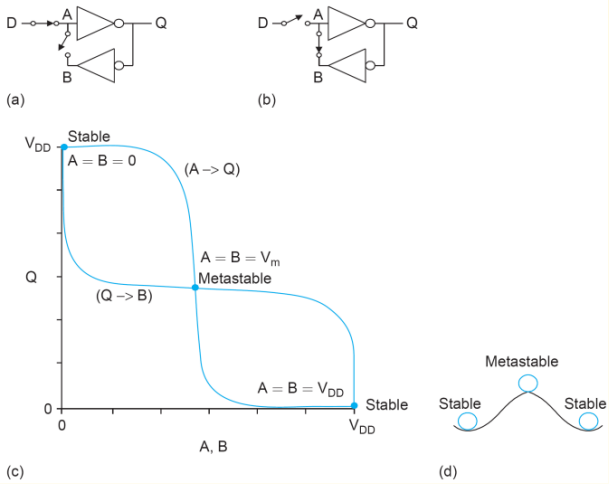


Level-Converter Flip-Flops and Latches



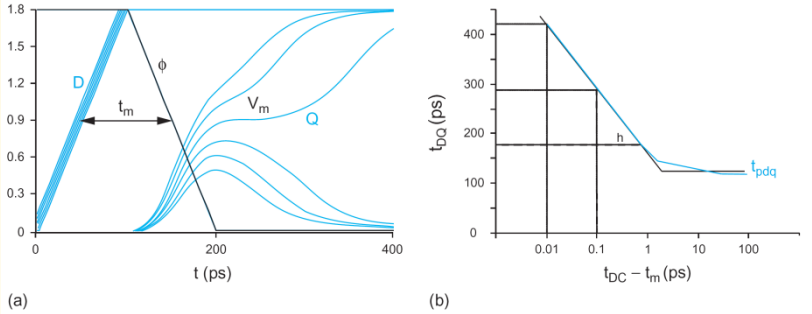
Blue Elements use V_{DDL}

Metastability

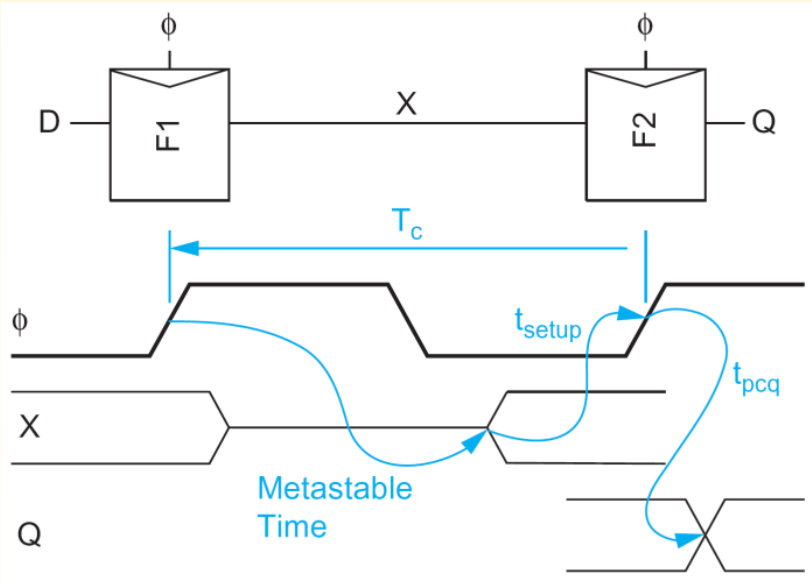


Metastable state in static latch

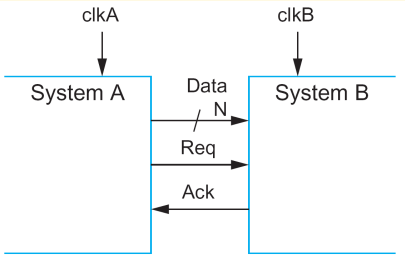
Metastable Transients and Propagation Delay

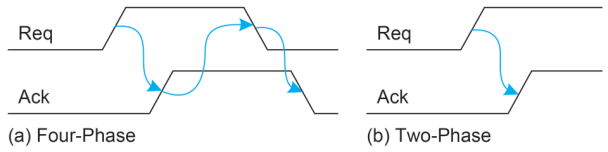


Simple Synchronizer



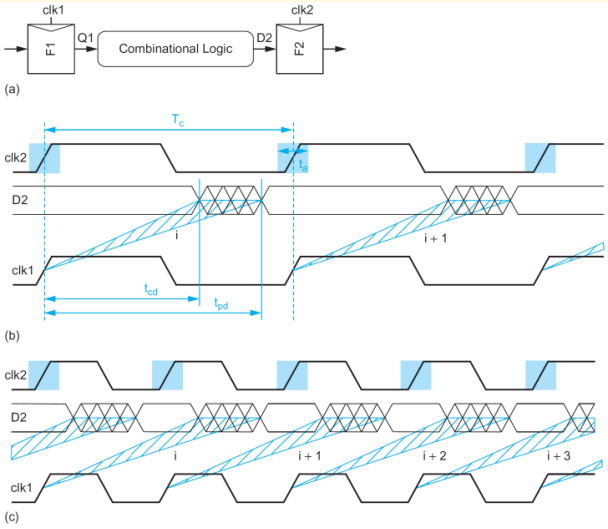
Asynchronous Systems – Communication





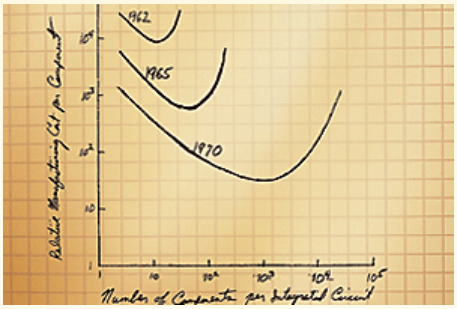
Handshake protocols

Wave Pipelining



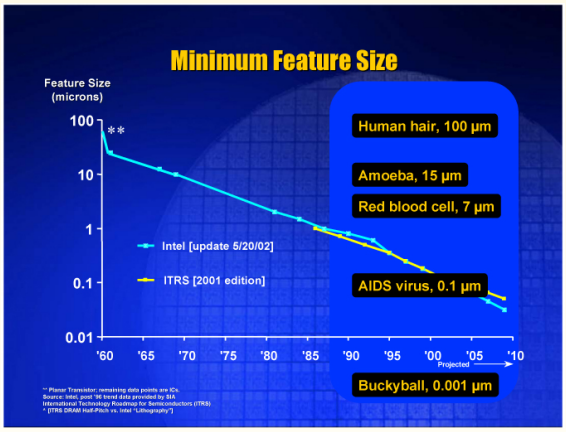
Chip Densities increase with Scaling

- In 1965, Gordon Moore predicted the exponential growth of the number of transistors on an IC (**Moore's Law**)
- Transistor count doubled every year since invention
- Predicted $>$ 65,000 transistors by 1975!
- Growth limited by power



Scaling

- The only constant in VLSI is constant change
- Feature size shrinks by 30% every 2-3 years
  - Transistors become cheaper, and faster
  - Wires do not improve (and may get worse)
- Scale factor S (typical technology nodes) $S = \sqrt{2}$



Scaling Assumptions

- What changes between technology nodes?
- Constant Field Scaling
  - All dimensions (x, y, z $\Longrightarrow$ W, L, t_{ox})
  - Voltage (V_{DD})
  - Doping levels
- Lateral Scaling
  - Only gate length L
  - Often done as a quick gate shrink (S = 1.05)

Device Scaling

Table 4.15 Influence of scaling on MOS device characteristics

Parameter	Sensitivity	Constant Field	Lateral
Scaling Parameters			
Length: L		$1/S$	$1/S$
Width: W		$1/S$	1
Gate oxide thickness: t_{ox}		$1/S$	1
Supply voltage: V_{DD}		$1/S$	1
Threshold voltage: V_{tn}, V_{tp}		$1/S$	1
Substrate doping: N_A		S	1
Device Characteristics			
β	$\frac{W}{L}\frac{1}{t_{ox}}$	S	S
Current: I_{ds}	$\beta(V_{DD}-V_t)^2$	$1/S$	S
Resistance: R	$\frac{V_{DD}}{I_{ds}}$	1	$1/S$
Gate capacitance: C	$\frac{WL}{t_{ox}}$	$1/S$	$1/S$
Gate delay: τ	RC	$1/S$	$1/S^2$
Clock frequency: f	$1/\tau$	S	S^2
Dynamic power dissipation (per gate): P	CV^2f	$1/S^2$	S
Chip area: A		$1/S^2$	1
Power density	P/A	1	S
Current density	I_{ds}/A	S	S

Observations

- Gate capacitance per micron is nearly independent of process
- But ON resistance $\times$ micron improves with process
- Gates get faster with scaling (good)
- Dynamic power goes down with scaling (good)
- Current density goes up with scaling (bad)
- Velocity saturation makes lateral scaling unsustainable

Solution

- Gate capacitance is typically about 2 fF/μm
- The FO4 inverter delay in the TT corner for a process of feature size f (in nm) is about 0.5f ps
- Estimate the ON resistance of a unit (4/2 λ) transistor
- FO4 = 5 τ = 15 RC
- RC = (0.5f)/15 = (f/30) ps/nm
- If W = 2f, R = 8.33 kΩ

Unit resistance is roughly independent of f

Scaling Assumptions

- Wire thickness
  - Hold constant vs. reduce in thickness
- Wire length
  - Local/scaled interconnect
  - Global interconnect
  - Die size scaled by $D_c \approx 1.1$

Interconnect Scaling

Table 4.16 Influence of scaling on interconnect characteristics

Parameter	Sensitivity	Reduced Thickness	Constant Thickness
Scaling Parameters			
Width: w		$1/S$	
Spacing: s		$1/S$	
Thickness: t		$1/S$	1
Interlayer oxide height: h		$1/S$	
Characteristics Per Unit Length			
Wire resistance per unit length: R_w	$\frac{1}{wt}$	S^2	S
Fringing capacitance per unit length: C_{wf}	$\frac{t}{s}$	1	S
Parallel plate capacitance per unit length: C_{wp}	$\frac{w}{h}$	1	1
Total wire capacitance per unit length: C_w	$C_{wf} + C_{wp}$	1	between 1, S
Unrepeated RC constant per unit length: t_{wu}	$R_w C_w$	S^2	between S, S^2
Repeated wire RC delay per unit length: t_{wr} (assuming constant field scaling of gates in Table 4.15)	$\sqrt{RCR_wC_w}$	$\sqrt{S}$	between 1, $\sqrt{S}$
Crosstalk noise	$\frac{t}{s}$	1	S

Interconnect Delay

Table 4.16 Influence of scaling on interconnect characteristics

Parameter	Sensitivity	Reduced Thickness	Constant Thickness
Scaling Parameters			
Width: w		$1/S$	
Spacing: s		$1/S$	
Thickness: t		$1/S$	1
Interlayer oxide height: h		$1/S$	
Local/Scaled Interconnect Characteristics			
Length: l		$1/S$	
Unrepeated wire RC delay	$l^2 t_{wu}$	1	between $1/S$, 1
Repeated wire delay	$l t_{wr}$	$\sqrt{1/S}$	between $1/S$, $\sqrt{1/S}$
Global Interconnect Characteristics			
Length: l		D_c	
Unrepeated wire RC delay	$l^2 t_{wu}$	$S^2 D_c^2$	between SD_c^2, $S^2 D_c^2$
Repeated wire delay	$l t_{wr}$	$D_c\sqrt{S}$	between D_c, $D_c\sqrt{S}$

Observations

- Capacitance per micron is remaining constant
  - About 0.2 fF/μm
  - Roughly 1/10 of gate capacitance
- Local wires are getting faster
  - Not quite tracking transistor improvement
  - But not a major problem
- Global wires are getting slower
  - No longer possible to cross chip in one cycle

Previously: International Technology Roadmap for Semiconductors (ITRS)

Table 4.17 Predictions from the 2002 ITRS

Year	2001	2004	2007	2010	2013	2016
Feature size (nm)	130	90	65	45	32	22
V_{DD} (V)	1.1–1.2	1–1.2	0.7–1.1	0.6–1.0	0.5–0.9	0.4–0.9
Millions of transistors/die	193	385	773	1564	3092	6184
Wiring levels	8–10	9–13	10–14	10–14	11–15	11–15
Intermediate wire pitch (nm)	450	275	195	135	95	65
Interconnect dielectric constant	3–3.6	2.6–3.1	2.3–2.7	2.1	1.9	1.8
I/O signals	1024	1024	1024	1280	1408	1472
Clock rate (MHz)	1684	3990	6739	11511	19348	28751
FO4 delays/cycle	13.7	8.4	6.8	5.8	4.8	4.7
Maximum power (W)	130	160	190	218	251	288
DRAM capacity (Gbits)	0.5	1	4	8	32	64

Now: International Roadmap for Devices and Systems

Technology Roadmaps

- Many steps needed to produce an IC
- Each step requires specialized (and expensive) equipment produced by different vendors
- Roadmaps give equipment manufacturers an idea what equipment would be used, and when the capability would be needed (example, scaling factor of $sqrt(2)$)

ITRS established in 2013

- Scaling projections to 2028
- With Moore's law coming to an end, the final roadmap was issued in 2016
- Through IEEE's Rebooting Computing initiative, the IRDS was started

Some of the Focus Team Topics in the International Roadmap for Devices and Systems

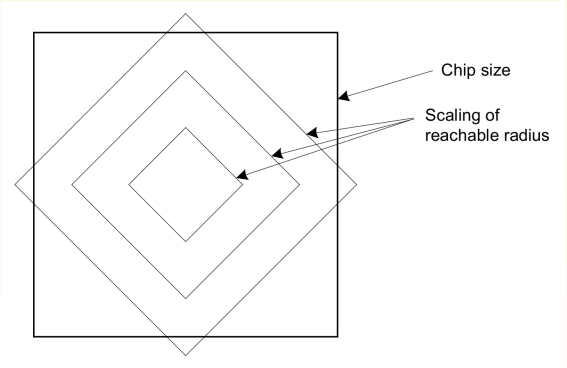
Application Benchmarking
Systems and Architectures
More Moore
Beyond CMOS
Packaging Integration
Outside System Connectivity
Factory Integration
Lithography
Metrology
Emerging Research Materials
Environment, Safety, Health, and Sustainability
Yield Enhancement
Cryogenic Electronics and Quantum Information Processing (added in 2018)

Scaling Implications

- Improved Performance
- Improved Cost
- Interconnect Woes
- Power Woes
- Productivity Challenges
- Physical Limits

Reachable Radius

- We can't send a signal across a large fast chip in one cycle anymore
- But the microarchitect can plan around this
  - Just as off-chip memory latencies were tolerated



Globally Asynchronous, Locally Synchronous (GALS)

VLSI Economics

- Selling price S_{total}
  - $S_{total} = C_{total}/(1 - m)$
- m = profit margin
- C_{total} = total cost
  - Nonrecurring engineering cost (NRE)
  - Recurring cost
  - Fixed cost

NRE

- Engineering cost
  - Depends on size of design team
  - Include benefits, training, computers
  - CAD tools:
    - Digital front end: \$10K
    - Analog front end: \$100K
    - Digital back end: \$1M
- Prototype manufacturing
  - Mask costs: \$500K – 1M in 130 nm process
  - Test fixture and package tooling

Recurring and Fixed Costs

Recurring costs

- Fabrication
  - Wafer cost/(Dice per wafer × Yield)
  - Wafer cost: \$500 - \$3000
  - Dice per wafer:

$$N = \pi \left[\frac{r^2}{A} - \frac{2r}{\sqrt{2A}} \right]$$

  - Yield: $Y = e^{-AD}$
    - For small A, Y ≈ 1, cost proportional to area
    - For large A, Y → 0, cost increases exponentially
- Packaging
- Test

Fixed costs

- Data sheets and application notes
- Marketing and advertising
- Yield analysis

Example

- You want to start a company to build a wireless communications chip. How much venture capital must you raise?
- Because you are smarter than everyone else, you can get away with a small team in just two years:
  - Seven digital designers
  - Three analog designers
  - Five support personnel

Solution

- Digital designers
  - \$70k salary
  - \$30k overhead
  - \$10k computer
  - \$10k CAD tools
  - Total: \$120k $\times$ 7 = \$840k
- Analog designers
  - \$100k salary
  - \$30k overhead
  - \$10k computer
  - \$100k CAD tools
  - Total: \$240k $\times$ 3 = \$720k
- Support staff
  - \$45k salary
  - \$20k overhead
  - \$5k computer
  - Total: \$70k $\times$ 5 = \$350k
- Fabrication
  - Back-end tools: \$1M
  - Masks: \$1M
  - Total: \$2M/year
- **Summary:**
  - 2 years at \$3.91M/year
  - \$8M design and prototype

Cost Breakdown

- New chip design is fairly capital-intensive
- Maybe you can do it for less?