Administrativia

- Project Phase 2 now on the web-site.
- Hw 6 due today.
- New homework to be posted in a week.
- Cory Hall closed on Monday (Power Outage)
  - Instructional computers in 353 Cory should come back on line on Tu.
- Enjoy Spring Break!
Class Material

- Last lecture
  - Registers
- Today’s lecture
  - Timing
- Reading (Ch 10)
**Latch versus Register (Flip-flop)**

- **Latch:** level-sensitive
  - clock is low - hold mode
  - clock is high - transparent

- **Register:** edge-triggered
  - stores data when clock rises

---

**Synchronous Timing**

Schematic of a synchronous circuit with a clock input (CLK), an input (In), a combinational logic block, and an output (Out). The combinational logic block processes the input and produces an output that is delayed by a single clock cycle. The circuit ensures that the output is synchronized with the clock edges, maintaining a consistent timing relationship between the inputs and outputs.
Delays can be different for rising and falling data transitions.

Latch Parameters

Delays can be different for rising and falling data transitions.

Register Parameters

Delays can be different for rising and falling data transitions.
Timing Constraints

\[ t_{\text{clk-q}} \]
\[ t_{\text{clk-q,min}} \]
\[ t_{\text{setup}}, \ t_{\text{hold}} \]
\[ t_{\text{logic}} \]
\[ t_{\text{logic,min}} \]

Cycle time (max): \( T_{\text{Clk}} > t_{\text{clk-q}} + t_{\text{logic}} + t_{\text{setup}} \)

Race margin (min): \( t_{\text{hold}} < t_{\text{clk-q,min}} + t_{\text{logic,min}} \)
Clock Nonidealities

- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, $t_{SK}$

- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) $t_{JS}$
  - Long term $t_{JL}$

- **Variation of the pulse width**
  - Important for level sensitive clocking

Clock Uncertainties

Sources of clock uncertainty
Both skew and jitter affect the effective cycle time and the race margin.
Positive and Negative Skew

(a) Positive skew

(b) Negative skew

Positive Skew

Launching edge arrives before the receiving edge
**Negative Skew**

Receiving edge arrives before the launching edge

Minimum cycle time:
\[ T_{\text{clk}} + \delta = t_{\text{clk-q}} + t_{\text{setup}} + t_{\text{logic}} \]

Worst case is when receiving edge arrives early (negative \( \delta \))
**Timing Constraints**

- **In** → **R1** → **Combinational Logic** → **R2** → **DQ**

- **CLK** leads to **t_{CLK1}** and **t_{CLK2}**

- **Hold time constraint:**
  \[ t_{(clk-q,min)} + t_{(logic,min)} > t_{hold} + \delta \]

- Worst case is when receiving edge arrives late

**Longest Logic Path in Edge-Triggered Systems**

- **Clk**
  - Latest point of launching
  - Earliest arrival of next cycle
**Clock Constraints in Edge-Triggered Systems**

If launching edge is late and receiving edge is early, the data will not be too late if:

\[ t_{clk-q} + t_{\text{logic}} + t_{\text{setup}} < T_{CLK} - t_{JS,1} - t_{JS,2} + \delta \]

Minimum cycle time is determined by the maximum delays through the logic

\[ t_{clk-q} + t_{\text{logic}} + t_{\text{setup}} - \delta + 2t_{JS} < T_{CLK} \]

Skew can be either positive or negative

**Shortest Path**

Earliest point of launching

Clk

\[ t_{\text{logic,min}} \]

\[ t_{clk-q,min} \]

Clk

\[ t_{\text{hold}} \]

Nominal clock edge

Data must not arrive before this time
**Clock Constraints in Edge-Triggered Systems**

If launching edge is early and receiving edge is late:

\[ t_{\text{clk-q,min}} + t_{\text{logic,min}} - t_{JS,1} > t_{\text{hold}} + t_{JS,2} + \delta \]

Minimum logic delay

\[ t_{\text{clk-q,min}} + t_{\text{logic,min}} > t_{\text{hold}} + 2t_{JS} + \delta \]

(This assumes jitter at launching and receiving clocks are independent – which usually is not true)

---

**Datapath with Feedback**

![Datapath Diagram]

- **Negative skew**
- **Positive skew**
- **Clock distribution**

EECS141  Lecture #16  23

EECS141  Lecture #16  24
Pipelining

Reference

<table>
<thead>
<tr>
<th>Clock Period</th>
<th>Adder Value</th>
<th>Absolute Value</th>
<th>Logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a_1 + b_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$a_2 + b_2$</td>
<td>$</td>
<td>a_1 + b_1</td>
</tr>
<tr>
<td>3</td>
<td>$a_3 + b_3$</td>
<td>$(a_2 + b_2)$</td>
<td>$\log(</td>
</tr>
<tr>
<td>4</td>
<td>$a_4 + b_4$</td>
<td>$(a_3 + b_3)$</td>
<td>$\log(</td>
</tr>
<tr>
<td>5</td>
<td>$a_5 + b_5$</td>
<td>$(a_4 + b_4)$</td>
<td>$\log(</td>
</tr>
</tbody>
</table>

Pipelined

EECS141 Lecture #16

Latch-Based Clocking

(Domino logic almost always uses latch-based clocking)

EECS141 Lecture #16
**Latch vs. Flip-flop**

- **In a flip-flop based system:**
  - Data launches on one rising edge
    - And must arrive before next rising edge
  - If data arrives late, system fails
    - If it arrives early, wasting time
  - Flip-flops have hard edges

- **In a latch-based system:**
  - Data can pass through latch while it is transparent
  - Long cycle of logic can borrow time into next cycle
    - As long as each loop finished in one cycle

---

**Time Borrowing Example**

![Diagram showing time borrowing example]
**Latch vs. Flip-flop Summary**

- Flip-flops generally easier to use
  - Most digital ASICs designed with register-based timing
- But, latches (both pulsed and level-sensitive) allow more flexibility
  - And hence can potentially achieve higher performance
  - Latches can also be made more tolerant of clock un-certainty
  - More in EE241

---

**CMOS Transistor Scaling**
Goals of Technology Scaling

- Make things cheaper:
  - Want to sell more functions (transistors) per chip for the same money
  - Or build same products cheaper
  - Price of a transistor has to be reduced
- But also want to be faster, smaller, lower power…

Technology Scaling

  - Double transistor density
  - Reduce gate delay by 30% (increase operating frequency by 43%)
  - Reduce energy per transition by 65% (50% power savings @ 43% increase in frequency)
- Die size used to increase by 14% per generation (not any more)
- Technology generation spans 2-3 years
Moore was not always accurate

Projected 2000 Wafer, circa 1975

2X reduction every ~5 years

Technology Scaling (1)

Minimum Feature Size

2X reduction every ~5 years
Technology Scaling (2)

Number of components per chip

Technology Scaling (3)

Propagation Delay

t_c decreases by 30%/year
f increases by 43%
Technology Scaling Models

- **Full Scaling (Constant Electrical Field)**
  
  ideal model — dimensions and voltages scale together by the same factor S

- **Fixed Voltage Scaling**
  
  most common model until 1990's
  
  only dimensions scale, voltages remain constant

- **General Scaling**
  
  most realistic for today's situation — voltages and dimensions scale with different factors
Scaling

- $W, L, t_{ox}: 1/S$
- $V_{DD}, V_T: 1/S$
- Area: $WL$
- $C_{ox}: 1/tox$
- $C_L: C_{ox}WL$
- $I_D: C_{ox}(W/L)(V_{DD}-V_T)^2$
- $R_{eq}: V_{DD}/I_{DSAT}$

Full Scaling (Dennard, Long-Channel)
**Full Scaling (Dennard, Long-Channel)**

- $W, L, t_{\text{ox}}$: $1/S$
- $V_{\text{DD}}, V_T$: $1/S$

- $t_p$: $R_{\text{eq}}C_L$
- $P_{\text{avg}}$: $C_LV_{\text{DD}}^2/t_p$
- $P_{\text{avg}}/A$: $C_{\text{ox}}V_{\text{DD}}^2/t_p$

---

**Scaling Relationships for Long Channel Devices**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Relation</th>
<th>Full Scaling</th>
<th>General Scaling</th>
<th>Fixed Voltage Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W, L, t_{\text{ox}}$</td>
<td>1/$S$</td>
<td>1/$S$</td>
<td>1/$S$</td>
<td></td>
</tr>
<tr>
<td>$V_{\text{DD}}, V_T$</td>
<td>1/$S$</td>
<td>1/$U$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$N_{\text{SUB}}$</td>
<td>$V/W_{\text{dep}}^2$</td>
<td>$S$</td>
<td>$S^2/U$</td>
<td>$S^2$</td>
</tr>
<tr>
<td>Area/Device</td>
<td>$WL$</td>
<td>1/$S^2$</td>
<td>1/$S^2$</td>
<td>1/$S^2$</td>
</tr>
<tr>
<td>$C_{\text{ax}}$</td>
<td>$1/t_{\text{ox}}$</td>
<td>$S$</td>
<td>$S$</td>
<td>$S$</td>
</tr>
<tr>
<td>$C_L$</td>
<td>$C_{\text{ox}}WL$</td>
<td>1/$S$</td>
<td>1/$S$</td>
<td>1/$S$</td>
</tr>
<tr>
<td>$k_{\text{m}}, k_p$</td>
<td>$C_{\text{ox}}WL$</td>
<td>$S$</td>
<td>$S$</td>
<td>$S$</td>
</tr>
<tr>
<td>$I_{\text{av}}$</td>
<td>$k_{n,p}V^2$</td>
<td>1/$S$</td>
<td>$S/U^2$</td>
<td>$S$</td>
</tr>
<tr>
<td>$t_p$ (intrinsic)</td>
<td>$C_LV/I_{\text{av}}$</td>
<td>$1/S$</td>
<td>$U/S^2$</td>
<td>1/$S^2$</td>
</tr>
<tr>
<td>$P_{\text{av}}$</td>
<td>$C_LV^2/t_p$</td>
<td>1/$S^3$</td>
<td>$S/U^3$</td>
<td>$S$</td>
</tr>
<tr>
<td>PDP</td>
<td>$C_LV^2$</td>
<td>1/$S^3$</td>
<td>1/$SU^2$</td>
<td>1/$S$</td>
</tr>
</tbody>
</table>
**Full Scaling (Dennard, Short-Channel)**

- \( W, L, t_{ox}: 1/S \)
- \( V_{DD}, V_T: 1/S \)

- Area: WL
- \( C_{ox}: 1/tox \)
- \( C_L: C_{ox}WL \)
- \( I_D: WC_{ox}v_{sat}(V_{DD}-V_T-V_{SAT}/2) \)
- \( R_{eq}: V_{DD}/I_{DSAT} \)

---

**Full Scaling (Dennard, Short-Channel)**

- \( W, L, t_{ox}: 1/S \)
- \( V_{DD}, V_T: 1/S \)

- \( t_p: R_{eq}C_L \)
- \( P_{avg}: C_L V_{DD}^2/t_p \)
- \( P_{avg}/A: C_{ox} V_{DD}^2/t_p \)
Transistor Scaling
(Velocity-Saturated Devices)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Relation</th>
<th>Full Scaling</th>
<th>General Scaling</th>
<th>Fixed Voltage Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>W, L, Lx</td>
<td>1/S</td>
<td>1/S</td>
<td>1/S</td>
<td></td>
</tr>
<tr>
<td>VDD, VRef</td>
<td>1/S</td>
<td>1/S</td>
<td>1/S</td>
<td></td>
</tr>
<tr>
<td>Ntub</td>
<td>V/2W_mol</td>
<td>S</td>
<td>S^2/4U</td>
<td>S^2</td>
</tr>
<tr>
<td>ActiveDevice</td>
<td>W2</td>
<td>1/S^3</td>
<td>1/S^3</td>
<td>1/S^3</td>
</tr>
<tr>
<td>C_m</td>
<td>1/td_m</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>C_pim</td>
<td>C_p, WL</td>
<td>1/S</td>
<td>1/S</td>
<td>1/S</td>
</tr>
<tr>
<td>k_p, k_m</td>
<td>C_p, WL</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Iout</td>
<td>C_m, WV</td>
<td>1/S</td>
<td>1/U</td>
<td>1</td>
</tr>
<tr>
<td>Current Density</td>
<td>I_d/Area</td>
<td>S</td>
<td>S^2/4U</td>
<td>S^2</td>
</tr>
<tr>
<td>f</td>
<td>VD, n</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Intrinsic Delay</td>
<td>R_d, C_m</td>
<td>1/S</td>
<td>1/S</td>
<td>1/S</td>
</tr>
<tr>
<td>P</td>
<td>I_d, V</td>
<td>1/S^3</td>
<td>1/U^2</td>
<td>1</td>
</tr>
</tbody>
</table>

An interesting question

- What will did cause this model to break?
  - Leakage set by kT/q
    - Temp. does not scale
    - V_T set to minimize power
  - Power actually increased
    - Leakage increased drastically
    - f increased faster than device speed
    - Hit cooling limit
  - Process Variation
    - Hard to build very small things accurately (less averaging)