Outline

- Sequencing
- Sequencing Element Design
- Max and Min-Delay
- Clock Skew
- Time Borrowing
- Two-Phase Clocking
Sequencing

- **Combinational logic**
  - output depends on current inputs

- **Sequential logic**
  - output depends on current and previous inputs
  - Requires separating previous, current, future
  - Called *state* or *tokens*
  - Ex: FSM, pipeline

Finite State Machine

Pipeline

10: Sequential Circuits  CMOS VLSI Design  Slide 3
If tokens moved through pipeline at constant speed, no sequencing elements would be necessary.

Ex: fiber-optic cable
- Light pulses (tokens) are sent down cable
- Next pulse sent before first reaches end of cable
- No need for hardware to separate pulses
- But dispersion sets min time between pulses

This is called wave pipelining in circuits.

In most circuits, dispersion is high
- Delay fast tokens so they don’t catch slow ones.
Sequencing Overhead

- Use flip-flops to delay fast tokens so they move through exactly one stage each cycle.
- Inevitably adds some delay to the slow tokens
- Makes circuit slower than just the logic delay
  - Called sequencing overhead
- Some people call this clocking overhead
  - But it applies to asynchronous circuits too
  - Inevitable side effect of maintaining sequence
Sequencing Elements

- **Latch**: Level sensitive
  - a.k.a. transparent latch, D latch

- **Flip-flop**: edge triggered
  - A.k.a. master-slave flip-flop, D flip-flop, D register

- **Timing Diagrams**
  - Transparent
  - Opaque
  - Edge-trigger
Sequencing Elements

- **Latch**: Level sensitive
  - a.k.a. transparent latch, D latch

- **Flip-flop**: edge triggered
  - A.k.a. master-slave flip-flop, D flip-flop, D register

- **Timing Diagrams**
  - Transparent
  - Opaque
  - Edge-trigger
Latch Design

- Pass Transistor Latch
- Pros
  - +
  - +
- Cons
  - –
  - –
  - –
  - –
  - –
Latch Design

- Pass Transistor Latch
  - Pros
    + Tiny
    + Low clock load
  - Cons
    - $V_t$ drop
    - nonrestoring
    - backdriving
    - output noise sensitivity
    - dynamic
    - diffusion input

Used in 1970’s
Latch Design

- Transmission gate

\[
\begin{align*}
\text{D} & \quad \text{Q} \\
\phi & \\
\end{align*}
\]
Latch Design

- Transmission gate
  - No $V_t$ drop
  - Requires inverted clock

\[ \text{Diagram of latch} \]
Latch Design

- Inverting buffer
  - +
  - +
  - + Fixes either
    - •
    - •
    - –
Latch Design

- Inverting buffer
  - Restoring
  - No backdriving
  - Fixes either
    - Output noise sensitivity
    - Or diffusion input
  - Inverted output
Latch Design

- Tristate feedback

[Diagram of latch design with symbols and logic gates]
Latch Design

- Tristate feedback
  + Static
  - Backdriving risk

- Static latches are now essential
Latch Design

- Buffered input

![Diagram of a latch circuit]
Latch Design

- Buffered input
  - Fixes diffusion input
  - Noninverting
Latch Design

- Buffered output

Diagram:

```
   +
   D  X  Q
   φ  φ  φ
```

CMOS VLSI Design 10: Sequential Circuits
Latch Design

- Buffered output
  + No backdriving

- Widely used in standard cells
  + Very robust (most important)
    - Rather large
  - Rather slow (1.5 – 2 FO4 delays)
  - High clock loading
Latch Design

- Datapath latch
  - D
  - X
  - Q
  - \overline{Q}
Latch Design

- Datapath latch
  - Smaller, faster
  - Unbuffered input
Flip-Flop Design

Flip-flop is built as pair of back-to-back latches

\[ \phi \]
Enable

- Enable: ignore clock when \( \text{en} = 0 \)
- Mux: increase latch D-Q delay
- Clock Gating: increase \( \text{en} \) setup time, skew
Reset

- Force output low when reset asserted
- Synchronous vs. asynchronous

Synchronous Reset

Asynchronous Reset

10: Sequential Circuits  CMOS VLSI Design  Slide 24
Set / Reset

- Set forces output high when enabled

- Flip-flop with asynchronous set and reset
Sequencing Methods

- Flip-flops
- 2-Phase Latches
- Pulsed Latches

![Diagram of sequencing methods]

10: Sequential Circuits
CMOS VLSI Design
Slide 26
Timing Diagrams

Contamination and Propagation Delays

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{pd}$</td>
<td>Logic Prop. Delay</td>
</tr>
<tr>
<td>$t_{cd}$</td>
<td>Logic Cont. Delay</td>
</tr>
<tr>
<td>$t_{pcq}$</td>
<td>Latch/Flop Clk-Q Prop Delay</td>
</tr>
<tr>
<td>$t_{ccq}$</td>
<td>Latch/Flop Clk-Q Cont. Delay</td>
</tr>
<tr>
<td>$t_{pdq}$</td>
<td>Latch D-Q Prop Delay</td>
</tr>
<tr>
<td>$t_{pcq}$</td>
<td>Latch D-Q Cont. Delay</td>
</tr>
<tr>
<td>$t_{setup}$</td>
<td>Latch/Flop Setup Time</td>
</tr>
<tr>
<td>$t_{hold}$</td>
<td>Latch/Flop Hold Time</td>
</tr>
</tbody>
</table>

Combinational Logic

A → Y

Latch

D → Q

Clk

D → Q

Clk
Max-Delay: Flip-Flops

\[ t_{pd} \leq T_c \left( \frac{\text{sequencing overhead}}{2} \right) \]

Combinational Logic

\[ Q_1 \quad T_c \quad D_2 \]

\[ \text{clk} \quad t_{pcq} \quad t_{pd} \quad t_{setup} \]

10: Sequential Circuits
CMOS VLSI Design
Slide 28
Max-Delay: Flip-Flops

\[ t_{pd} \leq T_c - \left( t_{\text{setup}} + t_{\text{pcq}} \right) \]

sequencing overhead

10: Sequential Circuits  CMOS VLSI Design  Slide 29
Max Delay: 2-Phase Latches

\[ t_{pd} = t_{pd1} + t_{pd2} \leq T_c - \left( \text{sequencing overhead} \right) \]

\[
Q1 \xrightarrow{\phi_1} \text{Combinational Logic 1} \xrightarrow{\phi_2} Q2 \xrightarrow{\phi_1} \text{Combinational Logic 2} \]

sequencing overhead
Max Delay: 2-Phase Latches

\[ t_{pd} = t_{pd1} + t_{pd2} \leq T_c - \left(2t_{pq}\right) \]

sequencing overhead

\[ \phi_1 \quad \phi_2 \quad \phi_1 \]

D1 \( \rightarrow \) Q1 \( \rightarrow \) Combinational Logic 1

D2 \( \rightarrow \) Q2 \( \rightarrow \) Combinational Logic 2

D3 \( \rightarrow \) Q3

\[ t_{pd1} + t_{pd2} \leq T_c - \left(2t_{pq}\right) \]

sequencing overhead
Max Delay: Pulsed Latches

\[ t_{pd} \leq T_c - \max \left( \text{sequencing overhead} \right) \]

(a) \( t_{pw} > t_{setup} \)

(b) \( t_{pw} < t_{setup} \)
Min-Delay: Flip-Flops

\[ t_{cd} \geq \]

\( F_1 \) \( Q_1 \) \( CL \) \( F_2 \) \( D_2 \)

\( t_{cd} \) \( t_{ccq} \) \( t_{hold} \)
Min-Delay: Flip-Flops

\[ t_{cd} \geq t_{\text{hold}} - t_{ccq} \]
\[ t_{cd1}, t_{cd2} \geq \]

Hold time reduced by nonoverlap

Paradox: hold applies twice each cycle, vs. only once for flops.

But a flop is made of two latches!
Min-Delay: 2-Phase Latches

\[ t_{cd1}, t_{cd2} \geq t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}} \]

Hold time reduced by nonoverlap

Paradox: hold applies twice each cycle, vs. only once for flops.

But a flop is made of two latches!
Time Borrowing

- In a flop-based system:
  - Data launches on one rising edge
  - Must setup before next rising edge
  - If it arrives late, system fails
  - If it arrives early, time is wasted
  - Flops have hard edges

- In a latch-based system
  - Data can pass through latch while transparent
  - Long cycle of logic can borrow time into next
  - As long as each loop completes in one cycle
Time Borrowing Example

Loops may borrow time internally but must complete within the cycle.
How Much Borrowing?

2-Phase Latches

\[ t_{\text{borrow}} \leq \frac{T_c}{2} - (t_{\text{setup}} + t_{\text{nonoverlap}}) \]

Diagram showing the timing relationships between \( \phi_1 \), \( \phi_2 \), \( t_{\text{borrow}} \), \( t_{\text{setup}} \), and \( t_{\text{nonoverlap}} \) within a 2-phase latch circuit.
Clock Skew

- We have assumed zero clock skew
- Clocks really have uncertainty in arrival time
  - Decreases maximum propagation delay
  - Increases minimum contamination delay
  - Decreases time borrowing
Skew: Flip-Flops

\[ t_{pd} \leq T_c - \left( t_{pcq} + t_{\text{setup}} + t_{\text{skew}} \right) \]

sequencing overhead

\[ t_{cd} \geq t_{\text{hold}} - t_{ccq} + t_{\text{skew}} \]
2-Phase Latches

\[ t_{pd} \leq T_c - \left(2t_{pdq}\right) \]

sequencing overhead

\[ t_{cd_1}, t_{cd_2} \geq t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}} + t_{\text{skew}} \]

\[ t_{\text{borrow}} \leq \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}} + t_{\text{skew}}\right) \]
Two-Phase Clocking

- If setup times are violated, reduce clock speed
- If hold times are violated, chip fails at any speed
- In this class, working chips are most important
  - No tools to analyze clock skew
- An easy way to guarantee hold times is to use 2-phase latches with big nonoverlap times
- Call these clocks $\phi_1, \phi_2$ (ph1, ph2)
Safe Flip-Flop

- In class, use flip-flop with nonoverlapping clocks
  - Very slow – nonoverlap adds to setup time
  - But no hold times
- In industry, use a better timing analyzer
  - Add buffers to slow signals if hold time is at risk
Summary

- **Flip-Flops:**
  - Very easy to use, supported by all tools

- **2-Phase Transparent Latches:**
  - Lots of skew tolerance and time borrowing

- **Pulsed Latches:**
  - Fast, some skew tol & borrow, hold time risk

<table>
<thead>
<tr>
<th></th>
<th>Sequencing overhead ((T_c - t_{pd}))</th>
<th>Minimum logic delay (t_{cd})</th>
<th>Time borrowing (t_{borrow})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flip-Flops</td>
<td>(t_{pcq} + t_{setup} + t_{skew})</td>
<td>(t_{hold} - t_{eq} + t_{skew})</td>
<td>0</td>
</tr>
<tr>
<td>Two-Phase Transparent Latches</td>
<td>(2t_{pdq})</td>
<td>(t_{hold} - t_{eq} - t_{nonoverlap} + t_{skew}) in each half-cycle</td>
<td>(T_c \cdot \frac{1}{2} - (t_{setup} + t_{nonoverlap} + t_{skew}))</td>
</tr>
<tr>
<td>Pulsed Latches</td>
<td>(\max(t_{pdq}, t_{pcq} + t_{setup} - t_{pew} + t_{skew}))</td>
<td>(t_{hold} - t_{eq} + t_{pew} + t_{skew})</td>
<td>(t_{pew} - (t_{setup} + t_{skew}))</td>
</tr>
</tbody>
</table>