Sequential Circuits

Combinational Circuits

Outputs depend on the current inputs

Sequential Circuits

- Outputs depend on current and previous inputs
- Requires separating previous, current, and future
- Called states or tokens
- Example: Finite State Machines (FSMs), Pipelines
Sequential Circuits

If tokens moved through pipeline at constant speed, no sequencing elements will be needed
   Ex: Fibre-optic cable, called \textit{wave pipelining} in circuits

However, dispersion is high in most circuits
   We need to delay fast tokens, so that they don't catch up with slow tokens

Use flip-flops to delay fast tokens so that they move through exactly one stage per cycle

Inevitably adds some delay to slow tokens

Makes circuit slower than just the logic delay
   Called sequencing overhead

Sometimes called clocking overhead
   But it applies to asynchronous circuits too
   Inevitable side effect of maintaining sequence
Sequential Elements

Latch
- Level sensitive
- Transparent latch
- D latch

Flip-Flop
- Edge triggered
- Master-slave flip-flop
- D flip-flop, D register
Sequential Circuits

Sequential Elements: Latch

Pass Transistor Latch

Pros:
- Tiny
- Low clock loads

Cons:
- $V_t$ drop
- nonrestoring
- backdriving
- output noise sensitivity
- dynamic
- diffusion input
Sequential Elements: Latch

Transmission Gate Latch
- No $V_t$ drop
- Requires inverted clock

Inverting Buffer
Pros:
- Restoring
- No backdriving
- Fixes either:
  - output noise sensitivity
  - Or diffusion input

Cons:
- Inverted output
Sequential Circuits

Sequential Elements: Latch

**Tristate feedback**

- Static
- Backdriving risk

Static latches are now essential

**Buffered Input**

- Fixes diffusion input
- Noninverting
Sequential Elements: Latch

**Buffered Output**
- Non backdriving

Widely used in standard cells
- Very robust (important feature)
- Rather large
- Rather slow (1.5 - 2 FO4 delays)
- High clock loading

**Datapath Latch**
- Smaller, faster
- Unbuffered input
Sequential Elements: Flip-Flop

Flip-Flop

Built as a pair of back-to-back latches

D
X
Q

D
X
Q
**Sequential Elements**

**Enable**

Ignore clock when enable is inactive
- Mux: increase latch D-Q delay
- Clock-gating: increase enable setup time, skew

**Symbol**

**Multiplexer Design**

**Clock Gating Design**
## Sequential Elements

### Reset

Force output low when reset is asserted

Synchronous vs. asynchronous
Sequential Elements

**Set / Reset**

Set forces output high when asserted

Flip-Flop with asynchronous set and reset
Timing Diagrams

Contamination and propagation delays

<table>
<thead>
<tr>
<th>Symbol</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>
Sequencing Methods

- **Flip-Flops**

- **2-Phase latches**

- **Pulsed latches**
Max-Delay: Flip-Flops

\[ T_C = t_{pcq} + t_{pd} + t_{setup} \]

\[ t_{pd} \leq T_C - (t_{setup} + t_{pcq}) \]

sequencing delay
Max-Delay: 2-Phase Latches

\[ T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2} \]

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

sequencing delay
Max-Delay: Pulsed Latches

\[ T_C \geq \max(t_{pdq} + t_{pd} t_{pcq} + t_{pd} + t_{setup} - t_{pw}) \]

\[ t_{pd} \leq T_C - \max(t_{pdq} t_{pcq} + t_{setup} - t_{pw}) \]

sequencing delay

\[ T_C \geq \max(t_{pdq} + t_{pd} t_{pcq} + t_{pd} + t_{setup} - t_{pw}) \]
Min-Delay: Flip-Flops

\[ t_{cd} \geq t_{hold} - t_{ccq} \]
Min-Delay: 2 Phase Latches

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

Hold time reduced by nonoverlap

Paradox: Hold applies twice each cycle vs. only once for flops
But flops have two latches!!!
Min-Delay: Pulsed Latches

\[ t_{cd} \geq t_{hold} - t_{ccq} + t_{pw} \]

Hold time increased by pulse width
**Time Borrowing**

In a flip-flop based system

- Data launches on one rising/falling edge
- Must setup before next rising/falling 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 when transparent
- Long cycle of logic can borrow time into the next cycle
- As long as each loop completes in one cycle

This mechanism is called *time borrowing*
**Time Borrowing**

Loops may borrow time internally but must complete within the cycle.
**Time Borrowing**

**How much borrowing?**

2 phased latches: \( t_{\text{borrow}} \leq \frac{T_C}{2} - (t_{\text{setup}} + t_{\text{nonoverlap}}) \)

pulsed latches: \( t_{\text{borrow}} \leq t_{\text{pw}} - t_{\text{setup}} \)
Clock Skew

We have assumed zero clock skew
Clock really have uncertainty in arrival time
decrease max-delay
increases min-delay
decreases time borrowing

Clock Skew: Flip-flops

\[ t_{pd} \leq T_c - (t_{pcq} - t_{setup} - t_{skew}) \]

\[ t_{cd} \geq t_{hold} - t_{ccq} + t_{skew} \]
2 phased latches

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

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

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

pulsed latches

\[ t_{pd} \leq T_c - \max(t_{pdq}, t_{pcq} + t_{setup} - t_{pw} + t_{skew}) \]

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

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