# Recap

- X74_163 synchronous counter
- FSM using counters

## RECAP
2 D FF Timing

Setup time: min time before K for data to be valid
2 D FF Timing (cont.)

**Hold Time**

\[ T_{\text{hld}} = \text{min time after } K \text{ for which data must remain valid} \]

![Diagram showing hold time](image)

What happens if setup and/or hold is violated? Indeterminate

![Diagram showing metastability](image)
2 D FF TIMING (cont.)

2.1 Shift Register Revisited

![Diagram of shift register with timing arrows and symbols]
2 D FF TIMING (cont.)

2.1.1 Timing Constraints for Shift Register

FF #0: If input = 0 all time, \( t_{su} \), \( t_{hld} \) automatic

FF #1: constraint \( t_{su} \): \( D_1 \) must be stable \( t_{su} \) before clock edge 2

\[ \Rightarrow t_{period} - t_{ckomax} > t \]

constrain \( t_{hold} \): \( D_1 \) must be stable for \( t_{hld} \) after clock edge 1

\[ \Rightarrow t_{ckomin} > t_{hld} \]
2 D FF TIMING (cont.)

2.2 Xilinx Timing (4005-4)

Shift register holdtime \( t_{\text{cko}} \) 0 min
setup time \( t_{\text{Dick}} \) 3.7 ns
\( t_{\text{IHCK}} \) 6.1 ns

\( F'/G' \) and \( H' \)

\( T_{\text{hold}} = 0 \), but watch out if clock skew
2 D FF TIMING (cont.)

2.2 Xilinx Timing (cont.)

How fast can it run?

\[
\frac{1}{\text{min clock period}} = \frac{1}{t_{cko} + t_{setup}}
\]

Depends on routing delay (non-deterministic)

Can use timing analyzer in XactStep

Example: 4-bit shift register: \( \sim 100 \) MHz clock (everything else is slower)
2 D FF TIMING (cont.)

2.3 Clock Skew

What is skew?

- Timing delays due to distance
  \[(2 \times 10^8 \text{ m/s}) \quad 1 \text{ ns} = 0.2 \text{ m} \quad \text{(off chip)}\]
- Clock distribution network on chip

See in shift register example

What is max \((T_0 - T_1)\) for proper operation?

min \((T_0 - T_1)\) for proper operation?
2 D FF TIMING (cont.)

2.3 Clock Skew (cont.) – $T_1 > T_0$

Ok if $t_{cko} > T_1 - T_0 + t_{hld}$
2 D FF TIMING (cont.)

2.3 Clock Skew (cont.) – $T_1 < T_0$

For $T_1 < T_0$, no problem due to clock skew

Ok if $T_0 + T_{cko} > T_1 + T_{hold}$

well designed logic $t_{cko} > T_{hold}$, and we are given $T_0 > T_1$.

How to guarantee low clock skew? ~0.1 ns

BUFFGS

clock source $\xrightarrow{\text{Global_clock}}$ limit 4
3 OTHER FF TYPES

Toggle Flip Flop

FTRSE

Convert D to T

<table>
<thead>
<tr>
<th>T</th>
<th>PS</th>
<th>NS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ D = T \oplus Q \]
3 OTHER FF TYPES (cont.)

**JK FF**

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>$Q_{n+1}$</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$Q_n$</td>
<td>hold</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>clear</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$\overline{Q}_n$</td>
<td>toggle</td>
</tr>
</tbody>
</table>

Convert D to JK

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>$Q_n$</th>
<th>$Q_{n+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
D = \overline{J} \overline{K} \cdot Q_n + JK \cdot \overline{Q}_n + JKQ_n + JKQ_n \\
= \overline{K}Q_n + J\overline{Q}_n
\]