D-type FF

\[ D \rightarrow S \quad Q \leftrightarrow D \quad \text{on } CLK \]

\[ \overline{Q} \]

Timing Diagrams

Draw logic state vs time

CLK

SR FF

More Feedback

Expect:

CLK

Q

\( \overline{Q} \)
Get:

\[ \text{Race Conditions!} \]

- \( Q \rightarrow \bar{Q} \rightarrow Q \) as fast as possible while gate is on!

Two Solutions

**Master-Slave**

![Master-Slave Circuit Diagram]

- \( Q_1 \) and \( Q_2 \)

**Edge Trigger**

State can only change when \( CLK \) edge goes up (or down)
Add input to feedback D F/F

State only oscillates when J=1

More general:

Note, S=R=1 is impossible

TT for JK F/F

<table>
<thead>
<tr>
<th>$Q_0 \overline{Q_0}$</th>
<th>J</th>
<th>K</th>
<th>$S=\overline{JK}$</th>
<th>$\overline{S}=\overline{JK}$</th>
<th>$Q_1 \overline{Q_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>01</td>
</tr>
</tbody>
</table>

Sometimes for K doesn't matter
<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$J$</th>
<th>$K$</th>
<th>$Q_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Can restate TT as excitation table.

Give changes to state then cause

<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>$J$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>x</td>
<td>0</td>
</tr>
</tbody>
</table>

State diagram

Scalers

Recall feedback D P/F

$Q = N \% 2$
Q is a clock with half the freq.!

CLK

1 2 3 4 5 6

This is a countdown timer.

Same with JK or T F/F

Count up by using falling edge trigger

CLK 0 1 2 3 4 5 6 7

Q_0

Q_1

Q_0 Q_1
Shift Register

A register is a series of F/F's

\[ B_1B_2B_3 \rightarrow A_1A_2A_3 \text{ on falling clock edge} \]

Now connect \( Q \rightarrow D \)

Serial In, Parallel Out
Modulo-$m$ Counter

If $m$ is $2^n$ can use Ripple Counter
Else need more circuits
Always need $n$ FF's
where $2^{n-1} < m < 2^n$

General Idea
Each state (bits) sets up inputs to go to the next state
Have to write down all the transitions!

State diagram

```
000
100
011
010
100
000
```

Mod-5 counter

```
3 FF's (JK)
Q0 lowest bit
Q1
Q2 highest bit
```

Transition Table

<table>
<thead>
<tr>
<th>Current</th>
<th>Next</th>
<th>$J_2$</th>
<th>$K_2$</th>
<th>$J_1$</th>
<th>$K_1$</th>
<th>$J_0$</th>
<th>$K_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>001</td>
<td>010</td>
<td>0</td>
<td>x</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>011</td>
<td>100</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>100</td>
<td>000</td>
<td>1</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Three Inputs

Three inputs

Six Functions

Karnaugh Orgy!

Note: 110, 101, 111 don't appear: they don't matter
\[ J_0 \]
\[ K_0 = Q_3 \]
\[ J_1 = Q_0 \]
\[ K_1 = Q_0 \]
\[ J_2 = Q_0 \cdot Q_3 \]
\[ K_2 = 1 \]
Counters in the labs

CD4029: already used in Lab 5

Binary/Decimal: \( p9 = 0 \) decimal \( (10) \)
\( = 1 \) binary \( (16) \)

Up/Down: \( p10 = 0 \) down
\( = 1 \) up

Presettable: \( p1 = 1 \)
\( p3, p13, p12, p4 \)

Outputs
\( p2, p14, p11, p6 \)

\( p7 : \text{carry out} \)

Carry In:
only count if \( p5 \) is \( B \) low
Presetable Divide-By-N Counter

Divide by 2—10

Basically a 5 part shift register with feed back

\[ \overline{Q_1} \rightarrow D \quad /2 \]
\[ \overline{Q_2} \rightarrow D \quad /4 \]
\[ Q_n \rightarrow D \quad /2n \]

\[ \overline{Q_1} \cdot \overline{Q_2} \quad /3 \]

\( Q_1 \)
\( Q_2 \)
\( \overline{Q_1 \cdot Q_2} = \overline{Q_1} \cdot \overline{Q_2} \)

\[ \overline{Q_2} \cdot \overline{Q_3} \quad /5 \]
\[ \overline{Q_3} \cdot \overline{Q_4} \quad /7 \]
\[ \overline{Q_4} \cdot \overline{Q_5} \quad /9 \]
LED's & Displays

\[ V_0 = 1.6 \text{ V} \]
\[ I = 10 \text{ mA} \]

1.6 is "low" so CMOS chip can't drive anything else.

Seven Segment Decoder/Driver

\{\text{74C48 \& MC14495}\}

Four inputs
Seven outputs

Translate 4-bits to Hex character