Experiment 9
Flip-Flops

• Objectives:
  – Understand how flip-flops can be developed from basic logic gates
  – Become familiar with the behavior of S-R and J-K flip-flops
  – Understand how a flip-flop can serve as a storage element in a simple controlled circuit.
Logic Gate and Flip-Flop Comparison

Logic Gate: The output of a logic gate depends only on the current values of its inputs.

Flip-Flop: The output of a flip-flop (after a time delay) depends on the current values of its inputs and its current state (i.e., its current output value),

Since a flip-flop can retain its current state even though its inputs may change, it can be used as a memory device or storage device.
The S-R Flip-Flop

S-R Flip-Flop Symbol

Input 1

Input 2

Truth Table for S-R Flip-Flop

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S^n$</td>
<td>$R^n$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The superscript $n$ is used to indicate the “current value” of the input or flip-flop state. The superscript $n+1$ is used to indicate the “subsequent value,” i.e., the value that results after the flip-flop has had time to react to the current inputs.
The S-R Flip-Flop

Truth Table Meaning:

1. If \( S^n = R^n = 0 \) and the current state of the flip-flop is \( Y^n \), then the next state \( Y^{n+1} \) will be the same as the current state.

2. If \( S^n = 0 \) and \( R^n = 1 \), then the next state \( Y^{n+1} \) will be 0, regardless of the current state of the flip-flop.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S^n )</td>
<td>( R^n )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The S-R Flip-Flop

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S^n$</td>
<td>$R^n$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Truth Table Meaning:

3. If $S^n = 1$ and $R^n = 0$, then the next state $Y^{n+1}$ will be 1, regardless of the current state of the flip-flop.

4. If $S^n = R^n = 1$, then the next "state" of the flip-flop $Y^{n+1}$ is **contradictory**; for example, $y = \bar{y} = 0$. 
The S-R Flip-Flop: The 1-1 Input Case

4. If \( S^n = R^n = 1 \), then the next "state" of the flip-flop \( Y^{n+1} \) is **contradictory**; for example, \( y = \bar{y} = 0 \).

**Note:** Both outputs of the flip-flop do have values, but these values contradict our definition of the complement of a variable.

Thus, the 1-1 input case should be avoided!

**Furthermore:** If both \( S \) and \( R \) are reset to 0 simultaneously, the next state \( y \) of the flip-flop cannot be reliably determined.

The J-K flip-flop overcomes this “flaw” of the S-R flip-flop.
Construction of the S-R Flip-Flop:

The “flip-flop”

Note the “crossover” as compared to the S-R symbol

Construction of the Synchronous S-R Flip-Flop:

Clock signal
Construction of the Synchronous S-R Flip-Flop:

Clock signal

Clocked S-R Flip-Flop Symbols:
Construction of the J-K Flip-Flop from the Synchronous S-R Flip-Flop:

The J-K flip-flop has a valid response to a 1-1 input.

If $J^n = K^n = 1$, and the current state of the flip-flop is $Y^n$, then the next state $Y^{n+1}$ will be $Y^{n+1} = \overline{Y^n}$.
Circuit for Testing Digiac Flip-Flops
Operating as J-K Flip-Flops

J-K flip-flop operation is synchronous

S-R operation is asynchronous and uses negative active inputs

J-K flip-flop operation is synchronous
Controlled Operation Circuit with Storage

Circuit performs the operation

\[ \tau: A \leftarrow [\overline{Q} \cdot (A \cdot S) + Q \cdot (A + S)] \]

where the symbols represent:
- \( \tau \) – a clock pulse
- \( A \) – accumulator register
- \( S \) – memory location
- \( Q \) – operation code

The circuit produces a result of either \( A \cdot S \) or \( A + S \), depending on the value of the operation code \( Q \).

*The result is placed back in \( A \) when a clock pulse occurs*
Implementation Notes

- In #2 and #3, use a switch to represent the clock.
- In #5 and #6, use the built-in clock of the Digiac.
- In #’s 1 through 3, try to wire your circuits in a symmetric manner from top to bottom; e.g., use the same length wires for the feedback paths and for other similar circuit paths; use the same type of gates to do similar things (don’t use an AND gate for the J input and then use a NAND followed by a NOT for the K input).