Unit 11
Latches

Combination Circuit

- no memory (no state)
- same input will always produce the same output
- output does not depend on the history (sequence) of inputs

Sequential Circuit

- stores state in storage elements
- output depends on the history or sequence of inputs
Sequential circuits

Synchronous → can be defined from the knowledge of its signals at discrete instances of time.
Timing → clock pulses

Asynchronous → depend on the input at any instance of time and the order in continuous time in which the input changes.
In general, complex asynchronous circuits are difficult to design because behavior depends on propagation delays (tpd) of the gate.

Synchronous Clocked Sequential Circuit

Changes state only in response to a clock pulse.
This circuit can hold two different stable states in memory.
Set-Reset Latch (SR Latch)

Output depends not only on the present inputs but also on the past sequence of inputs.

If we restrict the input so that \( S = 1, R = 1 \) is not allowed then \( P = Q' \). With this restriction this circuit is called an SR Latch.

Same circuit drawn in symmetric cross-coupled form.
when $S=1 \& R=1$ we have unstable behavior

oscillations

unpredictable behavior

S \rightarrow 0 \quad 0 \rightarrow 1 \rightarrow 0 \rightarrow 1

R \rightarrow 0

oscillate
Latches (Flip-flops are constructed from latches)

Stored value can be changed

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>( \bar{Q} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Set state

Reset state

Undefined

SR Latch (with NOR Gates)

Normal conditions \( R=0, S=0 \) \( \rightarrow \) state is preserved

Set state or reset state

Applying 1 to S \( \rightarrow \) latch goes to set state

Bring S to 0 \( \rightarrow \) latch remains in set state

Applying 1 to R \( \rightarrow \) latch goes to reset state

Bring R to 0 \( \rightarrow \) latch remains in reset state
make sure both S & R do not become 1 at the same time → undefined state

normal operation S = 1, R = 1 → preserves state
S = 0 → latch goes to Set state
S = 1 → latch remains in Set state
R = 0 → latch goes to Reset state
R = 1 → latch remains in Reset state

S = 0, R = 0 is avoided

SR Latch with control input

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>R</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Reset state (Q = 0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Set state (Q = 1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

State cannot change
Also known as Gate

D Latch

Ensures that S and R are never equal to 1 (R = 3)

Gated D Latch

Also known as Gate

D Latch with Logical 1 Control

D Latch with Logical 0 Control
SR Latch Characteristic Equation

\[ Q(t) \rightarrow \text{current state} \]
\[ P(t) \rightarrow \text{current output} \]
\[ Q^+ \rightarrow \text{next state} \]
\[ Q(t+\varepsilon) \]

\[ Q^+ = ((S+Q)' + R)' = (S+Q)R' = RS' + RQ' \]
\[ P = (Q+S)' = S\overline{Q}' \]

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State ( Q^+ )</th>
<th>Present Output ( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>SR SR SR SR SR</td>
<td>SR SR SR SR SR</td>
</tr>
<tr>
<td></td>
<td>00 01 11 10</td>
<td>00 01 11 10</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 1</td>
<td>1 1 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>1 0 1 1</td>
<td>0 0 0 0 0</td>
</tr>
</tbody>
</table>

\( 0 \rightarrow \text{stable state} \)

For all stable states \( P = Q' \) except when \( S = R = 1 \)
Making \( S = R = 1 \) don't care

\[
\begin{array}{c|ccc|c}
S & R & Q & Q' + \\
\hline
00 & 0 & 1 & 0 \\
01 & 1 & X & X \\
10 & 0 & X & X \\
11 & X & X & X \\
\end{array}
\]

not allowed

\[ Q^+ = S + R'Q \]  \text{characteristic equation (next-state equation)}

Uses of SR Latches:
- used as a component in flip-flops
- debouncing switches

![Diagram of SR Latch and debouncing circuit]

+V

\( a \)

\( b \)

double throw switch

bounce at 1

bounce at \( a \)

bounce at \( b \)

switch at \( a \)

switch between \( a \) & \( b \)

Switch at 1
GatedLatchCharacteristic
Equation

State is preserved
transparent latch

Q⁺ = G'Q + GD  characteristic equation of a Gated Latch