Storage elements

- Previously, we have considered combinational circuits where the output values depend only on the values of signals applied to the inputs.
- Another class of logic circuits have the property that the outputs depend not only on the current inputs, but also on the past behavior of the circuit.
- Such circuits include storage elements that store the values of logic signals.
Sequential circuits

• Contents of the storage elements represent the **state** of the circuit
• Input value changes may leave the circuit in the same state or cause it to change to a new state
• Over time, the circuit changes through a sequence of states as a result of changes in the inputs
• Circuits that exhibit this behavior are referred to as **sequential circuits**

Alarm control system

• Suppose we wish to construct an alarm circuit such that the output remains active (on) even after the sensor output that triggered the alarm goes off
  – A typical car alarm is representative of this type of circuit
• The circuit requires a memory element to remember that the alarm has to be active until a **reset** signal arrives

```
Sensor        Set
             |
|              Memory element
|                        On/Off’
|         Alarm
Reset
```


**A simple memory element**

Simple memory element: feedback path provides basis for the ‘remembering’ of data

Load = 0, TG2 = on providing a feedback path

Load = 1, TG1 = on providing a way to load new data

---

**Basic SR latch**

- A similar circuit, constructed with NOR gates can also be constructed
- Inputs, **Set** and **Reset**, provide the means to changing the state, Q, of the circuit
- This circuit is referred to as a basic latch
Basic SR latch

- When R=S=0 the circuit remains in its current state (either Q_a=1 and Q_b=0 or Q_a=0 and Q_b=1)
- When S=1 and R=0, the latch is **set** into a state where Q_a=1 and Q_b=0
- When S=0 and R=1, the latch is **reset** into a state where Q_a=0 and Q_b=1
- Where S=1 and R=1, Q_a=Q_b=0 (there are actually problems with this state as we will see)

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q_a</th>
<th>Q_b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0/1</td>
<td>1/0</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>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Basic SR latch timing diagram
Basic SR latch timing diagram

- If the propagation delays from \( Q_a \) and \( Q_b \) are exactly the same, the oscillation at time \( t_{10} \) would continue indefinitely.
- In a real circuit there would probably be some (mostly insignificant) difference in the delays and the latch would eventually settle into one of its two stable states (but we don’t know which one it would be).
- Thus the \( S=R=1 \) combination is generally considered an unallowed combination in the SR latch.

Gated SR latch

- The basic SR latch changes its state whenever its inputs change.
- It may be desirable to add an enable signal to the basic SR latch that allows us to control when the circuit can change states.
- Such a circuit is referred to as a gated SR latch.
Gated SR latch circuit

CLK    S   R   Q(t+1)
0      X   X   Q(t)
1      0   0   Q(t)
1      0   1   0
1      1   0   1
1      1   1   X

Q(t) = present state
Q(t+1) = next state

Gated SR latch timing diagram

Time
Gated SR latch with NAND gates

\[\begin{array}{c}
S \\
\text{Clk} \\
R \\
\end{array} \quad \begin{array}{c}
\text{NAND} \\
\text{NAND} \\
\end{array} \quad \begin{array}{c}
Q \\
\bar{Q} \\
\end{array}\]

Gated D latch

- Another useful latch has a single data input, \(D\), and it stores the value of this input under the control of a clock signal.
- This is referred to as a **gated D latch**
  - Useful in circuits where we want to store some value
  - The output of an adder/subtractor circuit would be one example.

\[\begin{array}{c}
D \text{ (Data)} \\
\text{Clk} \\
\end{array} \quad \begin{array}{c}
\text{NAND} \\
\text{NAND} \\
\end{array} \quad \begin{array}{c}
Q \\
\bar{Q} \\
\end{array}\]
**Gated D latch**

<table>
<thead>
<tr>
<th>CLK</th>
<th>D</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>Q(t)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- **CLK**
- **D**
- **Q**

---

**Level versus edge sensitivity**

- Since the output of the D latch is controlled by the level (0 or 1) of the clock input, the latch is said to be **level sensitive**
  - All of the latches we have seen have been level sensitive
- It is possible to design a storage element for which the output only changes at the point in time when the clock changes from one value to another
- Such circuits are said to be **edge triggered**
  - Can be **positive edge triggered**
    - Sensitive to the 0->1 transition of the clock
  - Can be **negative edge triggered**
    - Sensitive to the 1->0 transition of the clock

---

**Dr. D. J. Jackson  Lecture 24-15**
Effects of propagation delays

- Previously we have ignored the effects of propagation delay. In practical circuits, it is essential to account for these delays.
- For the gated D latch (and others as well), it is important that the value of D not change at the time the clock (clk) goes from 1 to 0.
  - The designer must make sure the signal is stable when the critical change in the clock takes place.
- The minimum time the D signal must remain stable prior to the active edge of the clock signal is called the setup time ($t_{su}$).
- The minimum time the D signal must remain stable after the active edge of clock is the hold time ($t_{h}$).
  - Typical CMOS values are: $t_{su}=3\text{ns}$ and $t_{h}=2\text{ns}$.

Setup and hold times

![Diagram showing setup and hold times](image-url)