SR Latch - Complete Summary

Your Name

July 2, 2025

1 Introduction

An **SR Latch (Set-Reset Latch)** is a basic **bistable** (two stable states) sequential circuit that stores **1 bit** of data. It is constructed using **cross-coupled NOR gates** or **NAND gates**, forming a feedback loop to maintain its state.

2 Types of SR Latches

- **NOR-based SR Latch**

- **NAND-based SR Latch** (also called **SR Latch with active-low inputs**)

3 NOR-based SR Latch

3.1 Structure

Two NOR gates with feedback.

3.2 Inputs

- **S (Set)** – Sets output $Q = 1$

- **R (Reset)** – Resets output $Q = 0$

3.3 Truth Table

S	R	Q	Q'	State
0	0	Q	Q'	**Hold (Memory)**
1	0	1	0	**Set (Q = 1)**
0	1	0	1	**Reset (Q = 0)**
1	1	0	0	**Invalid (Race)**

3.4 Invalid State

When $S = R = 1$, both outputs Q and Q' become 0 (violates $Q' = \overline{Q}$).

4 NAND-based SR Latch (Active-Low Inputs)

4.1 Structure

Two NAND gates with feedback.

4.2 Inputs

- $\overline{S}$ (Set) – Active-low (0 sets $Q = 1$)

- $\overline{R}$ (Reset) – Active-low (0 resets $Q = 0$)

4.3 Truth Table

$\overline{S}$	$\overline{R}$	Q	Q'	State
1	1	Q	Q'	**Hold (Memory)**
0	1	1	0	**Set (Q = 1)**
1	0	0	1	**Reset (Q = 0)**
0	0	1	1	**Invalid (Race)**

4.4 Invalid State

When $\overline{S} = \overline{R} = 0$, both outputs Q and Q' become 1 (violates $Q' = \overline{Q}$).

5 Characteristics

- **Level-Triggered**: Changes state based on input levels (not edges).

- **Asynchronous**: No clock signal required.

- **Race Condition**: Occurs when both inputs are active simultaneously.

6 Applications

- Basic memory storage in registers.

- Debouncing switches.

- Temporary state storage in control circuits.

7 Limitations

- **No Clock Control**: Cannot synchronize with a clock (unlike flip-flops).

- **Glitches**: Sensitive to input changes.

- **Invalid State**: Must avoid $S = R = 1$ (NOR) or $\overline{S} = \overline{R} = 0$ (NAND).

8 Conclusion

The **SR Latch** is the simplest sequential circuit used for **1-bit storage**. It has two stable states (**Set & Reset**) but suffers from an **invalid state** when both inputs are active. It serves as the foundation for more complex storage elements like **Flip-Flops**.