THE UNIVERSITY OF AZAD JAMMU AND KASHMIR, MUZAFFARABAD



|  |  |
| --- | --- |
| COURSE TITLE | CALD |
| COURSE CODE | CS-1205 |
| ASSIGNMENT TITLE | SR Latches |
| STUDENT NAME | Tayyab Hanif |
| STUDENT ROLL NUMBER | 2024-SE-11 |
| INSTRUCTOR NAME | Engr. Sidra Rafique |
| SUBMISSION DATE | 29th August, 2025 |
| LAB No. | 09 |

**S-R Latches**

* The **S-R latch** is the **simplest type of sequential logic circuit**.
* It is a **bistable multivibrator** → meaning it has **two stable states**: storing either 0 or 1.
* Unlike combinational circuits, its output depends on **both present input and past output** (i.e., it has memory).

**In short**.. An S-R latch is a **basic memory element**.

**Inputs and Outputs**

* Inputs: **R (Reset)** and **S (Set)**
* Outputs: **Q** and **Q̅** (complement of Q)

**Basic Implementations**

There are **two common ways** to build an S-R latch:

**(A) Using NOR gates**

* Inputs are active **HIGH**.
* Circuit: Two cross-coupled NOR gates.

**Truth Table (NOR-based latch):**

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Q (next state) | Q̅ |
| 0 | 0 | No change (holds state) | Previous |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | **Invalid** (both outputs = 0) | **Invalid** |

**(B) Using NAND gates**

* Inputs are active **LOW**.
* Circuit: Two cross-coupled NAND gates.

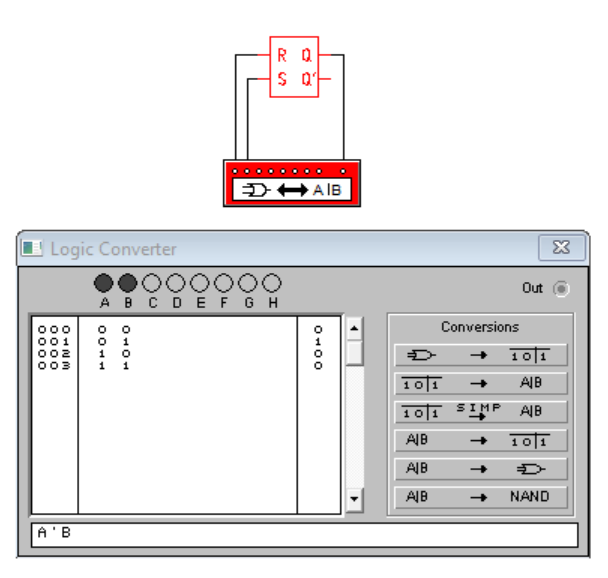
**Truth Table (NAND-based latch):**

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Q (next state) | Q̅ |
| 1 | 1 | No change (holds state) | Previous |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | **Invalid** (both outputs = 1) |  |

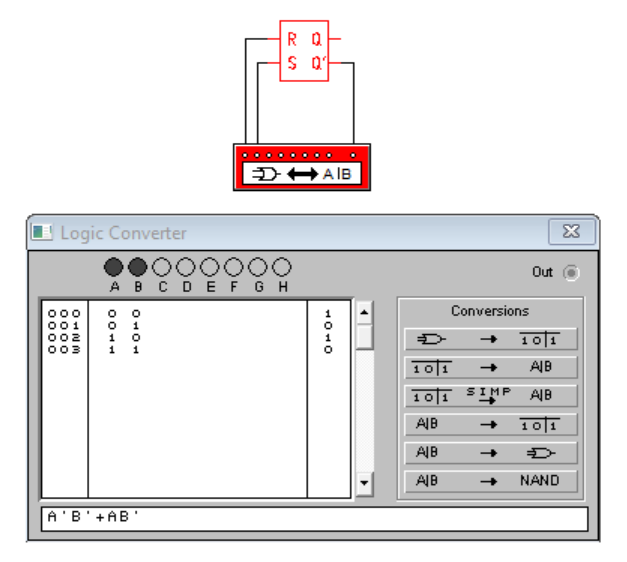
**Behavior Explanation**

* **Set (S=1, R=0 in NOR / S=0, R=1 in NAND)** → Forces Q = 1.
* **Reset (S=0, R=1 in NOR / S=1, R=0 in NAND)** → Forces Q = 0.
* **Hold (S=R=0 in NOR / S=R=1 in NAND)** → Keeps previous state.
* **Invalid (S=R=1 in NOR / S=R=0 in NAND)** → Produces undefined output (Q and Q̅ both same).

**Truth table & Expression (Q):**



**Truth table & Equation (Q`):**

****