

The **74HC595** is an 8-bit serial-in, parallel-out shift register with a storage register and tri-state outputs. It is a commonly used integrated circuit (IC) in electronics projects to expand the number of digital outputs using a microcontroller like an Arduino or Raspberry Pi.

Key Features:

1. **Shift Register:**
 - The 74HC595 receives serial data (one bit at a time) through a data pin.
 - It then shifts the data through a series of flip-flops when a clock pulse is received.
2. **Storage Register:**
 - After shifting, the data is latched into a storage register.
 - The storage register holds the 8 bits of data, which are then output through the 8 parallel pins.
3. **Tri-State Output:**
 - The output pins can be set to a high-impedance state to avoid conflicts with other devices sharing the same output lines.
4. **Cascading:**
 - Multiple 74HC595 ICs can be chained together to control more outputs by connecting the serial output (Q7') of one to the serial input (DS) of the next.

Pinout:

- **DS (Data Serial Input):** Serial data input.
- **SHCP (Shift Register Clock):** Shifts data on a rising edge of the clock.
- **STCP (Storage Register Clock):** Latches data into the output register on a rising edge.
- **OE (Output Enable):** Enables or disables outputs (active low).
- **MR (Master Reset):** Clears all data when active (active low).
- **Q0-Q7 (Parallel Outputs):** Outputs the stored 8-bit data.
- **Q7' (Serial Output):** Outputs the serial data for chaining to another shift register.

What It Does:

1. **Expands Output Pins:**
 - Microcontrollers have a limited number of GPIO (General Purpose Input/Output) pins. The 74HC595 lets you control up to 8 outputs using just 3 pins from the microcontroller (Data, Clock, and Latch).
2. **Controls LEDs, Motors, etc.:**
 - Often used to control multiple LEDs, seven-segment displays, relays, or other digital components.
3. **Efficient Use of Microcontroller Resources:**
 - Since data is sent serially, fewer microcontroller pins are needed.

Typical Use Case:

1. **Connecting to Arduino:**
 - Send data serially using the `shiftOut()` function.
 - Use one pin for the data, one for the clock, and one for the latch.
 - Update the latch to display the new data on the outputs.
2. **Driving a 7-Segment Display:**
 - The 8 outputs can connect to the segments (a-g + dp) of a single 7-segment display.
 - Multiple shift registers can be cascaded to control multiple displays.

How It Works:

1. **Serial Data Input:**
 - Data is input bit by bit via the DS pin.
 - A clock pulse (SHCP) shifts the bits into the shift register.
2. **Latch Data:**
 - Once all 8 bits are shifted in, a pulse on the STCP pin transfers the data to the storage register.
3. **Output:**
 - The data in the storage register is output on Q0-Q7.
4. **Cascading:**
 - If more than 8 outputs are needed, the Q7' pin of one 74HC595 can be connected to the DS pin of another, allowing the chain to grow as needed.

Advantages:

- Reduces the number of required GPIO pins.
- Can drive many outputs with just one IC.
- Easy to use and widely supported in hobbyist projects.

[7400 Series Guide: 74HC595 Serial-to-Parallel Shift Register](#)

[How 74HC595 Shift Register Works ? | 3D animated](#)

[How Shift Registers Work - The Learning Circuit](#)