8-bit Parallel-In/Serial-Out shift register

Daniel Josué Rodríguez Agraz *Ingeniería en Electrónica y Comunicaciones, Unniversidad de Guadalajara*

I. Objectives

- Use Verilog to design and simulate an 8-bit Parallel-In/Serial-Out shift register using the IC 74HC165 specifications.
- Design a test bench for the circuit verification.

II. Introduction

A register is a digital circuit with two main functions: data storage and data movement. The storage characteristic of a register refers to the total number of bits of digital data it can hold. The shifting characteristic allows data to flow from one stage to another within the register or to its input or output, depending on the clock pulses applied.[1]

The figure **??** shows the symbol of an 8-bit parallel-in/serial-out shift register.

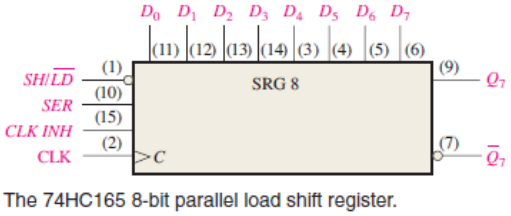


Fig. 1: Symbol of an 8-bit shift register.

A register is created by combining flip-flops to store and shift data (1's and 0's) and typically does not possess an internal state sequence.

There are two main types of registers: serial and parallel. In a serial shift register, bits are stored sequentially, as shown in Figure 2. The most significant bit (MSB) of the input signal is placed in the least significant bit (LSB) position of the register, and the rest of the bits are shifted to the right. The output is the value of the most significant bit. of the register.[1]

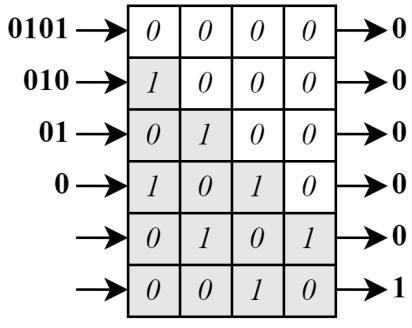


Fig. 2: Example of a 4-bit shift serial register, where each cell represents a flip-flop.

In a parallel register, bits are stored simultaneously from parallel lines, as illustrated in Figure 3.

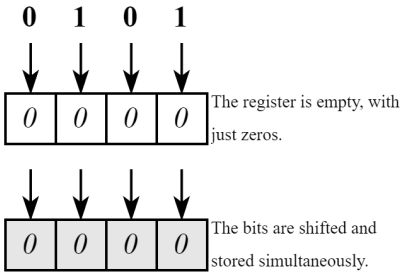


Fig. 3: Example of a 4-bit shift parallel register, where each cell represents a flip-flop.

III. Methodology

The 8-bit Parallel-In/Serial-Out shift register has been developed in two stages. In the first step the register has been designed based on the IC 74HC165 specifications available in the literature [2]. Using the truth table and the sequence diagram provided by the datasheet, shown in the Table I, extracted from the page 13th.

INPUTS			FUNCTION
SH/$\overline{LD}$	CLK	CLK INH	
L	X	X	Parallel load
H	H	X	No change
H	X	H	No change
H	L	↑	Shift(1)
H	↑	L	Shift(1)

(1) Shift : Content of each internal register shifts towards serial output Q_H. Data at SER is shifted into the first register

TABLE I: Input signals function table for the shift register SN74HC165.

As the datasheet mentions on page 1, the CLK and CLKINH pins are interchangeable, because low CLK and a low-to-high transition of CLK INH also accomplish clocking, CLK INH must be changed to the high level only while CLK is high. Parallel loading is inhibited when SH/LD is held high. While SH/LD is low, the parallel inputs to the register are enabled independently of the levels of the CLK, CLK INH, or serial (SER) inputs.[2]

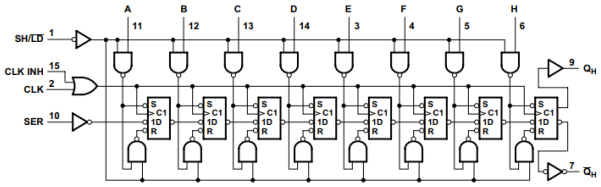


Fig. 4: Logic diagram for the shift register SN74HC165.

To develop the register, I referenced the diagrams provided in the datasheet on page 1, Figure 4. I considered that CLK and CLKINH (which is referred to as *clken* in the code) are ORed, and implemented this logic directly in the code, as shown below.

Listing 1: Verilog code for an 8 bit shift register.

```
module PISO_REG_4B(
input reg[0:7] D_in,
input reg LOAD, SER,
input reg clk, clken,
output reg Q, QNEG
);

reg [0:7] D;
```

clk	clken	LOAD	ser
0	1	1	0

TABLE II: Initialization values.

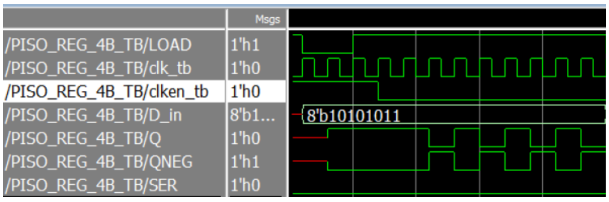
```
assign QNEG = ~Q;
assign Q = D[7];

always@(posedge clk or posedge clken)
begin
if (~LOAD)
begin
  D <= D_in;
end
else if (~clk | ~clken)
begin
  D <= D >> 1;
  D[0]<=SER;
end
end
endmodule
```

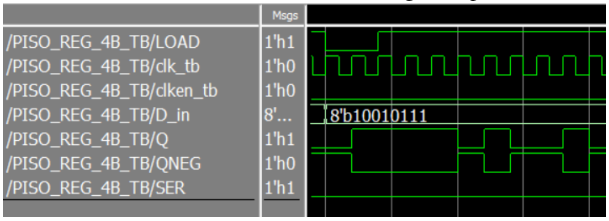
IV. Results

For the test bench all the input pins were initialized as shown in the Table II. It can be noticed from the truth table I, when initialized either clk or clkinh (clken) in a high state, the value at the output is indefinite.

We then loaded a value into the register by setting LOAD to high, and the output takes the value of the most significant bit of the data as shown on the Figure 5 where the signal in the Figure 5a imitates the time diagram provided by the datasheet on page 12.



(a) Simulation results of the shift register (part 1).



(b) Simulation results of the shift register (part 1).

Fig. 5: Time diagram of the test bench.

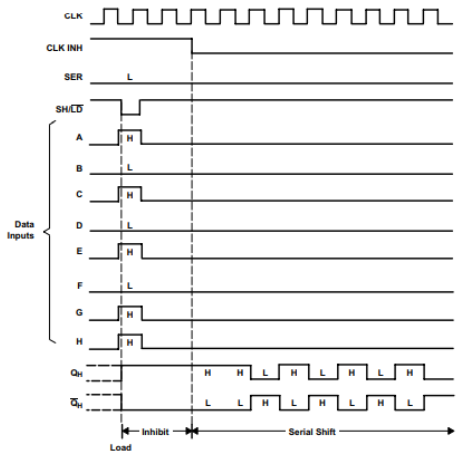


Fig. 6: Time diagram provided by the datasheet.[]

Finally, to start the shifting process, the *clken* signal was set to low. As shown in Figure 7, when the pin is set to high again, the register stops shifting, and the output takes the value of the last internal most significant bit.

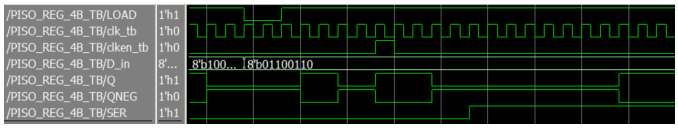


Fig. 7: Simulation results of the shift register (part 3).

The serial pin replaces the most significant bit of the internal register until all the parallel data leaves the register, this allows us to put two of this registers together to create a 16-bit shift register.

V. Conclusiones

In this document, an 8-bit Parallel-In/Serial-Out shift register was designed and tested. The register, based on the IC 74HC165 specifications, produced the same results under identical inputs during the test bench evaluation. The test bench results indicate, that the designed register can be used as a commercial Parallel-In/Serial-Out shift register.

References

[1] T. L. Floyd, *Fundamentos de Sistemas Digitales*, 9th ed.
[2] Texas Instruments, *SNx4HC165 8-Bit Parallel-Load Shift Registers*, Revised Dec. 2015 ed., dec 1982, SN74HC165 datasheet.