HW 5 - Shift Register

Jonathan Schwartz

Section 003L

TA: Connor Rickermann

Date submitted: 4/22/2021

**Theory**

Shift registers are a set of consecutive D-FFs, each of which is able to store a bit, so when a new bit of data is pushed in, the shift register, utilizing one clock to connect all the D-FFs, is able to push each bit over to the next D-FF and store the data. They can have serial input which is pushed through consecutively as described above, but they can also have parallel input which is able to use presets and clears to set the entire shift register to a new set of data immediately back to the preset values desired, regardless of the clock cycle.

We require a debouncer circuit to control the clock because when we use the clock, we need to make sure that we never have non-discrete impedance. The debouncer circuit allows us to only use high and low voltage without anything in-between.

**Testing**

| Theoretical | D | C | B | A | Experimental | D | C | B | A |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1010 Input (Serial) | 1 | 0 | 1 | 0 | 1010 Input (Serial) | 1 | 0 | 1 | 0 |
| 1011 Input (Parallel) | 1 | 0 | 1 | 1 | 1011 Input (Parallel) | 1 | 0 | 1 | 1 |
| Output Contents (Shift 0s on) | 0 | 0 | 0 | 0 | Output Contents (Shift 0s on) | 0 | 0 | 0 | 0 |
| 1010 Input (Parallel) | 1 | 0 | 1 | 0 | 1010 Input (Parallel) | 1 | 0 | 1 | 0 |
| Divide Previous By 2 | 0 | 1 | 0 | 1 | Divide Previous By 2 | 0 | 1 | 0 | 1 |

If I were to input the last two digits of my netID (15) in BCD (00010101), the decimal value of the shift register would be 21, and if I were to shift the contents of this register to the right, it would result in the new byte: 00001010, which is 10 in decimal, and this is the equivalent of a floor-division by 2.

**Questions**

What are the main functional purposes of a register?

A shift register is very useful for storing a sequence of bits as data, and to be able to shift them left and right (multiplication and division by 2). Registers are also useful for data storage in general because it allows data to be stored as it is transferred, so that it will maintain its order without error. You can also create counters and comparison codes from this because they can store a string of bits instead of just one, and this string can change if necessary.

The chip photo looks like a SN5474. Is this the same as a SN7474?

Both of the chips in question are Dual D-Type Positive-Edge-Triggered Flip-Flops with Preset and Clear options as well. The only difference I was able to find with the datasheets and pinouts online was the structure of the pinout. Each chip has the same functionality, but the pins are mapped to different inputs/outputs.