

**Experiment.No.5**

**Sequential Logic Circuits**

**Student’s names: Ahmad Salah**

**Student’s id’s: 1130083**

**1) Table of content:**

|  |  |
| --- | --- |
| **Page Number** | **Description**  |
| **1** | **Cover page** |
| **2** | **Table of content** |
| **3** | **Abstract** |
| **4** | **Theory** |
| **7** | **Procedure** |
| **17** | **Tasks** |
| **18** | **Discussion** |
| **19** | **Conclusion** |

**2) Abstract:**

**In this experiment we were able to understand the differences between combinational and sequential circuits, and the use of flip-flops and the operation of latches, but in concentrating in flip-flops, using these flip-flops we were able to make counters and shift registers using “D”, “T”, and “JK” flip-flops each of them had different connection way finally we studied synchronous and asynchronous counters.**

**3)Theory:**

**3.1) Latches and Flip flops:**

**3.1.1) RS latch with Basic Logic Gates:**

**The SET-RESET flip flop is designed with the help of two NAND gates, these flip flops are also called RS Latch.**

**3.1.2) RS latch with control input:**

**The problems with S-R flip flops using NAND gate is the invalid state. This problem can be solved by changing outputs when certain invalid states are met, regardless of the condition of either the Set or the Reset inputs.**

**3.1.3)** **D latch with RS latch:**

**D flip flop is actually a slight modification of the above explained clocked SR flip-flop. The D input is connected to the S input and the complement of the D input is connected to the R input.**

**3.1.4)** **JK latch with RS latch:**

**A J-K flip flop can also be defined as a modification of the S-R flip flop. The only difference is that the intermediate state is more repeated and precise than that of a S-R flip flop.**

**3.1.5)** **JK Flip-flop with master- slave RS latches**

**The Master-Slave Flip-Flop is basically two gated SR flip-flops connected together in a series configuration with the slave having an inverted clock pulse.**

**3.2) Registers:**

**3.2.1) Shift Register with D Flip-Flops:**

**The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop. The simplest possible shift register is one that uses only flip flop.**

**3.2.2)**  **4-Bit Shift Register with serial and parallel load:**

**The serial shift right and parallel load are activated by separate clock inputs which are selected by a mode control input, the data is transferred from the serial or parallel D inputs to the Q outputs synchronous with the HIGH to LOW transition of the appropriate clock input**

**3.3)** **Counters:**

**3.3.1) 2-bit Synchronous Counter:**

**By connecting the CLK input of the second JK flip flop to Q of the first JK FF, we obtain a 2 bit Up Counter. The output is at both Q of the flip flops.**

**The count sequence for Q1 Q0 is 00,01,10,11,00,01…**

**3.3.2)** **3-bit (divide-by-eight) Ripple Counter:**

**By connecting the CLK input of the second JK flip flop to Q of the first JK FF, and connecting the CLK of the third JK flip flop to Q of the second JK FF we obtain a 3 bit Up Counter. The output is at both Q of the flip flops.**

**The count sequence for Q2Q1Q0 is 000,001,010,011,100,101**

**4) Procedure:**

**4.1) RS latch with Basic Logic Gates:**

**The circuit of the S-R flip flop using NAND Gate, S stands for SET and R stand for RESET, and this flip-flop is designed by connecting S to the first NAND gate as the first input, and Q’ is the second one, and the first NAND gate gives Q as output, which is the input to the second NAND gate along with R, and its truth table and the design are shown below.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S** | **R** |  | **Q** | **Q’** |
| **0** | **0** |  | **1** | **1** |
| **0** | **1** |  | **1** | **0** |
| **1** | **0** |  | **0** | **1** |
| **1** | **1** |  | **0** | **1** |

****

**4.2) RS latch with control input:**

**For this, a clocked S-R flip flop is designed by adding two NAND gates to a basic NAND Gate flip flop, where the first NAND gate output (that were added) replaces the S input to the second level, so the input for it is S along with the clock, and the second NAND gate output obviously replaces R and it has R and clock as an inputs for it, as it shown in the design below along with the results.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S** | **R** |  | **Q** | **Q’** |
| **0** | **0** |  | **1** | **0** |
| **0** | **1** |  | **0** | **1** |
| **1** | **0** |  | **1** | **0** |
| **1** | **1** |  | **1** | **1** |

****

**4.3) D latch with RS latch:**

**This flip-flop is designed by adding two NAND gates to the RS flip-flop, as in the previous case, but this time D is the input instead of (S,R), so D is connected in place of S, and D’ is connected in place of R, and when the clock changes, the input changes which leads to the change in the output.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLK2** | **D** |  | **Q** | **Q’** |
| **0** | **0** |  | **0** | **1** |
| **0** | **1** |  | **0** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** |  | **0** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** |  | **1** | **0** |

****

**4.4)** **JK latch with RS latch:**

 **The behavior of inputs J and K is same as the S and R inputs of the S-R flip flop, where J is connected in the place of S (SET) and K in place of R (RESET), but in this case J and clock share the NAND gate with Q’, and for the other NAND gate K, clock, and Q represents the inputs.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLK2** | **J** | **K** |  | **Q** | **Q’** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** |  | **0** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **1** |  | **0** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **0** |  | **1** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** |  | **1** | **1** |

****

**4.5)** **JK Flip-flop with master- slave RS latches:**

**This flip-flop is constructer by adding JK latch with RS latch (Master) to RS latch (Slave), this one its more complicated than the previous cases, but they all still working on the same base ground, so in this one the slave feedback the master with Q and Q’ and the master gives the slave S and R, so pulse the clock, the next state is displayed as shown below.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CLK1** | **J** | **K** |  | **S** | **R** | **Q** | **Q’** | **🡪** | **S** | **R** | **Q** | **Q’** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** |  | **0** | **1** | **0** | **1** | **🡪** | **0** | **1** | **0** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** |  | **0** | **1** | **0** | **1** | **🡪** | **1** | **0** | **1** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** |  | **1** | **0** | **1** | **0** | **🡪** | **0** | **1** | **0** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** |  | **0** | **1** | **0** | **1** | **🡪** | **1** | **0** | **1** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** |  | **1** | **0** | **1** | **0** | **🡪** | **0** | **1** | **0** | **1** |

****

**4.2) Registers:**

**4.2.1)** **Shift Register with D Flip-Flops**

**The register can shift binary digit to lift or right or both sides, to construct this we connected the outputs to the lambs, then we cleared the register by setting B to 1, then we set A to 1 and made some pulses, the labs started to light up one by one, and when we set A to 0 the lambs started to turn off one by one, so we can observe that with each pulse the state of the lamb moved to the next one, so when the first one is active, after the first pulse the next one is activated and because A is set to 1 the first one activated too, and the design is for this register is shown below.**

****

**4.2.2)** **4-Bit Shift Register with serial and parallel load:**

**4.2.2.1)** **Shift- right serial- input:**

**This register shifts the binary by one digit to the right for each pulse, it works as we described before, the MODE clears the register and it enables and disables the register, when the serial input is set to 1 and mode is set to 0, with every pulse from the clock the binary activity is shifted one to the right, in other words, each lamb takes the state of the lab in its lift, and the last one takes the state from the input.**

****

**4.2.2.2)** **Parallel- load Register:**

**This register sets the output based on the inputs from A,B,C, and D, so in the first state the output would be the same as the inputs, each clock pulse the binary digits are shitted as the labs show.**

****

**4.3)** **Counters:**

**4.3.1) 2-bit Synchronous Counter:**

**The 2-bit counter add with each clock pulse 1 to the value that it has before, and we constructed this counter using JK flip-flops by connecting two JK flip-flops together and connect the J line with the K line, this make it work “pretty much” as D flip-flops, then we connect the same clock to all the flip-flops, after that we start applying the pulse. With each pulse it adds one to the value it has (initially 0), and this result is shown because with each pulse it adds changes the state of the first flip-flop, and with each two pulse it changes the state of the second flip-flop.**

**This gives us a counter from 0-3 🡪 00,01,10,11**

**Then it starts over.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLK** |  | **Q1** | **Q0** | **D1** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **1** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **0** | **2** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** | **3** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **1** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **0** | **2** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** | **3** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **1** | **1** |

****

**4.3.2)** **3-bit (divide-by-eight) Ripple Counter:**

**This counter work as the previous one, but in this case the each clock is connected from the clock from the clock of the flip-flop that is before it, and because this counter has three flip-flops is count from 2^3 = 8**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLK** |  | **Q2** | **Q1** | **Q0** | **D1** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **0** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **1** | **1** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **1** | **0** | **2** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **1** | **1** | **3** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **0** | **0** | **4** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **0** | **1** | **5** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** | **0** | **6** |
| **C:\Users\User\Desktop\Capture.PNG** | **1** | **1** | **1** | **7** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **0** | **0** |
| **C:\Users\User\Desktop\Capture.PNG** | **0** | **0** | **1** | **1** |

****

**5) Tasks:**

**5.1) How do we get Serial shift-left using IC7495?**

****

**5.2) 3-bit Synchronous Counter.**

****

**6) DISCUSSION:**

**6.1) Although latches are useful for storing binary information, they are rarely used in sequential circuit design, why?**

Latches are rarely used in sequential circuits because we can't control then since they work in any change in the clock so it is difficult to control them therefore we use flip flops instead.

**6.2) What is the disadvantage of the RS flip flop?**

the disadvantages of RS flip flop is that it has a indeterminate state it is difficult to control and to work with it.

**6.3) What is the difference between “synchronous” and “ripple” counters?**

The synchronous counters there is one main clock entered to all flip-flop on it so the clock is one and entered to all of the flip flops in the counter. The Ripple counter the main clock is applied to the first flip flop and the next flip flop take a clock from the previous one and so on.

**7) Conclusion**

**In this experiment we constructed SR flip-flops, JK flip-flops,**

**D flip-flops, shift left and shift right registers, asynchronous and ripple counters. We constructed the flip-flops and knew how they work so we can construct the counters and the shit registers, because it basically are made of flip-flops, the clock is add so we can move from stage to another, we faced some problems during the experiment in constructing the shit registers, but after the experiment we learned how the shit registers work, so this experiment was very helpful and I think it may benefit us in the future.**