**DAILY ASSESSMENT 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 29-05-2020 | **Name:** | Sheela Golasangi |
| **Course:** | Logic Design | **USN:** | 4AL16EC068 |
| **Topic:** | Analysis of clocked sequential circuits and Digital clock design. | **Semester & Section:** | VIII  ‘B’ |
| **Github Repository:** | Sheela – Course |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of the session**  **C:\Users\india\Pictures\Screenshots\Screenshot (355).pngC:\Users\india\Pictures\Screenshots\Screenshot (359).pngRobot Voice: How to Make Any Circuit Talk** |
| **REPORT**  **Analysis of Clocked Sequential Circuits**   * The digital circuits we have seen so far (gates, multiplexer, demultiplexer, encoders, decoders) are combinatorial in nature, i.e., the output(s) depends only on the present values of the inputs and not on their past values. * In sequential circuits, the “state” of the circuit is crucial in determining the output values. For a given input combination, a sequential circuit may produce different output values, depending on its previous state. * In other words, a sequential circuit has a memory (of its past state) whereas a combinatorial circuit has no memory. * Sequential circuits (together with combinatorial circuits) make it possible to build several useful applications, such as counters, registers, arithmetic/logic unit (ALU), all the way to microprocessors.   **NAND latch (RS latch)**   1. R   A B  1 0  X1 X2  0 1  0 1 1 0  1 1  0 0  previous  invalid  R S  1 0  Q Q  0 1  0 1 1 0  1 1  0 0  previous  invalid  Q  Q  X1     1. X2 S  * The combination *A* = 1, *B* = 0 serves to *reset X*1 to 0 (*irrespective of* the previous state of the latch). * The combination *A* = 0, *B* = 1 serves to *set X*1 to 1 (*irrespective of* the previous state of the latch). * In other words,   + *A* = 1, *B* = 0 *→* latch gets reset to 0.   + *A* = 0, *B* = 1 *→* latch gets set to 1. * The *A* input is therefore called the RESET (R) input, and *B* is called the SET (S) input of the latch. * *X*1 is denoted by *Q*, and *X*2 (which is *X*1 in all cases except for *A* = *B* = 0) is denoted by *Q*. * Now that we have flip-flops and the concept of memory in our circuit, we might want to determine what a circuit is doing. * The behavior of a clocked sequential circuit is determined from its inputs, outputsο and state of the flip-flops (i.e., the output of the flip-flops). * The analysis of a clocked sequential circuit consists of obtaining a table of a diagramο of the time sequences of inputs, outputs and states.   **Analysis Procedure:**   * We have a basic procedure for analyzing a clocked sequential circuit: Write down the equations for the outputs and the flip-flop inputs. * Using these equations, derive a state table which describes the next state. Obtain a state diagram from the state table. * It is the state table and/or state diagram that specify the behavior of the circuit. * Notes: The flip-flop input equations are sometimes called the excitation equations. * The state table is sometimes called a transition table.   **Analysis of Clocked Sequential Circuits (with D Flip Flop)**  **Steps:**  **Step 1:** Find out the input and output equation.  C:\Users\india\Pictures\Screenshots\Screenshot (360).png  **Step 2:** State table.  C:\Users\india\Pictures\Screenshots\Screenshot (361).png  **Step 3:** State Daigram.  C:\Users\india\Pictures\Screenshots\Screenshot (362).png    **Digital Clock Design**  https://2.bp.blogspot.com/-0o7dZ2ojHLQ/WFwAroSjIJI/AAAAAAAAAVo/Shw5HyskPOAtzQhu6gLwRNRihSMfCV5nACLcB/s1600/15151018_1079803105450735_2141653692_n.jpg  The main parts of the circuit are as follows:   1. **Timer 555**: Responsible for generating the clock pulses for the counters, the frequency of the output should be 1Hz which means 1 second for each pulse. 2. **Counters**: Responsible for generating the time in BCD (Binary Coded decimal). 3. **Decoders**: Takes the BCD of the counter as input and produces 7 segment output. 4. 7 **segments**: Displays the time, of course!  * Seconds have 2 displays, 2 decoders and 2 counters. The same for minutes and hours. * One thing you should know before explaining how the circuit works is that 7490 decade counters respond only to the negative going edge of 555 pulses, which means it will change its state only when the clock goes from 1 to 0.   **The circuit works as follows :**   * 555 timer produces 1 second pulses to the clock input of the first counter which is responsible the first column of seconds, so its output will change every second. * The counter produces numbers from 0 to 9 in BCD form and automatically resets to 0 after that. * So the output of the first counter will count from 0 to 9 every second and that's exactly what we want from it, so we are done here. Let’s move to the next one.   **What do we want here?**  **First**, we want the 2nd counter to start counting when the 1st one moves for 9 to 0 (that makes 10 seconds!)  **How can this be done?**  let's check the output of the fist counter in BCD :  MSB---LSB  0:  0000  1:  0001  2:  0010  3:  0011  4:  0100  5:  0101  6:  0110  7:  0111  8:  1000  9:  1001  0:   0000  Remember that 7490 decade counters respond only to the pulses that go from 1 to 0 and notice that this case only happens in the BCD code above when the output changes from 9 to 0 (the Most significant bit changes from 1 to 0). So, we'll just connect the clock input of the 2nd counter to the most significant bit of the output of the first counter.  **Second**, Since we have 60 seconds in the minute we want the 2nd counter to count only to 5, that makes 59 seconds maximum, when it take another pulse it doesn't count to 60, instead it resets itself to 0 and send a pulse to the first counter in minutes to tell it to count 1 minute  **How can this be done?**   * From the BCD code above (6: 0110) when the output is 6 the two middle bits are 1 (Q1,Q2), * So By ANDing these two bits the output will be 1, This output will be connected to the reset pin of the same counter (2nd one) and the clock input of the next one (3rd). * When the output is 6 then AND gate output (1) will reset the same counter and its outputs goes 0000 so the output of the AND gate again goes to 0 (1---->0), that will clock the next counter. Beautiful!   **Notice that the output of the counters are named : Q0 , Q1 , Q2 , Q3**   * The 4th counter will be the same as the second one so we are clocking it using the Most Significant Bit of the output of the previous one. * Again, the 5th counter is the same as the 3rd one and takes its clock from the AND gate. * The 5th and the 6th counters are responsible for hours so they are  limited to 23, and resets themselves to 00 when the 5th counter is 4 and the last one is 2 (24). * This is done using and gate with Q2 (3rd bit) of the 5th counter as one input and Q1 (second bit) of the last counter as the other input, and the output of this AND gate will be connected to both resets of the last 2 counters. * When the last counter is 0(0000) or 1(0001), Q1 which is one of the inputs to the AND gate will be 0 so the output of the AND gate will be zero. when it counts to 2 this bit will be 1 so the output of the and gate will depend on the the other input which is Q2 of the previous counter, and this bit will be zero until it reaches 4 (0100),So, the output of the and gate will be 1 (0--->1) resetting both counters to 00. * The output of these counters are converted to 7 segment output using 7447 decoders, then to the 7 segment, we won't get into the details of their datasheets.   Learning Sequential Logic Design for a Digital Clock  * This instructable is for two purposes 1) to understand and learn the fundamentals of sequential logic 2) use that knowledge to create a digital clock. * Digital clocks have been built by countless electronics hobbyists over the world. So why have I chosen to implement that? Well usually clock circuits available on the internet (all circuits I have seen) use the 7490 counter (I have used 7493 but I will show why), microprocessors or Arduino boards. But not all of us have the means to buy microprocessors or Arduino boards (as far as I am concerned they are expensive). I wanted to try a different circuit for the same clock and I also chose it because it requires a lot of counters, and counters are based on sequential logic. When I say digital clock, you should expect something like the one in the picture! * It's my stand that just looking at the circuit diagram and replicating it on a bread-board is not what electronics is about. Almost all digital circuits from traffic lights etc. to even computers are all based on sequential logic (its importance). Therefore, I have included the theory of flip-flops and sequential logic design in hope that it would help the reader to design circuits of their own. * I have done my best to explain sequential logic design between Step 3 to 8. However, if all you want is the clock, then please skip everything from step 3 to step 8. But if you go through those steps you will understand how to work with flip-flops etc. and though it takes a lot of time I can assure that you will be left with a wealth of knowledge. * Usual clocks based on decade counters have a hour counter from 0 -23. I have only used IC's but still got a 12 hour clock, which I have not seen elsewhere. I have also added a small alarm module. The alarm is again achieved using IC's not by programming boards (which quite frankly are comparatively easy). It is not much but I did whatever extra nicks I could do. The main emphasis however, is learning sequential logic and developing a breadboard based clock using that knowledge. |