# Simulations of Hexadecimal Counters via the 74LS161 Integrated Circuit – Applications of Visual-Based Multisim and Verilog Projections

*ECE230 Honors Option - Xiangbo Cai*

## 1. Introduction

In everyday life, the increasingly popular need for higher efficacy chip characteristics drives innovation in digital circuitry, particularly in the realm of counters. This report aims to embark on a comprehensive exploration of the practical applications and operational intricacies of a crucial component within this domain. The 74LS161 counter chip, especially in the realm of counters, is a very vitally relevant chip in digital circuits. This report mainly discusses and studies the relevant characteristics of 74LS161 enhanced with hexadecimal counting.

It introduces the method of analyzing the operations of the integrated counter 74LS161 and representing the physical manifestation of these operations with simulation software like NI Multisim. Additionally, the logic analyzer in Multisim, which can intuitively, "using different design methods," describe the working process of the counter and display various input and status output signals of the counter in both a synchronous and multi-track manner [1].

Furthermore, the usage of Vivado's Verilog modeling software to simulate the underlying circuit logic of the 74LS161 chip allows for a clearer model in the electronic efficacy of this counter implementation. By assigning a testbench to test the corresponding output results and simulate the waveforms, comparison of the simulated waveforms and logical outputs verify increased efficacy in the implementation of hexadecimal counters.

Overall, in testing the operational intricacies of the 74LS161 chip, by providing more refined ideas of practical implications of enhanced chip characteristics, this report aims to contribute to the advancement of digital circuit design. Its applications in alternative counting, such as the hexadecimal system, are vital to creating more practical and efficient electrical systems.

## 2. Binary Counter

### *2.1 Introductory Logic regarding counter*

According to binary addition principles, when adding 1 to a multi-digit binary number, if all the bits below the $i-th$ bit are 1 (from $i-1, i-2, \dots, 1$), then the $i-th$ bit will change state (from 0 to 1, or from 1 to 0). The state of the lowest bit changes every time 1 is added. The following diagram shows a state:

$$1\,0\,1\,1\,\mathbf{0}\,\mathbf{1}\,\mathbf{1}$$

$$+\qquad\qquad 1$$

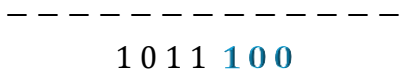


**Figure 1**: Based on the above explanation, it can be seen that the lowest three bits change.

Synchronous counters are typically constructed using $T\ flip-flops$, and there are two main structural forms. One form controls the state of the $T$ input terminal. When each $CLK$ signal (counting pulse) arrives, the flip-flops whose inputs at the control terminal $Ti$ should be flipped are set to $Ti = 1$, while those that shouldn't be flipped are set $Ti = 0$. Another form controls the clock signal; when each counting pulse arrives, it is only applied to the $CLK$ input terminals of the flip-flops that should be flipped, while those that shouldn't be flipped remain unchanged. At the same time, all flip-flops are set to $T = 1$. In this way, the different states of the counter circuit can be used to record the number of inputs $CLK$ pulses.

From this, it can be seen that when controlling through the T terminal's state, the logic expression for the input terminal of the $i-th$ flip-flop should be:

$$T_i = Q_{i-1}\ Q_{i-2} \dots Q_1\ Q_0$$

$$= \prod_{j=0}^{i-1} Q_j \quad (i = 1, 2, \dots, n-1)$$

*2.2 Further Background Information with the 74LS161 counter chip*

The 74LS161 is a synchronous 4-bit counter IC (Integrated Circuit) from the 74LS family of TTL (Transistor-Transistor Logic) chips. It is designed to count up or down in binary increments, depending on the logic levels applied to its control inputs. 74LS161 can reach the functionality of a 4-bit hexadecimal counter.

**Table 1**: Table Representation of input and output types of the 74LS161 chip.

| **Input** | | **Output** | |
|---|---|---|---|
| clk | time clock, period happen | tc | carry out value |
| clr | clear value, reset to all 0 | Q [3:0] | count output number |
| ld | enable to load number | | |
| EP | work control 1 | | |
| ET | work control 2 | | |
| P [3:0] | user input initial value | | |

In practical production of counter chips, additional control circuits are often added to increase the circuit's flexibility. The designed circuit should not only have the function of a binary adder counter but also have additional functions such as preset number, hold, and asynchronous clear. In the diagram, "ld" is the preset control terminal, "P [3:0]" is the input terminal, "tc" is the carry output terminal, "clr" is the asynchronous clear terminal, and "EP" and "ET" are the operating state control terminals. The table below shows the function table of the 4-bit synchronous binary counter 74LS161.

**Table 2**: Functions Table for the representation of the 4-bit synchronous hexadecimal 74LS161

| clk | clr | ld | EP | ET | Output State |
|---|---|---|---|---|---|
| × | 0 | × | × | × | Set to 0000 |
| ↑ | 1 | 0 | × | × | Set to initial |
| × | 1 | 1 | 0 | 1 | Not changed |
| × | 1 | 1 | × | 0 | NC but tc=0 |
| ↓ | 1 | 1 | 1 | 1 | Counts |

### 3. Simulation of 74LS161 on Multisim

Now, the simulation of the 74LS161 integrated circuit can be applied to the conceptual information. By using counting-base types like hexadecimal, “in many applications, [these] counters can be designed using these available ICs. In case any other sequence of count or modules is required, then the circuit *can* be designed. These methods are validated through experiments and, if correct, in practice can be used directly” [2]. In accordance with the Part II table form, it can be easily summarized that the below result, which is as follows: When clr=0, asynchronous clear occurs. When clr=1 and ld=0, synchronous load occurs. When clr=ld=1, and EP=ET=1, shows addition counting that is performed in binary two's complement. When clr=ld=1, and EP*ET=0, the counting state remains unchanged. Sequentially, it is natural that the generation the Multisim graph to simulate the above result should further test the desired efficacy increase expectations.

As can be seen from the figure below, the K1 is linked with the clr, and K2 is linked with the ld. In Figure 2, via a Multisim graph, the types clr=0 and ld=0, and our output=0 are set.

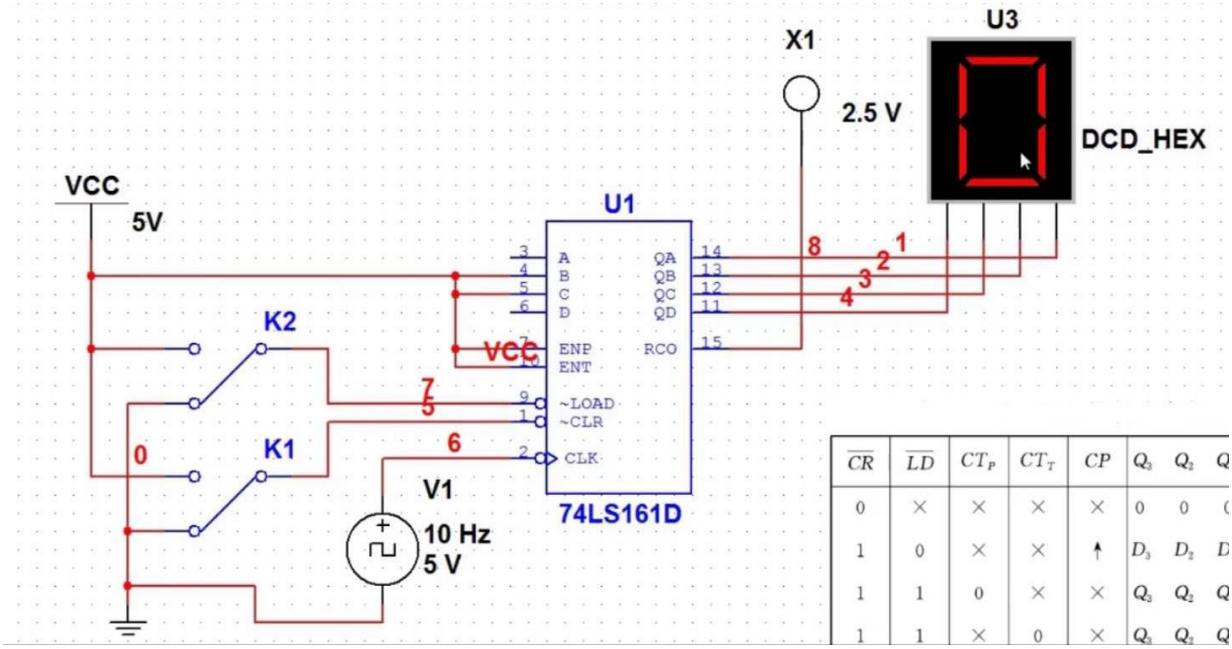


**Figure 2**: When clr=ld=0, the output are set to be 0

In Figure 3, the types clr=1 but ld=1 are set so that the output vector Q loads the input vector P, and, as the initial input vector P in the graph is set to be 6, the output should be 6. The resulting Multisim figure is shown below:

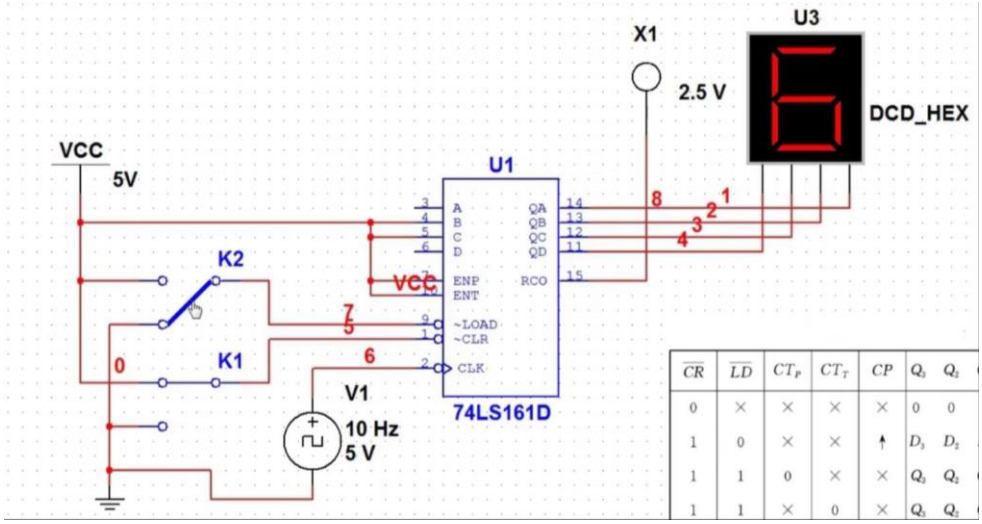


**Figure 3**: When clr =1 but ld =0, the output Q =D (in this case, D initial to be 6)

When the EP and ET are also set to 1, then the 74LS161 will start to count. As this is a dynamic process, it is relatively difficult to show the representative animation here. However, the exact moment when the output result increments to 8 during the second cycle is captured below.

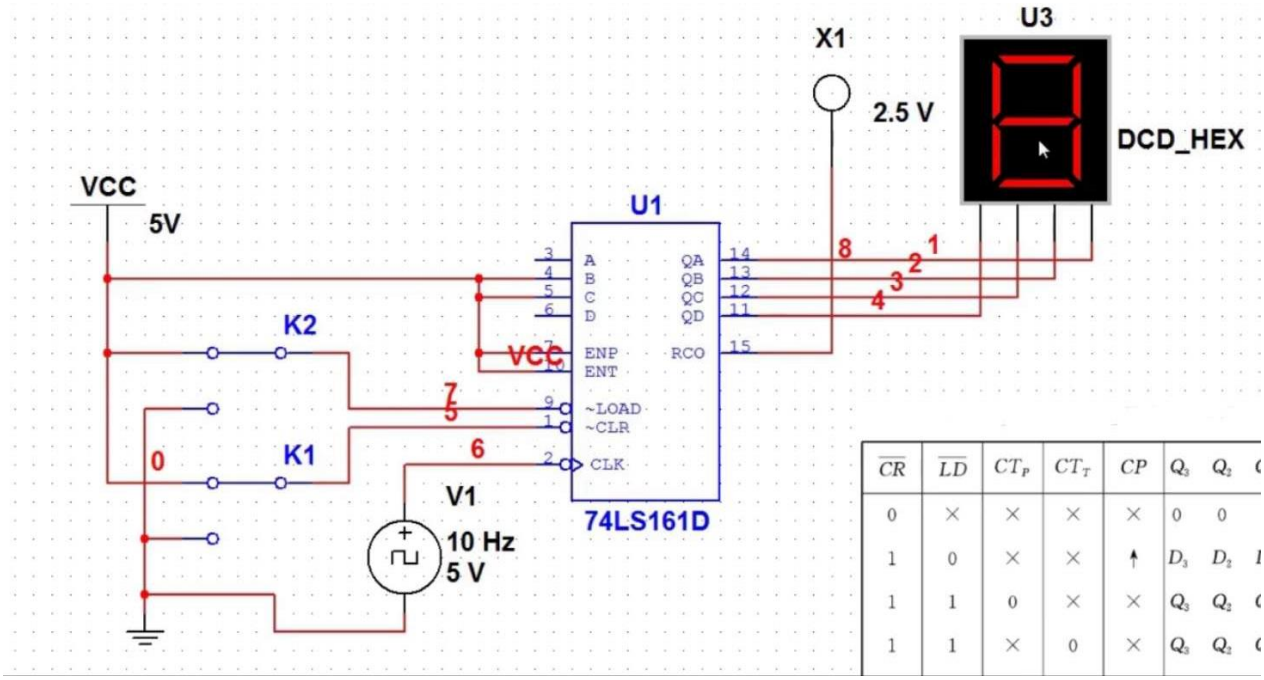


**Figure 4**: When clk=ld=1, and ET=EP=1, output start to count

When the output result reaches F (16 in decimal), the carry output TC will flicker, indicating that the part is about to carry once. Then, the output P [3:0] will return to 0 and continue counting, while TC will flicker for an extremely short period (becoming 1), and then it will extinguish (becoming 0) again. The result as shown below:

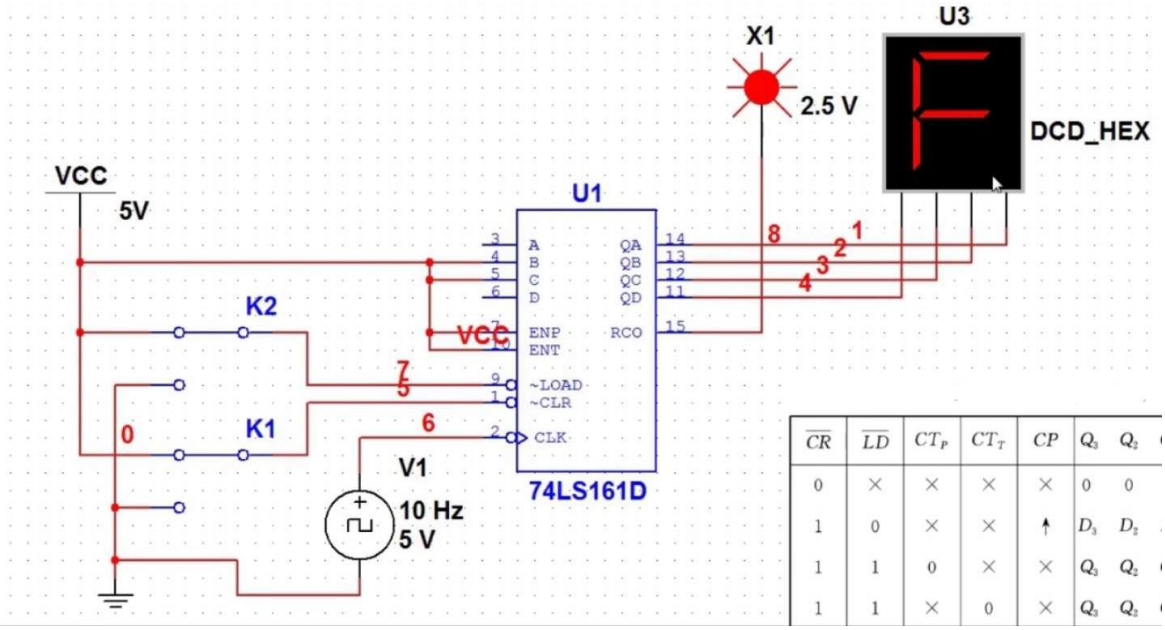


**Figure 5**: When clr = ld = 1, and EP=ET=1, and Q reach highest F(15), tc flips to 1

## 4. Simulation of 74LS161 on Verilog code

Utilizing the above principles and discovered ideas, the underlying logic of the 74LS161 chip can be accurately transformed into corresponding Vivado Verilog code for implanted simulation. In the module design, we include input ports clk, clr, ld, an input vector p [3:0], output ports tc, a carry-out display bit, and output vector q [3:0]. Based on the logic summary of the 74LS161 chip in the second part, as well as a basic understanding of Verilog basic systems for coding, the corresponding circuit code can be designed as follows.

In the following Verilog, cet can be used to represent the ET and cep for EP. As an alternative to the standard, the type clr is set as high active instead of negative active.

```
<Verilog>

// 74LS161 chip Verilog code
module ttl_74161(
    input ld,
    input [3:0] p,
    output reg[3:0] q,
    input cet,
    input cep,
    input clk,
    output tc,
    input clr
    );

    assign tc = cet & (&q);
```

```
    initial begin
        q <= 0;
    end

    always @ (posedge clk or negedge clr) begin
        if (clr) begin
            q <= 0;
        end else begin
            if (ld == 1) begin
                q <= p;
            end
            if (cep & cet) begin
                q <= q + 1;
            end
        end
    end
endmodule
```

**Figure 6**: Verilog Code of 74LS161 Module Design Part

According to the output results, three groups of test data were set up—corresponding to the clear, load, and counting states, respectively. The following is the related testbench code.

```
<Verilog>
// Testbench of the 74LS161
module ttl_74161_tb;

    // Inputs
    reg ld, cet, cep, clk, clr;
    reg [3:0] p;

    // Outputs
    wire [3:0] q;
    wire tc;

    // Instantiate the module under test
    ttl_74161 uut (
        .ld(ld),
        .p(p),
        .q(q),
        .cet(cet),
        .cep(cep),
        .clk(clk),
        .tc(tc),
        .clr(clr)
    );

    // Clock generation
    always begin
        #5 clk = ~clk;
    end

    // Reset
    initial begin
        ld = 0;
        p = 4'b0000;
        clk = 0;
```

```
    clr = 0;
    cep = 0;
    cet = 0;
  end

  // Test scenarios
  initial begin
    // Scenario 1: clr = 0
    #15 clr = 1;
    #15 clr = 0;

    #15 p = 3;
    // Scenario 2: clr = 1, ld = 1
    #15 ld = 1;

    // Scenario 3: clr = 1, ld = 1, cep = 1, cet = 1
    # 15 cep = 1; cet = 1;

    // End simulation
    #350 $finish;
  end

endmodule
```

**Figure 7**: Verilog Code for the 74LS161 Testbench

Now using the code above, the result can be simulated as such below:

Figure 8 depicts the initial state where clr is set to 1, causing the output q [3:0] to be 0000 in binary, as indicated by the 4b prefix. This state is clearly highlighted by the pink line and the light bar in the accompanying photo, emphasizing the clarity of the representation.

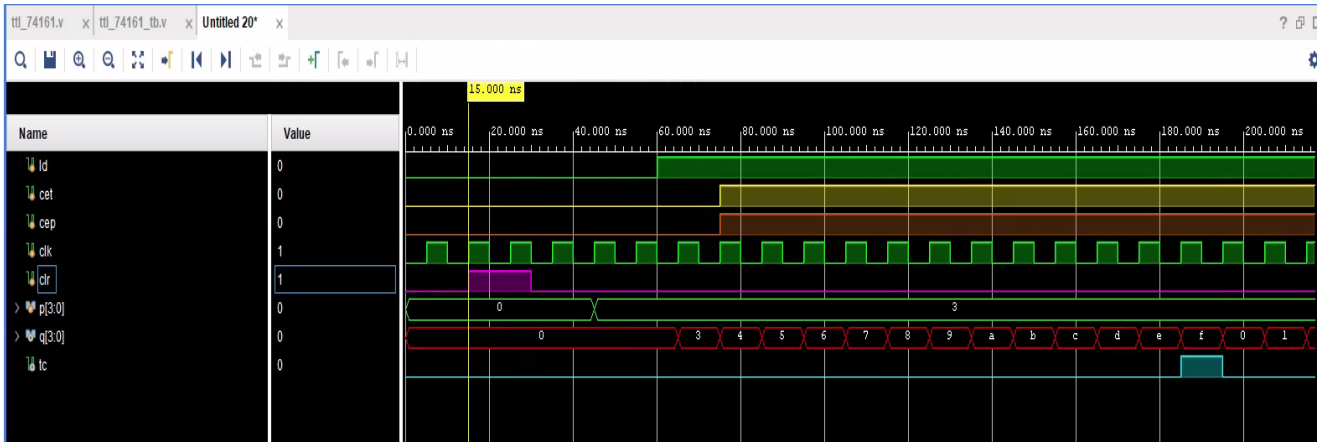


**Figure 8**: When clr=1, all value is set to be zero

Figure 9 shows the load value, when active high (equal to 1), causes the input value p [3:0] to be loaded into q [3:0]. This behavior is clearly observable in the highlighted red vector and light bar section of the diagram. As depicted, when the clock signal (clk) experiences a positive edge, the red vector begins to store the input p [3:0] vector value.

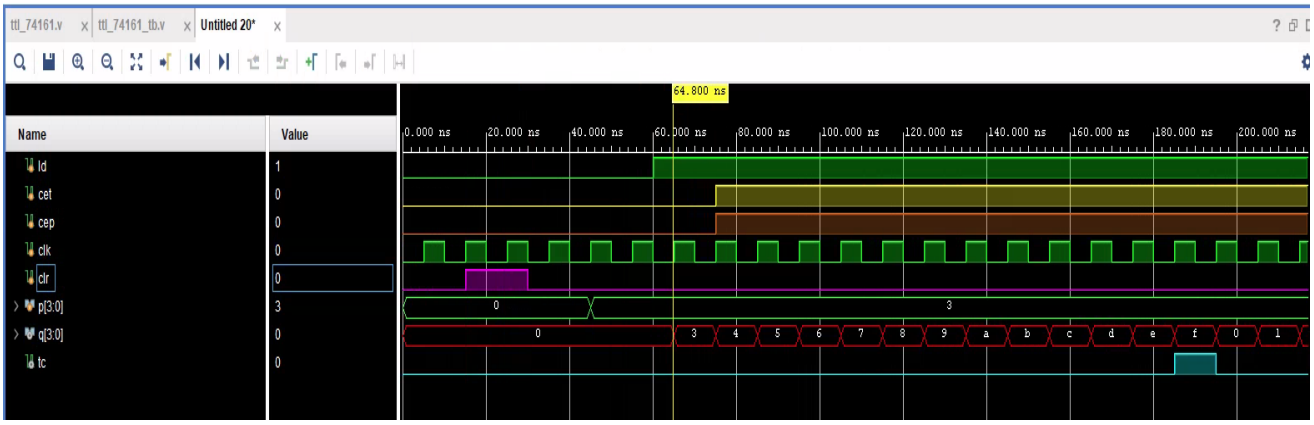


**Figure 9**: When ld=1, the input p value 3 is loaded into output q [3:0]

In Figure 10, activating both control inputs, cet and cep, to 1 initiates the count-up sequence of the output value q [3:0] in hexadecimal. This behavior is clearly demonstrated in the diagram, indicating that when cet and cep are both high, the counting operation begins.

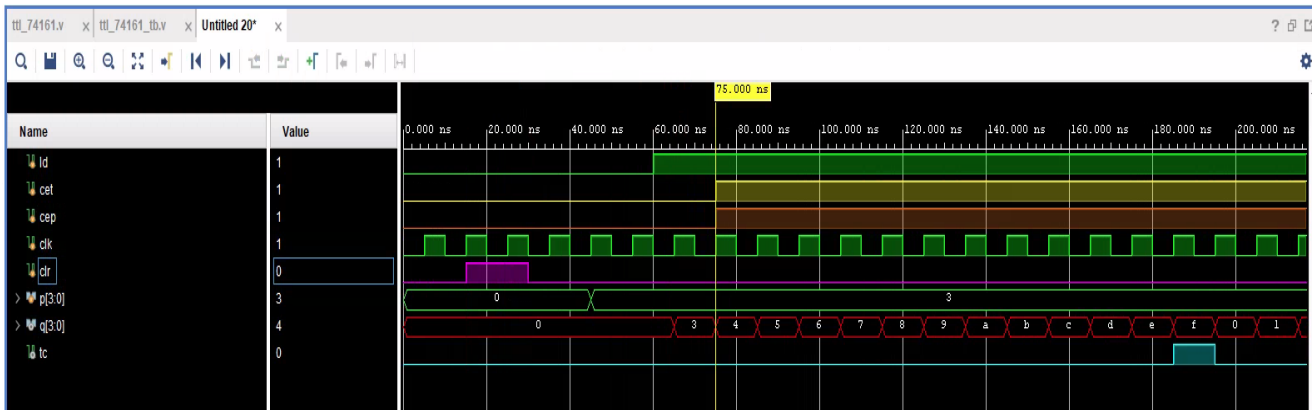


**Figure 10**: When cet(ET)=cep(EP)=1, output q start to count up

When the output vector q [3:0] reaches 4b'1111, we set the terminal count wrap (tc) to be 1, and q [3:0] will reset to 4b'0000 and restart, and tc back to 0. As can be seen in Figure 11, the blue bar indicates the tc output situation.

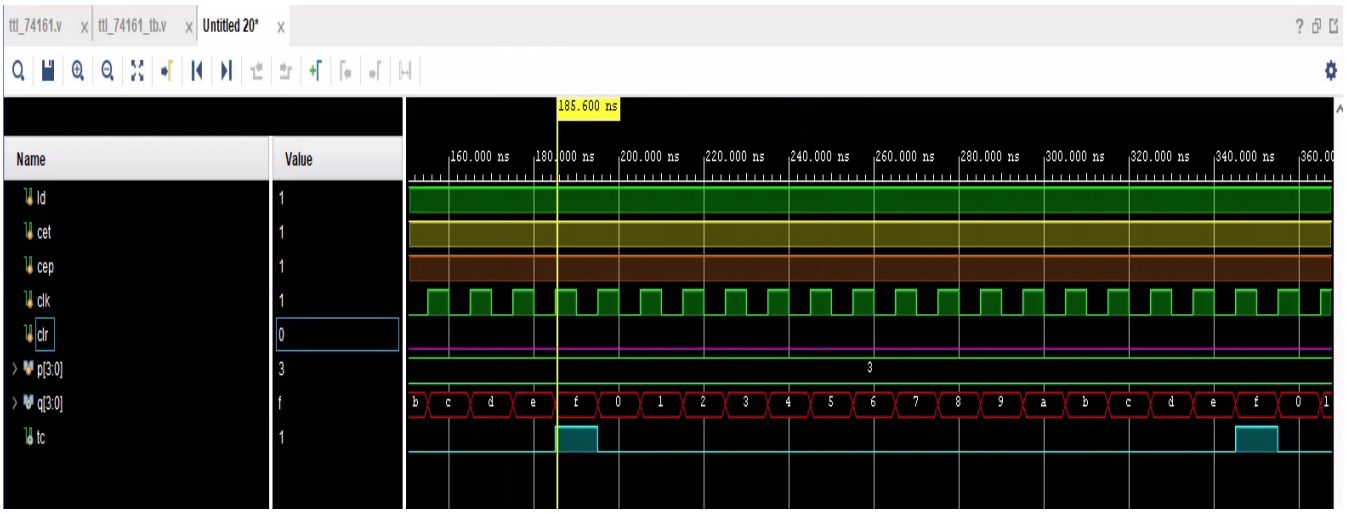


**Figure 11**: ★ When output vector q=4b'1111, tc flip to 1 and carry in

## 5. Conclusion

As has been demonstrated, this report mainly focuses on the study of hexadecimal counter implementation in counter-based systems like the 74LS161 IC. By examining the logic of the 74LS161, the real model of the 74LS161 is transformed into a fully-fledged digital circuit model at the macro level. Simultaneously, for the underlying logic of the 74LS161, conversion of the chip results in Verilog code that relevantly embodies the logic. This Verilog code was then compiled and simulated alongside the Multisim model, producing the desired waveform. This waveform is not only in line with the digital circuit model of the standard Multisim digital circuit, but also clearly demonstrates increased time response in, like hexadecimal, more complex counting-based systems.

Further relating to the modern world, through various digital logic applications and electronic devices, the applications of 74LS161 are very extensive [3]. Counters, steppers, frequency dividers, pulse sequence generators, and automatic control systems can all be made more efficient and practical. The use of these circuits is becoming increasingly common and continues to benefit the progression of electronic-counter-system technology. Understanding how to better implement these circuits, like 74LS161, is essential for us to have a deeper understanding of digital circuits and improve our increasingly digital world.

## 6. References

[1] L. Li, L. Meng, and F. Wang, “Design and simulation of frequency divider circuit based on multisim,” E3S Web of Conferences, vol. 268, p. 01058, 2021. doi:10.1051/e3sconf/202126801058

[2] J. Zhao, “The applications of MSI Counter — 74X161,” 2011 International Conference on Electrical and Control Engineering, Sep. 2011. doi:10.1109/iceceng.2011.6057822

[3] D. Bao, H. Guo, X. Bao, and L. Tao, “Research and practice of blended teaching of digital electronic technology course for application- oriented undergraduate,” 2021 2nd International Conference on Information Science and Education (ICISE-IE), Nov. 2021. doi:10.1109/icise-ie53922.2021.00075