

Lab 11 – Digital Dice

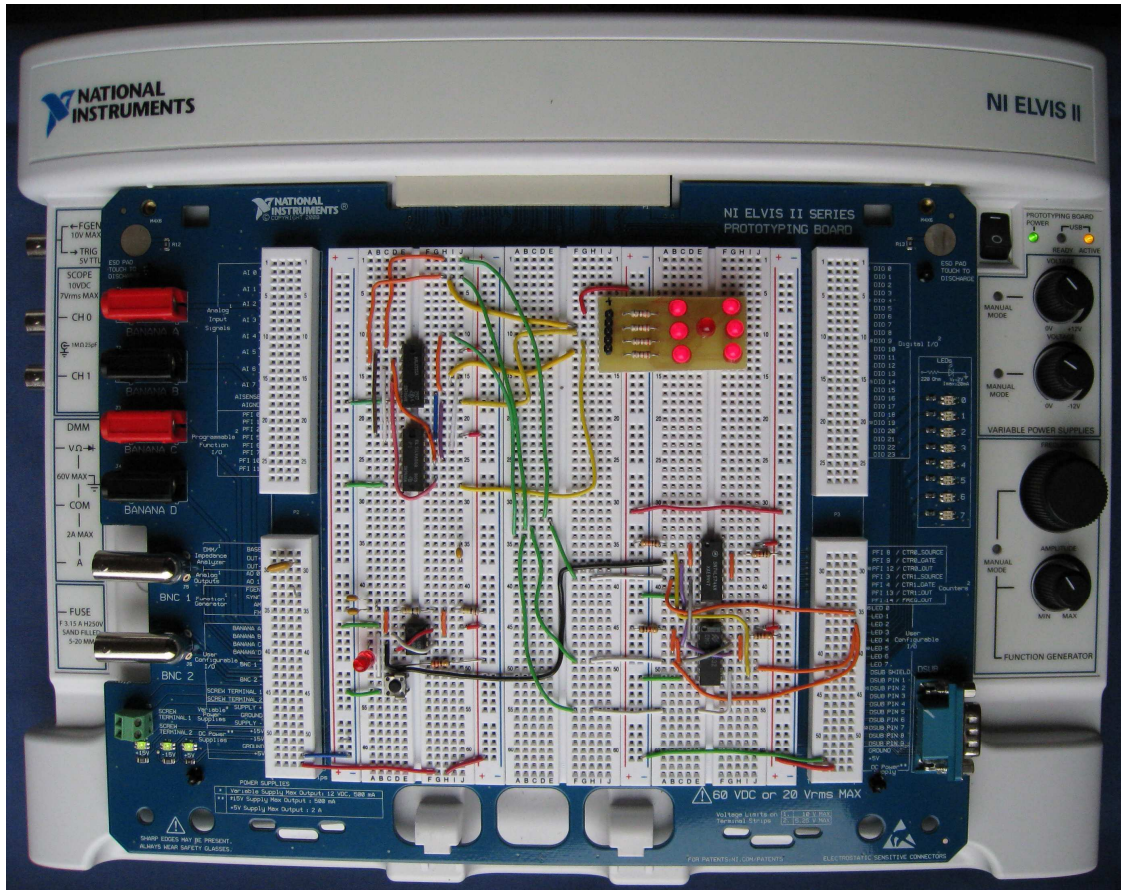


Figure 11.0. Digital Dice Circuit on NI ELVIS II Workstation

From the beginning of time, dice have been used for games of chance. Cubic dice similar to modern dice date back to before 5000 BC. The Greeks and Romans used dice made of stone or ivory with spots on the side inlaid with dark ink or bits of lead. In this lab, light emitting diodes (LEDs) in a dice pattern are used to display the lucky number.

Goal: In this lab, explore using Multisim to design an electronic version of digital dice with standard TTL ICs. From the virtual Multisim design, build a prototype circuit using real TTL ICs on the NI ELVIS II protoboard. Also use NI ELVIS II SFPs to test your design within a Multisim environment and, with a click of the mouse, use the same test instruments to test the real circuit.

Soft Front Panels (SFPs)

DMM[V, Ω , C]

Oscilloscope (Scope)

Function generator (FGEN)

Required Components

Two 100 k Ω resistor (brown, black, yellow)

Five 330 Ω resistors (orange, orange, brown)

Four 1 k Ω pull-up resistors (brown, black, red) [optional]

Eight red LEDs

0.1 μ f capacitor

Two 1 μ f capacitors

555 timer IC

Two 7474 Dual D flip-flops

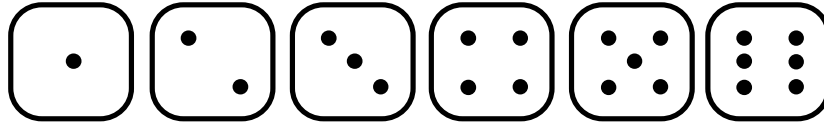
1 7404 hex inverter

Single-pole single-throw (SPST) switch or push button (normally open)

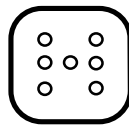
Either And or Nand Logic gates

Some Background

Consider the case of a single die. On each of its six sides, one of the following patterns appears, representing the numbers one through six.

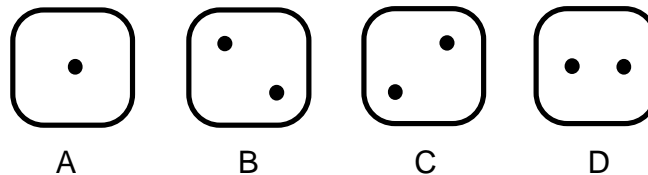


These patterns are traditional. You can think of them as seven lights arranged in an “H” pattern.



By turning on the appropriate lights, you can create any one of the six patterns on the face of a die.

On closer inspection, there are only four unique patterns from which the pattern for any face can be formed. Call these base patterns A, B, C, and D:



The base state A has only one light and it occupies the snake-eye position in the dice’s center position. The three remaining states all use two lights occupying one of the two diagonal positions or the horizontal position.

If you write down a truth table for the presence or absence of these base patterns as a function of the die face number, the meaning of the base states becomes clearer.

Die face	A	B	C	D
1	√			
2		√		
3	√	√		
4		√	√	
5	√	√	√	
6		√	√	√

Table 11.0. How Base States form Dice Patterns

The base pattern A is used by all of the odd numbers (1, 3, and 5). Pattern B is found in all numbers except 1. Base pattern C is found in the numbers 4, 5, and 6. Pattern D is used only when representing the number 6.

You can build a single dice display using seven LEDs and four current limiting resistors.

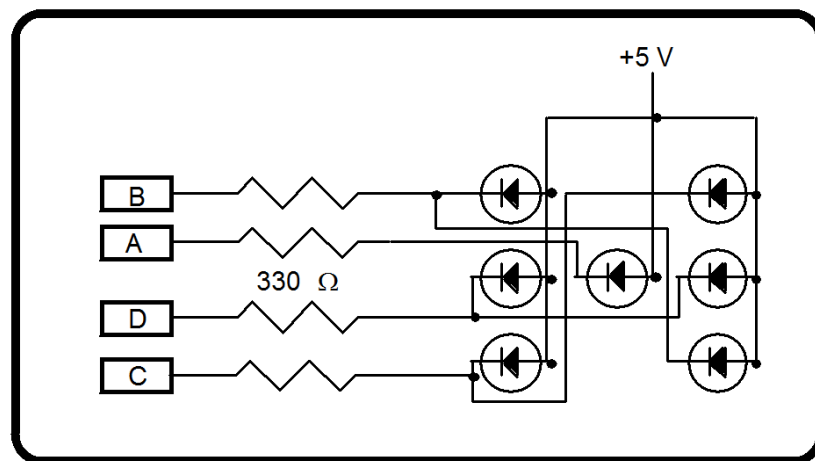


Figure 11.1. Dice Display Circuit

In the Figure 11.1 schematic diagram, shorting an input pad A, B, C, or D to ground turns on that base state. In the combinations of A, B, C, or D shown in Table 11.0, you can generate any of the standard dice displays.

Exercise 11.1: Multisim Dice Display Using Seven LEDs

Launch Multisim and open the program **Display1.ms11**. It has seven virtual LEDs and four current limiting resistors together with four switches. Run this program and operate the switches in the combinations shown Table 10.0 to verify that the truth table generates the dice displays.

Replace the switches with the NI-ELVISmx Digital Writer SFP.

For reference, look at **Display2.ms11**.

By doing this, you can exercise the dice display using the NI-ELVISmx Digital Writer. Use the mapping

Line 0 → A
Line 1 → B
Line 2 → C
Line 3 → D

Set all the lines 0..1 to LOinitially. All the LEDs should be lit. This is the lamp test state.

Note that the LEDs are active low, that is, setting the lines (A..D) LO turns on the selected base set of LEDs.

Verify your circuit design follows the truth table in Table 11.0.

End of Exercise 1

Exercise 11.2: Converting a Multisim Design into a Real Circuit

Complete the following steps to build and test a real dice display.

1. On the NI ELVIS II protoboard, build a real digital dice display using seven LEDs and four resistors. Use the schematic diagram in Exercise 11.1 or the program Display2.ms11 as a wiring guide.

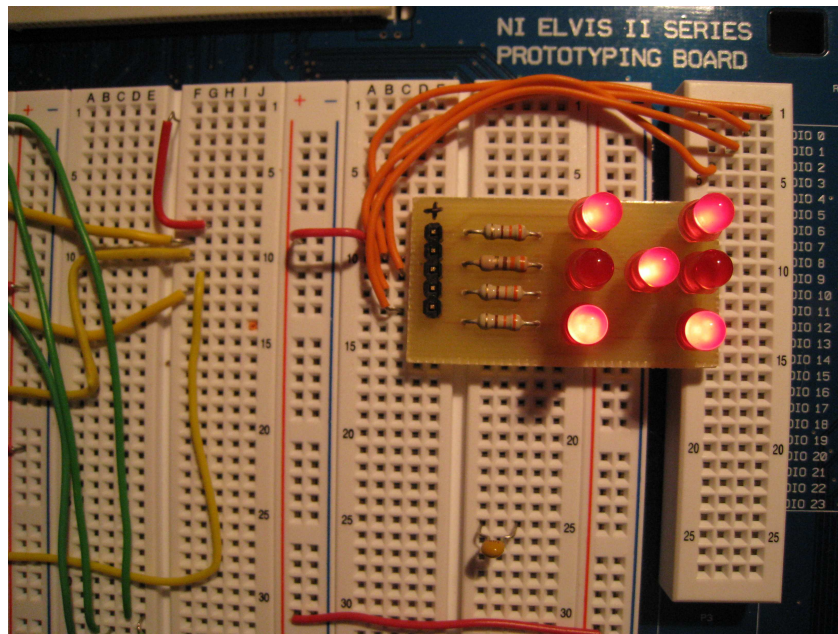


Figure 11.2. Digital Dice Module on NI ELVIS II Protoboard

2. Use the NI-ELVISmx Digital Writer as a diagnostic test instrument to stimulate your circuit. Recall that the LEDs are active LO.
3. Use the values A, B, C, and D in Table 11.0 to verify the operation of your design.

NOTE: You can have both Multisim and the NI ELVIS protoboard active at the same time. In this exercise, the instrument control device [box] determines which application NI-ELVISmx Digital Writer controls.

[Multisim] selects the active Multisim panel

[Dev 1 (NI ELVIS)] selects the NI ELVIS II protoboard

By switching back and forth, you can easily compare your Multisim design with the real circuit on the NI ELVIS II protoboard.

End of Exercise 11.2

Exercise 11.3: A Modulo 6 Counter

A modulo 6 counter is any counter with six unique states that repeat the sequence every six counts. It could be a binary counter with decimal equivalents of (1, 2, 3, 4, 5, 6, 1..) or, as in a LabVIEW or C+ program, counting from zero (0, 1, 2, 3, 4, 5, 0..). It could be a pseudorandom number generator that randomly generates a six-number sequence like (5, 1, 6, 4, 3, 2, 5..).

In this lab, build a simpler modulo 6 counter using a three-element shift register configured as a switched tail ring counter Figure 11.3.

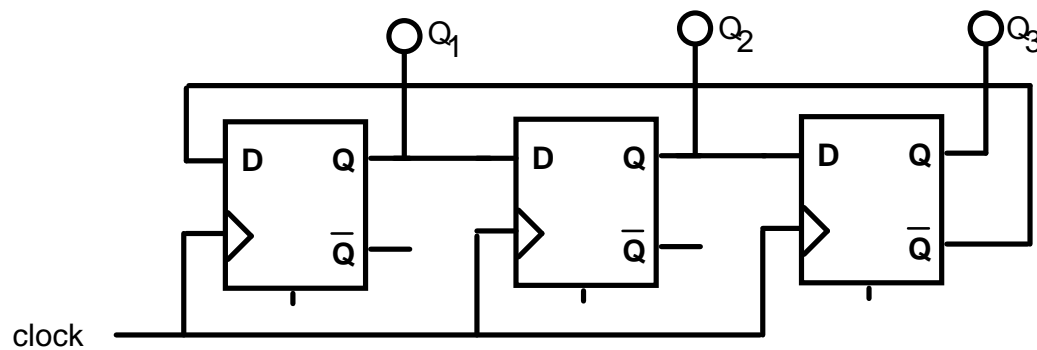


Figure 11.3. Schematic Diagram of a Modulo 6 Switch-tail Ring Counter

It uses three D-type flip-flop elements configured as a shift register. Note the switched tail; the inverted output of the last element is connected back to the input of the first element. On command from a rising edge of the clock pulse, all the D inputs shift to their respective outputs. Each clock pulse generates a new sequence of outputs on Q1, Q2, and Q3, which repeats after six pulses.

The truth table is shown in Table 11.1.

Cycle	Q ₁	Q ₂	Q ₃
1	0	0	0
2	1	0	0
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	0	0	0

same as cycle 1

Table 11.1. Modulo 6 Truth Table

With Multisim, design a modulo 6 counter using two 7474 Dual D edge-triggered flip-flop ICs. Each IC contains two D flip-flop elements labeled in Multisim as 1 and 2. The input is labeled as D for data source and the outputs are labeled as Q and \bar{Q} where \bar{Q} is the complement of Q. Each element has a Clear (CLR) and a Preset (PR) input, which is pulled HI for clocking operations. You form a shift register when you connect all the clock inputs (CLK) together and to the circuit clock.

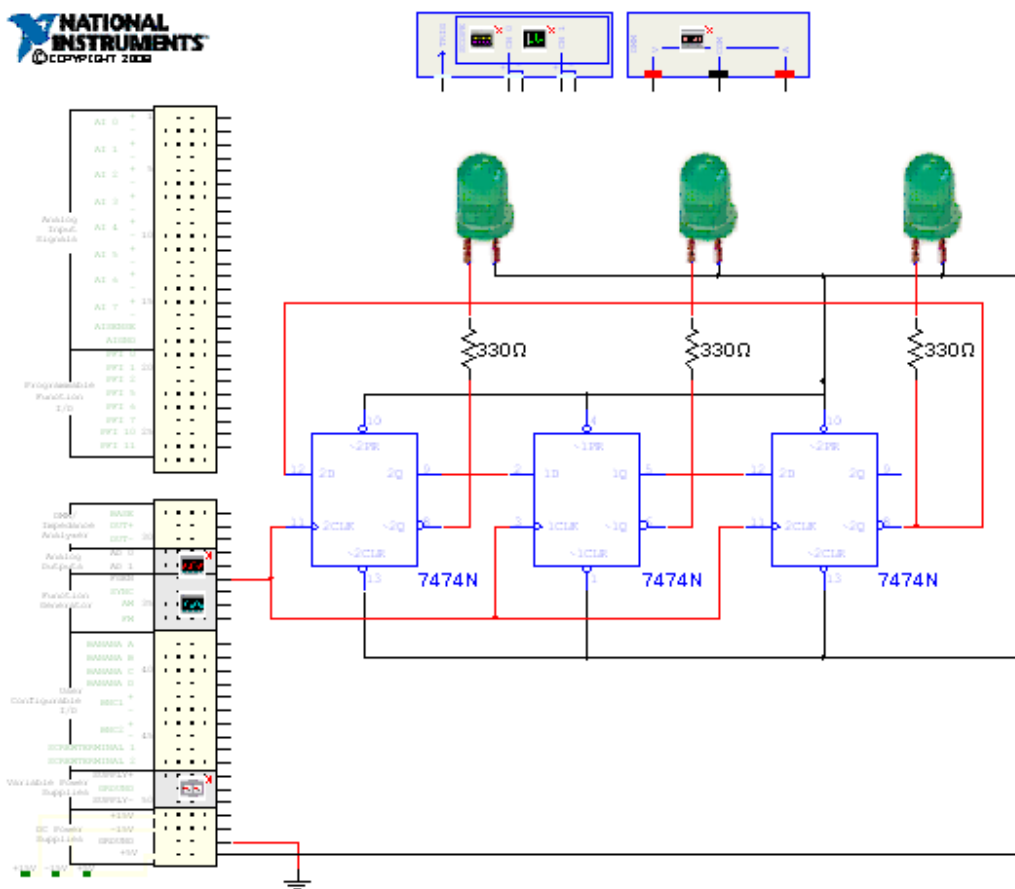


Figure 11.4. Multisim Version of a Switch Tail Ring Counter with LED Displays

To visualize the outputs Q1, Q2, and Q3, an LED is connected to each output (\bar{Q}) and the power supply. Note when a Q bit is HI, current is sunk through \bar{Q} to ground, which turns on the respective LED. The illuminated state signifies that Q is HI.

Use the NI-ELVISmx Function Generator as the circuit clock. Set the output for TTL levels, set the amplitude equal to 5 V_{pk-pk}, and set an offset of 2.5 V. Set the function to

Square Wave and frequency near 60 Hz. Also try setting the frequency to around 2 Hz so that the pattern can be observed and checked.

Verify your design by comparing it with the example Mod 6.ms11 in the NI ELVIS program library.

End of Exercise 11.3

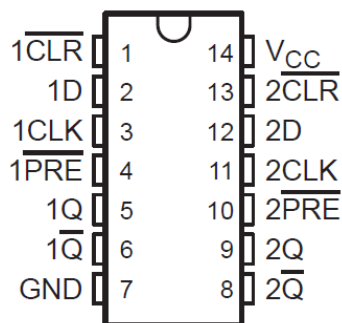
Exercise 11.4: Convert the Mod 6 Multisim Design into a Real Circuit

Complete the following steps to convert your Multisim design to a real circuit on an NI ELVIS II protoboard:

1. Transfer your working design into real circuit components. You need two 7474 ICs, three visible LEDs, and three current limiting resistors.
2. One can also use three of the LED indicators on the NI ELVIS II protoboard labeled LED 0, 1, and 2.

Map Q1 to LED0, Q2 to LED0, and Q3 to LED0.

Note that these LEDs are active HI, that is, a HI state turns the LED on.



7474 Pinouts

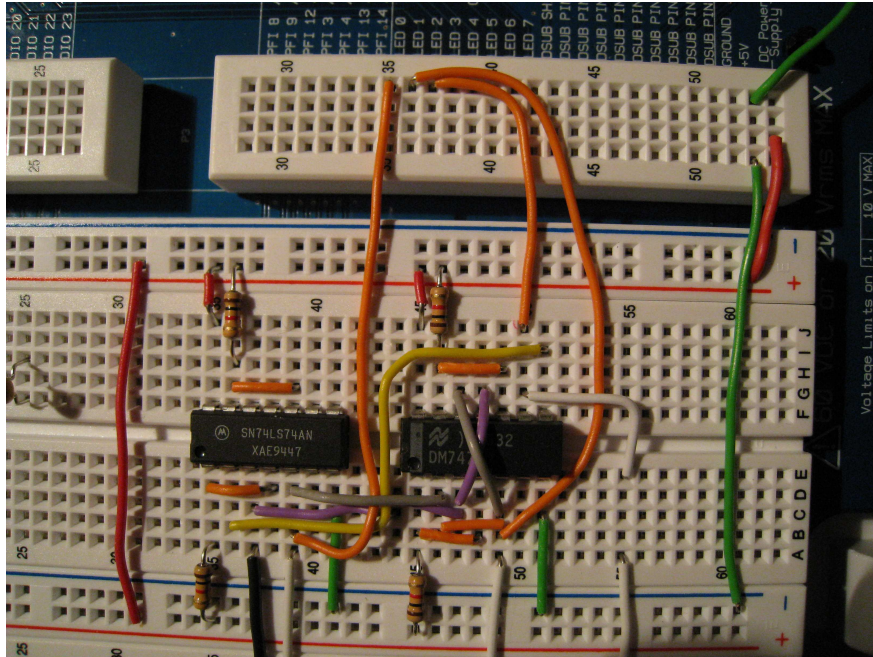


Figure 11.5. Real Component Modulo 6 Counter connected to NI ELVIS II LEDs

The Multisim program Mod 62.ms11 demonstrates this option.

3. Use the same NI-ELVISmx Function Generator as you did in Exercise 11.3.
4. Switch back and forth between Multisim and the NI ELVIS II protoboard to verify the operation of your modulo 6 counter.

End of Exercise 11.4

Exercise 11.5: Building the System Clock

To roll the dice, you can use a high clock speed. A clock frequency of several kilohertz is fast enough so that the eye cannot tell which numbers are appearing on the LED dice display. It looks just like the dice are rolling.

Complete the following steps to build the system clock.

1. Review the digital clock circuit from Lab 5, Exercise 5.2.
2. Design a Multisim circuit for a TTL clock using a 555 timer IC configured as a stable oscillator.

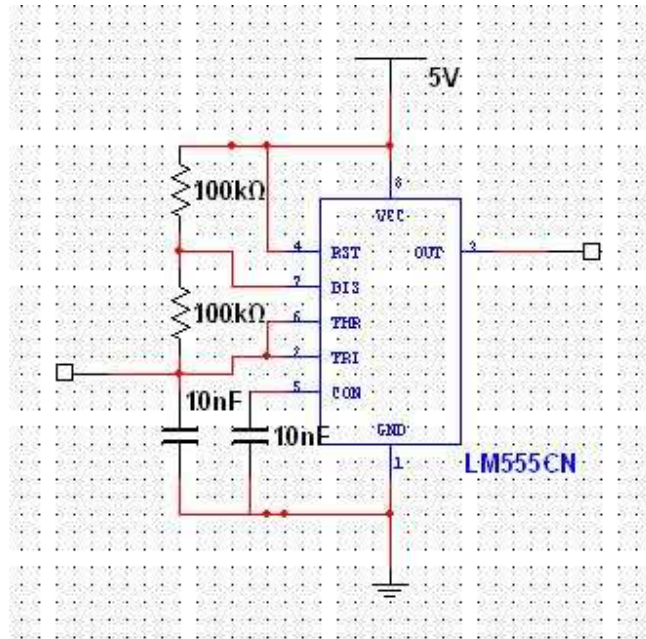


Figure 11.6. Multisim 555 Clock Circuit

3. Use the following components

$$R_A = 100 \text{ k}\Omega$$

$$R_B = 100 \text{ k}\Omega$$

$$C = 0.01 \text{ }\mu\text{f}$$

to complete the clock circuit and define the frequency of operation.

4. Use the NI ELVIS II SFP (**Scope**) to verify the circuit operation and measure the clock frequency.
5. To stop the dice rolling, you need only to stop the clock. You can simulate this feature, “One-armed bandit,” using a simple switch connected between Ground and pin 3 of the 555 timer IC.
6. Modify your clock design with this addition and verify the circuit operation. You should see the oscillator stop when the switch is closed.

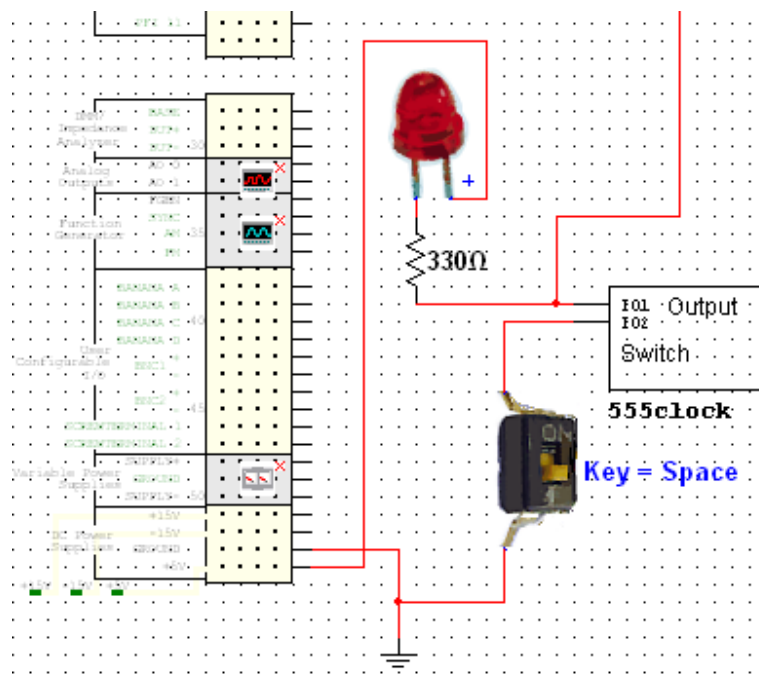


Figure 11.7. Added components to convert 555 Clock to Gated Oscillator

7. When the circuit is operating correctly, replace the component list above with a second set of components to slow the clock down to a speed where your eye can follow the action.

$$R_A = 100 \text{ k}\Omega$$

$$R_B = 100 \text{ k}\Omega$$

$$C = 2 \text{ }\mu\text{f}$$

8. Modify your circuit a second time by adding a single LED between the +5 V supply and by connecting a current limiting resistor to the 555 output (pin 3).
9. Power up the protoboard and click on **Run** to verify the slow speed clock operation.
10. Use the Multisim program Clock.ms11 in the NI ELVIS II library folder to check on your program.

End of Exercise 11.5

Exercise 11.6: Building a Real Clock Circuit on an NI ELVIS II Protoboard

Complete the following steps to build a real clock design with modifications on an NI ELVIS II protoboard.

1. Transfer your second clock design to the NI ELVIS II protoboard. You need a 555 timer IC, resistors R_A and R_B , capacitor C , a $330\ \Omega$ resistor, a visible LED, and a switch.

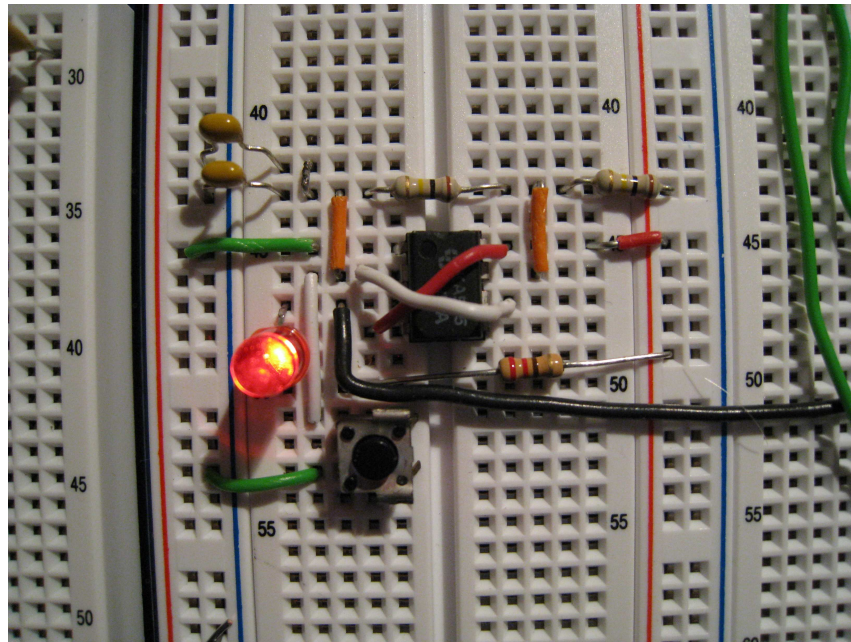


Figure 11.8. Gated System Clock Circuit on NI ELVIS Protoboard

2. When the circuit is completed, power up the protoboard. You should see the LED flashing.
3. Power off the protoboard.
4. Remove the NI ELVIS SFP connection from your modulo 6 counter and replace it with a connection between the 555 output (pin 3) and the clock input of the modulo 6 counter.
5. Power up the protoboard. You should see the clock output LED flashing and the three output lines Q1, Q2, and Q3 on the modulo 6 counter displaying the count on the protoboard LEDs 0, 1, and 2.
6. Power off the protoboard.
7. Replace the second (R_A , R_B , C) set with the first (R_A , R_B , C) set. When you apply power to the protoboard, you should not be able to see any counting – just a blur on the LEDs.

8. Push the switch and see the current count on the modulo 6 output lines.

End of Exercise 11.6

Exercise 11.7: Building the Three- to Four-Line Encoder

The encoder takes the three counter outputs, Q1, Q2, and Q3, and sets the base states, A, B, C, and D, to represent the dice numbers 1 through 6. There is no a priori reason to decide which counter output corresponds to which count. However, a little foresight makes the choices easier.

#	Q ₁	Q ₂	Q ₃	Q ₁ '	Q ₂ '	Q ₃ '
6	0	0	0	1	1	1
4	1	0	0	0	1	1
2	1	1	0	0	0	1
1	1	1	1	0	0	0
3	0	1	1	1	0	0
5	0	0	1	1	1	0

Table 11.3. All possible states of a Modulo 6 Switch Tail Ring Counter

For example, each output has three (1) states and three (0) states. You could use one of these outputs, say Q3, to signify odd states 1, 3, and 5. You could then use another output state, say Q2, to code the family 4, 5, and 6. These two lines then decode two of the base patterns for “free.”

To decode the two remaining base patterns with a particular pattern of the three counter lines, you can use a three-input AND gate. Decode “Not 1” with the combination Q1, Q2, and Q3 and decode the final base state “6” with Q1, Q2, and Q3.

Build the encoder using two three-input AND gates with four inverter gates. The schematic diagram in Figure 11.9 shows the encoder circuit.

NOTE: All pin-outs for IC's can be found online by searching the part number on the IC.

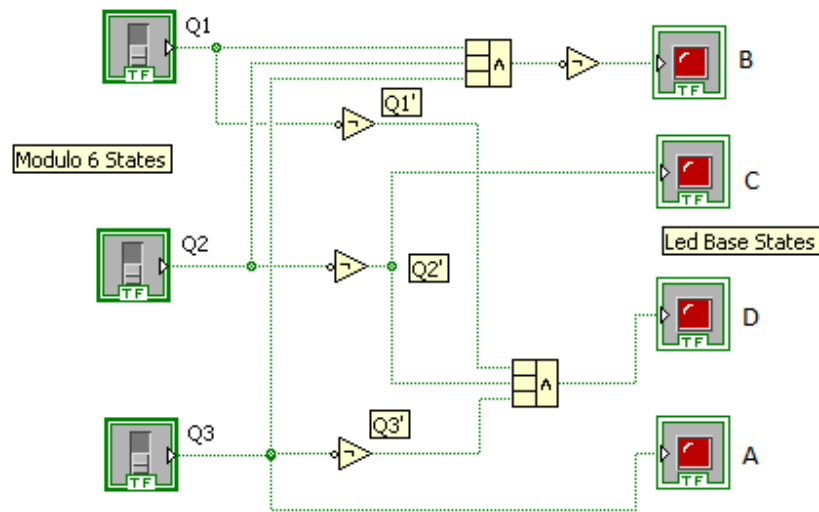


Figure 11.9. LabVIEW coded version of a 3-to-4 line Encoder

Recall that the dice display in Exercise 11.2 requires that the dice display inputs be active LO to turn on the LEDs. This implies that the outputs of the encoder in Figure 11.9 must be inverted. You can do this by replacing the AND gates with NAND gates and rearranging to connections to A, B, C, and D. Study the Figure 11.9 design and make the necessary corrections. Sketch the new encoder circuit on paper.

Complete the following steps to build the new encoder design.

1. Design a Multisim circuit for the new encoder. Use a 7410 (triple three-input NAND gate) IC and a 7404 (hex inverter) IC.
2. Set the encoder input lines Q1, Q2, and Q3 with three output lines, DIO 0, 1, and 2, of the NI-ELVISmx Digital Writer.
3. View the output states A, B, C, and D with four lines, DIO 8, 9, 10, and 11, of the NI-ELVISmx Digital Reader.
4. For assistance, take a peek at Encoder.ms11 in the Hands-On NI ELVIS II library folder.
5. Run the simulation and verify the truth table for the three- to four-line encoder.

End of Exercise 11.7

Exercise 11.8: Building and Testing the Digital Dice Encoder

Complete the following steps to build and test the encoder.

1. Transfer your Multisim encoder design to a real circuit on an NI ELVIS II protoboard. You need a 7410 (triple three-input NAND gate) IC and a 7404 (hex inverter) IC.

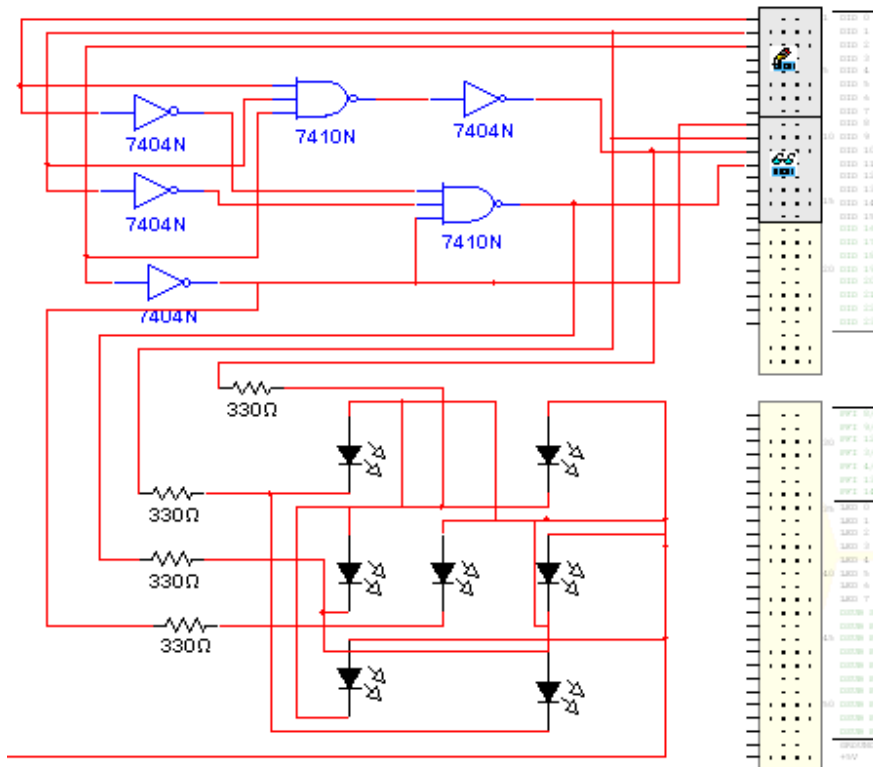


Figure 11.10. Multisim circuit for 3-to-4 line Encoder with Dice Display

2. Use the NI-ELVISmx Digital Writer to set the inputs Q1, Q2, and Q3.
3. Use the NI-ELVISmx Digital Reader to view the encoder output states A, B, C, and D.
4. When you are satisfied with the circuit operation, replace the lines to the NI-ELVISmx Digital Reader with the input lines for the real digital dice display.
5. Make one final check on the encoder/LED display circuit.
6. Power up your circuits. Verify the encoder/LED display generates the dice patterns (1 ..6).

7. If all works as expected, connect the four digital dice components: clock, counter, encoder, and display.

End of Exercise 11.8

Exercise 11.9: Electronic Dice

To roll a die, turn on the protoboard and set the switch to open. All the numbers appear on the LED display but at a speed too fast for the eye to follow the patterns.

By flipping the switch to close, you can stop the counter. This state is converted into one of the six possible display patterns, maybe your favorite number.

End of Exercise 11.9