AN-1172 LED Cube with Animation to the Music Rhythm

This Application Note describes how to design a simple 3x3x3 LED cube and control it with a GreenPAK™ SLG46538V. We will discuss how the GreenPAK uses its features to create great animations on the cube to the rhythm of the music.

Large 3D LED cubes are often used for visual entertainment, sometimes paired with a microphone to interact with music or voices, and can be found in concerts and clubs. In this app note we will discuss three animation schemes powered by GreenPAK, and explain how to create animations for LED cubes of different sizes.

Building the cube

There are plenty of tutorials online that explain how to build an LED cube, but we recommend this one, which is very clear and easy to follow. The cube should have three layers and nine independent columns.

Once you have built your cube, we can proceed to the next step. Some LED cube tutorials use solderable breadboards (aka perfboards) as the base for the cube circuit, but this is not necessary. The good news is that it is easy to manage the power source for the cube since there are only 27 LEDs to control. Larger cubes would need more careful design for controlling the power source.

Schematic

The overall schematic of the circuit is shown in Figure 1. It uses 3 NPN transistors to sink current to GND for each of the three layers. For this project we have selected the SLG46531V as the number of pins fits our requirement nicely.

It is important to use the three transistors to connect each layer to the GreenPAK, because the GreenPAK is not able to support the maximum current if every LED in the layer is on. Each of the columns is wired with 470Ω resistors to limit the current enough for the LEDs.

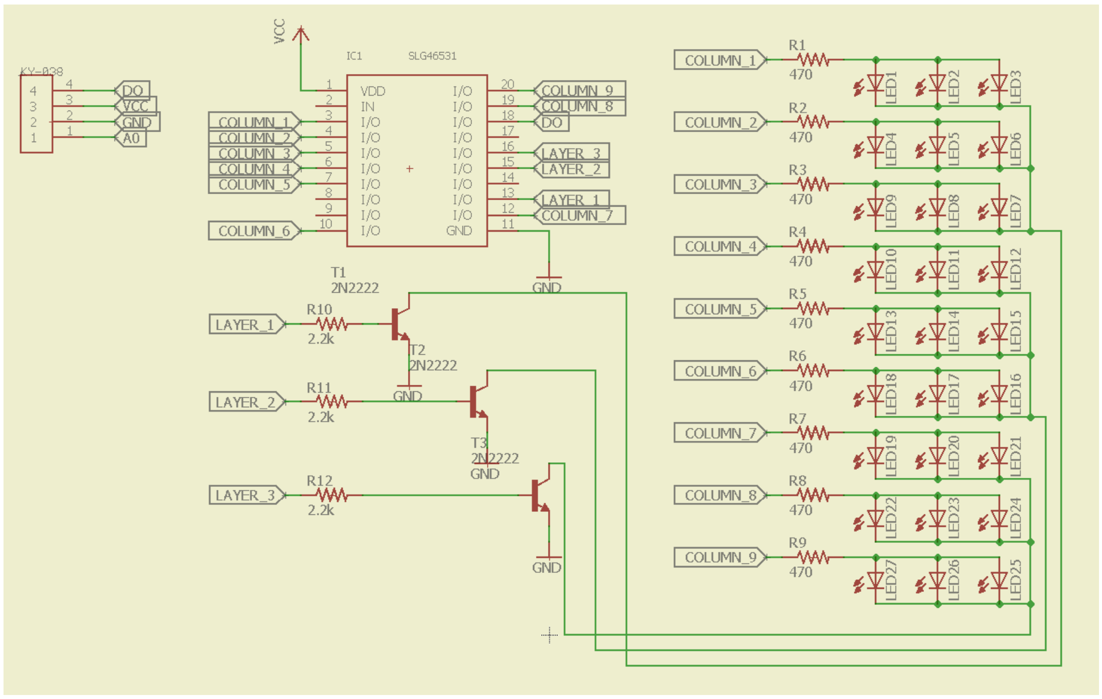

Figure 1. 3D Cube 3x3x3 Schematic

GreenPAK Design

When designing an animation for the cube, it is important to know the limitations of the GreenPAK: each GreenPAK IC can control only one animation. Despite this, when a complex cube (such as an

8x8x8 cube) is designed with numerous integrated GreenPAKs, great animations can be designed at very low cost and with good performance.

The most important components for designing an LED cube using GreenPAK are: the Asynchronous State Machine (ASM), D Flip Flops (DFFs), Look-Up Tables (LUTs) and logic gates, an appropriate number of pins, and the clock source. The ASM's task is to change from one state to another, which helps show different kinds of LED combinations on the cube.

Due to a limited number of ASM output bits, one should use DFFs and LUTs to cover all the pins associated with the columns and layers of the cube. DFFs can be configured as shift registers, which is crucial for animation designs because it enables each of the ASM states to be configured to directly control outputs to the cube. The clock source of the DFFs enables sound syncing, as the sound detection sensor will behave as the clock. The sound will excite the DFFs to shift the current bit data to a new one, resulting in LEDs animation.

There were three animations designed for this Application Note. Animation #1 was designed without sound detection, and Animations #2 and #3 were designed to sync with the music detected by the sensor.

Animation #1

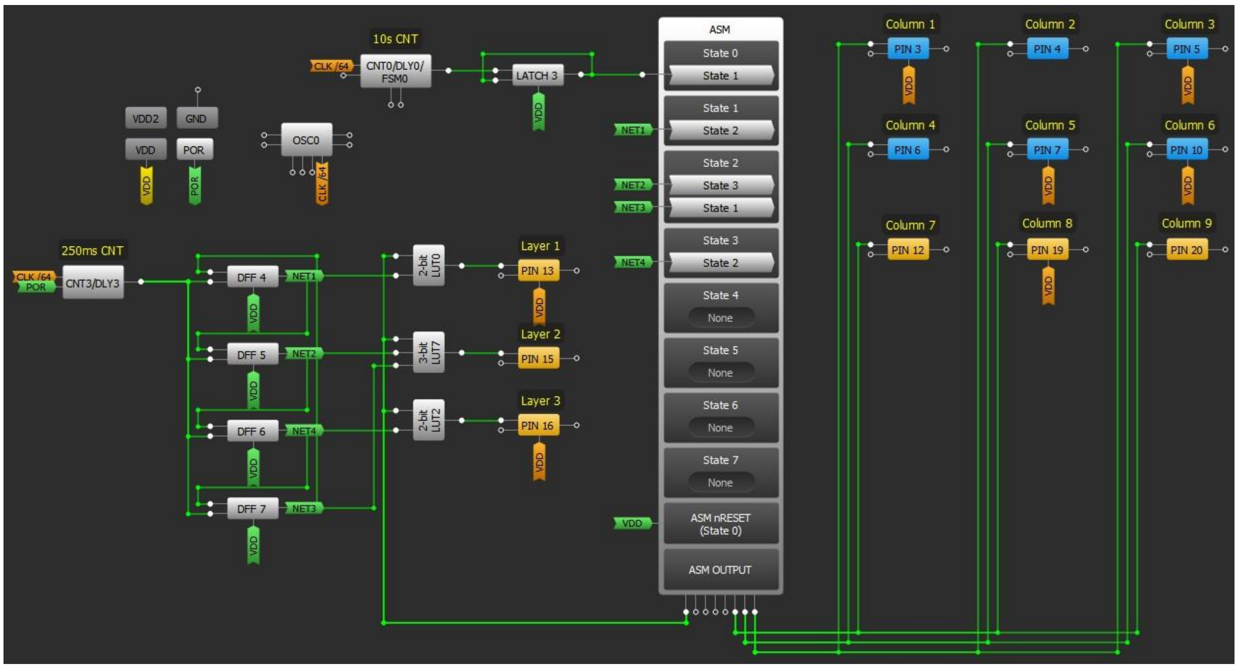

Figure 2. Animation #1 GreenPAK Design

The purpose of this design is to simulate a simple cube animation that turns ON all LEDs layer by layer, and after a certain amount of time (when CNT0 reaches its limit) the cube will turn ON all LEDs. This allow the user to easily test if all LEDs on the cube are connected and functional.

Animation #2

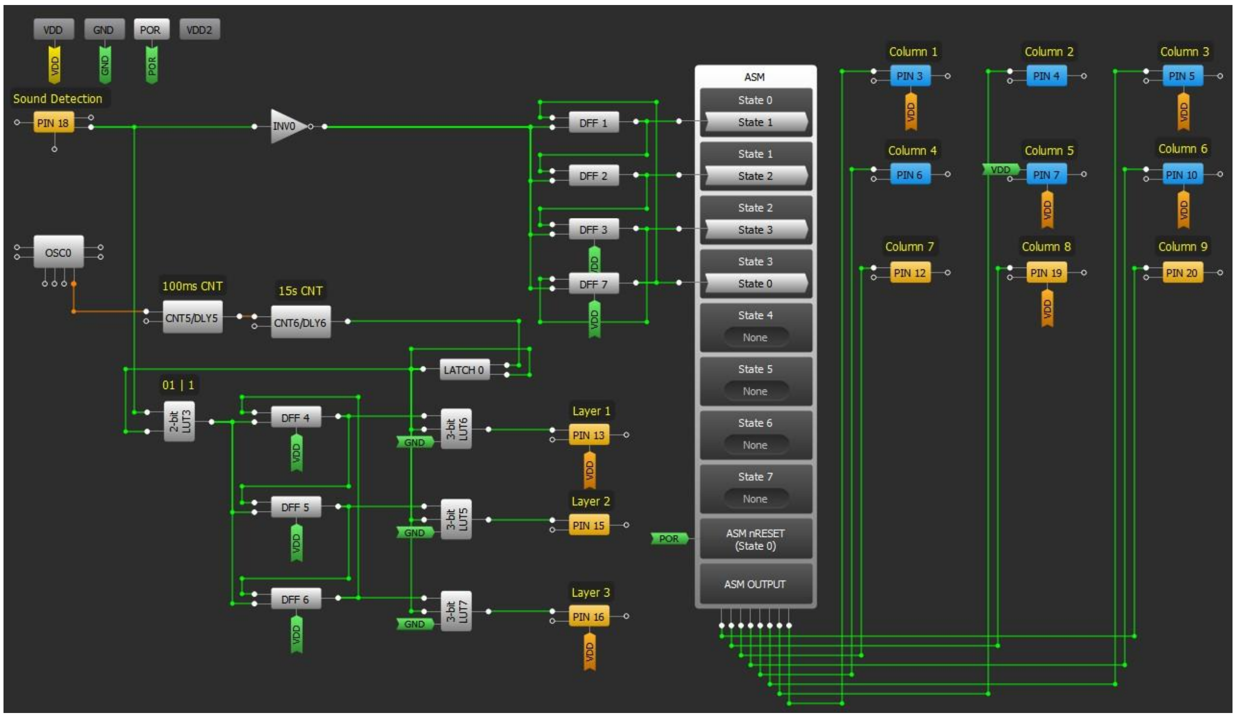

Figure 3. Animation #2 GreenPAK Design

For the sound detection circuit, we used an off-the-shelf KY-038 sound detection module to sync the sounds of a song to the cube, which allows it to behave in many ways for visual diversity. This was done by first connecting the module's VDD and GND, and then connecting its Digital Output (DO) to Pin 18 of the GreenPAK. Each time the module detects a sound, it toggles the signal sent to Pin 18. The Analog Output (AO) of the KY-038 is not used in this application. A more extensive calibration of the built-in potentiometer would be necessary for the module to detect sound as desired.

The second animation designed for the cube rotates in sync with music. Figure 5 is a representation of a single layer of the cube. The columns are controlled directly from the ASM tool, except COLUMN_5. The ASM, with four states (from 0 to 3), will follow an order sequentially (as shown in Table 1 and Figure 4).

States	ASM Outputs	Columns
State 0	OUT3, OUT4	Column_4, Column_6
State 1	OUT2, OUT5	Column_3, Column_7
State 2	OUT1, OUT6	Column_2, Column_8
State 3	OUT0, OUT7	Column_1, Column_9

Table 1. ASM State Outputs for Animation #2

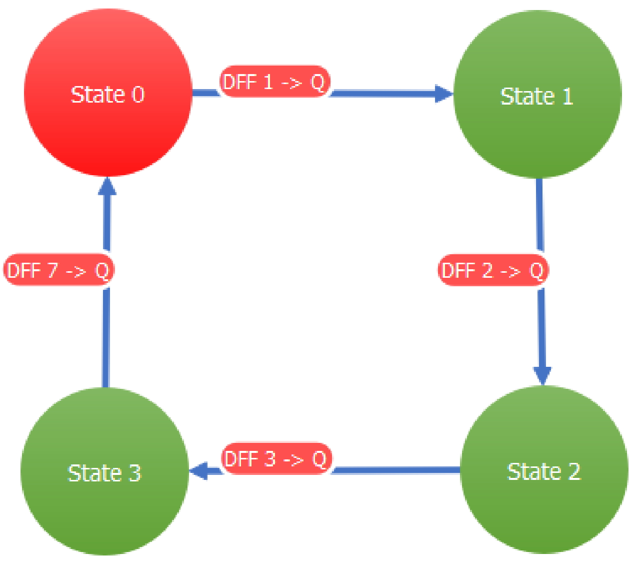

Figure 4. ASM State Transition for Animation #2

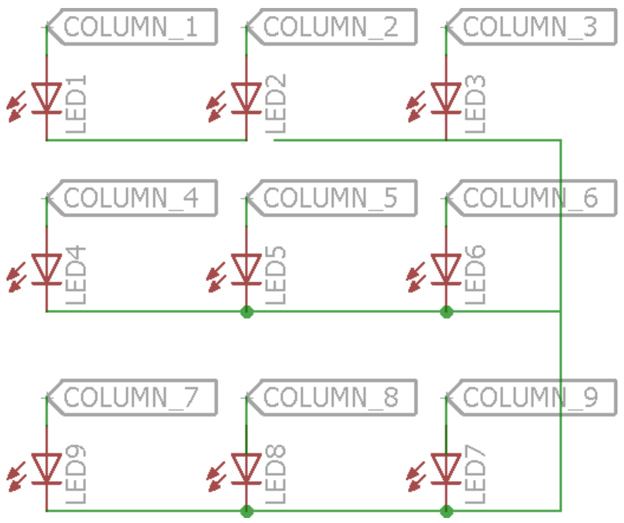

Figure 5. 3D Cube Single Layer Representation

As you can see, the columns will alternate as a kind of roulette, rotating around the cube. Layers are alternated by the DFF shift register, as well as the enable pins of the ASM which allow the alternation of the columns.

On DFFs you can set a static clocked time with a counter, but in this animation the sound detection sensor is used as the clock source. This shifts the data with the rhythm of the music. This could be called a "variable clock source," because every piece of music has unique timing and sounds, resulting in a random animation with many combinations of transitions.

Notice how Pin 18, the sound detection sensor, is connected to each of the CK input pins of the DFFs. Part of the design (beginning of the animation) is to wait 15 seconds with CNT5 and then to set LATCH0 to save its new value HIGH. During this initial period, all three layers will be LOW (look at the LUTs 5, 6 and 7), meaning that all layers will work at the same time while the ASM is "rotating roulette."

Note: It's important to calibrate the sensor module's potentiometer. The distance from the speaker to the sensor's microphone is very important to calibrate the sensitivity.

Animation #3

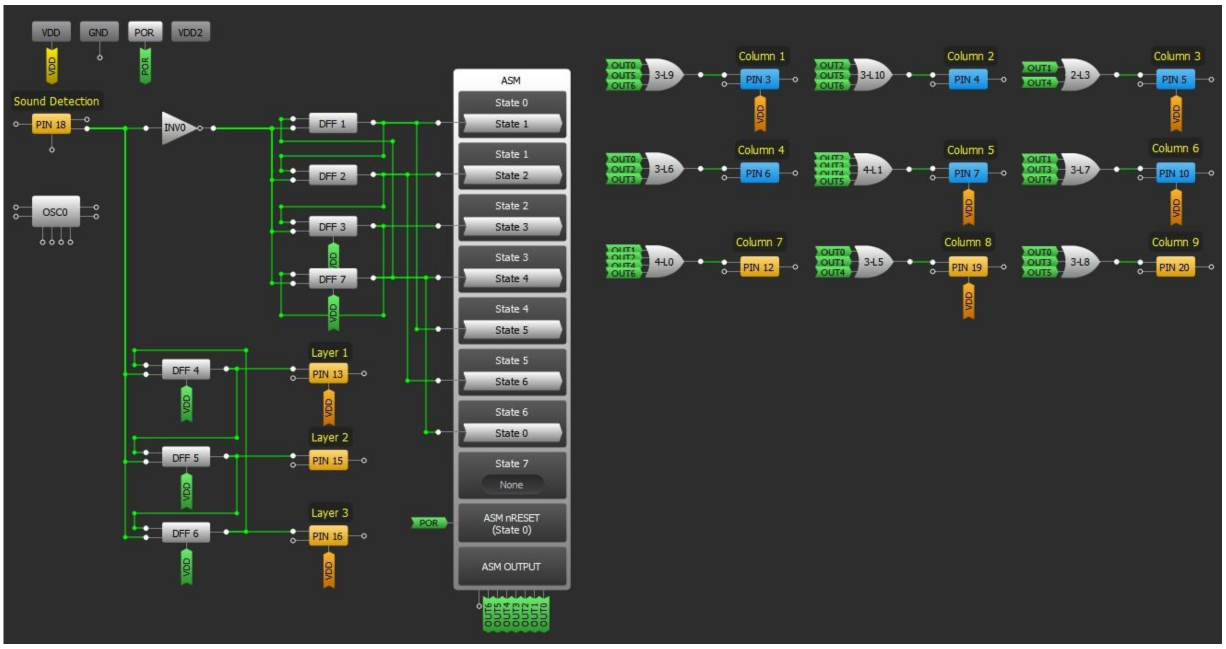

Figure 6. Animation #3 GreenPAK Design

This animation is more random than the previous one. It was designed with more LUTs to control columns shared with different ASM outputs. Take another look at Figure 4 to follow this animation with respect to the ASM states, which are different from Animation #2. An ASM with seven states (from 0 to 6) will follow the order seen in Table 2 and Figure 7.

States	ASM Outputs	Columns
State 0	OUT0	Column_1, Column_4, Column_8, Column_9
State 1	OUT1	Column_3, Column_6, Column_7, Column_8
State 2	OUT2	Column_2, Column_4, Column_5, Column_7
State 3	OUT3	Column_4, Column_5, Column_6, Column_9
State 4	OUT4	Column_3, Column_5, Column_6, Column_7, Column_8
State 5	OUT5	Column_1, Column_2, Column_5, Column_9
State 6	OUT6	Column_1, Column_2, Column_7

Table 2. ASM State Outputs for Animation #3

In this animation, the columns will alternate with some different random patterns, making it more dynamic than the previous animation. This animation uses more LUTs as OR gates to create more combination possibilities when the animation is running with the music. OR gates accept as many inputs as possible from the ASM outputs. In this way, a pin associated with a column can be controlled from different input sources, to create different figures for each state executed. As for the previous animation, the ASM diagram (Figure 7) is very simple. It runs as a cycle or a loop, so once it reaches the final state, it will return to the initial one.

From the beginning, the 3D Cube will start to change its state once it detects a signal from Pin 18 (the sound detection sensor), to dance with the music. As in the previous animation, states in the ASM will be excited from DFFs shifting the bits; the same occurs for the layers' associated pins.

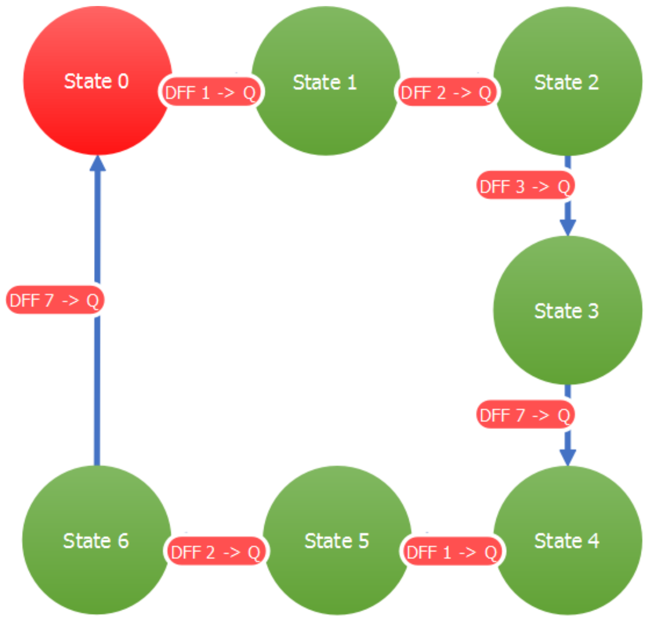

Figure 7. ASM State Transitions for Animation #3

Additional Information

We've discussed how to create some LED cube animations just using GreenPAK chips, but there is another way to design these animations. Microcontrollers or microprocessors can be used for more complex animations. One could implement the GreenPAK 5 as an I/O expander using the I2C and ASM tools, with the goal for the GreenPAK design inside the general system architecture being to decrease the number of pins used from the processors to a bigger cube such as 16x16x16 LEDs.

One can easily use GreenPAK as the main controller for the cube design with cubes from 3x3x3 to 8x8x8. However, keep in mind that the more LEDs there are in your LED cube, the more challenging it will be to design a control scheme. You may need to use multiple GreenPAK chips to control a large cube.

Conclusion

The GreenPAK can be a great ally when you wish to design a 3D cube for visual entertainment. With multiple GreenPAK chips you can create just about any animation that you can imagine. Using a sound detector allows your design to interact with music, causing a magical effect. Thanks to GreenPAK's small size, simplicity, and high configurability, you can create a high quality and very compact circuit for driving a 3D cube, with low cost and a quick development process.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks