ISSUE: April 2019

Configurable IC Simplifies Control Of Animated LED Turn Signals

by Muhammad Saqib and Aamir Hussain Chughtai, Dialog Semiconductor, Lahore, Pakistan

Many automotive manufacturers now provide indicator lights (turn signals) with animated LED patterns to enhance the aesthetics of these lights and to create a "trademark" look that distinguishes their vehicles. In fact, these animated LED patterns have become a norm in the automotive industry. The animations can be of different running patterns and can be implemented without an MCU but typically require several discrete ICs. These control components are used in combination with an LDO regulator and one or more automotive LED drivers.

The major requirements of such designs are: reproducible performance during normal operation, an option to force all LEDs on, low power consumption, disabling the LDO regulator during a fault, loading the LED driver before enabling it, etc. Additionally, the requirements can vary from one manufacturer to another. Moreover, usually in automotive applications, TSSOP ICs are preferred due to their robustness as compared to QFN ICs since the latter are prone to solder fatigue issues especially in harsh environments.

While all of the requirements for the animated LED patterns are currently met in the automotive industry using discrete ICs to generate the LED control signals, an alternative approach based on the SLG46620 configurable mixed signal IC (CMIC) is presented here. The configurability of the CMIC provides a high level of flexibility, which allows a supplier to easily cater to the varying requirements of several manufacturers without any change in hardware design. Moreover, this approach achieves significant reductions in the associated PCB footprint and the bill-of-materials cost.

This article presents a detailed description of how to achieve different animated LED patterns in turn signals using the SLG46620. After a brief explanation of the component cost savings made possible by the IC, two possible system design schemes are given using either a single-channel or multiple-channel LED driver. Then the article explains how the CMIC can be configured as a 16-state Moore machine to generate the three example LED patterns. In subsequent sections, the equations for the enable signal and the LED driving signals are given along with the internal configurations of the CMICs for each of the three example LED patterns.

The performance of each of these design examples is then verified experimentally. Links to the actual design files and the related development tools are provided in the reference section.

Industry Value

To generate the LED turn-signal patterns shown in this article, automotive companies currently employ designs consisting of a number of discrete ICs. But, by using an SLG46620 CMIC to control the sequence of automotive indicator LED patterns, the bill of materials can be dramatically reduced.

The SLG46620 would replace at least the following components in a current industrial design:

- 1 555 Timer IC (e.g. TLC555QDRQ1)
- 1 Johnson Counter (e.g. CD4017)
- 2 D-type positive-edge-triggered flip-flops (e.g. 74HC74)
- 1 OR gate (e.g. CAHCT1G32)
- Several passive components (i.e. inductors, capacitors, resistors, etc.)

Table 1 quantifies the cost associated with the existing design currently used in industry to generate the indicator light sequential-turn-signal patterns. We can then compare this with the cost of a single SLG46620 CMIC that performs the same function.

Table 1. Cost of LED turn-signal control circuitry using an existing industry solution.*

IC Name	Quantity	Unit Price (Ref. Digi-key 23-05-2018) ($)	Price ($)
TLC555QDRQ1	1	0.86	0.86
SN74HC74QDRG4Q1	2	0.58	1.16
CD74HC4017E	1	0.58	0.58
CAHCT1G32QDBVRQ1	1	0.42	0.42
		Total Price:	3.02

*Note: This table omits the cost of other auxiliary components (i.e. several resistors, capacitors and inductors) that are required.

In contrast with the $3 cost for the ICs in the industrial design, the selected CMIC SLG 46620 would cost less than $0.50. So the total cost of the LED control circuitry decreases significantly with the CMIC-based design. In addition, a significant comparative PCB footprint reduction is also achieved.

System Design

Fig. 1 shows the diagram of the first proposed scheme. The major components of the scheme include an LDO voltage regulator, an automotive LED driver, an SLG46620 CMIC, 11 logic-level MOSFETs and 10 LEDs. The LDO voltage regulator ensures that appropriate voltage is provided to the CMIC and if the battery voltage drops from a certain level the CMIC gets reset through the PG (power good) pin. During any fault condition, detected by the LED driver, the LDO voltage regulator gets disabled.

The SLG46620 CMIC generates the digital signals to drive the turn-signal LEDs labelled 1-10 through the MOSFETs. Moreover, the selected CMIC also produces the enable signal for the single-channel driver which in turn drives MOSFET Q1 to load the driver running in constant-current mode.

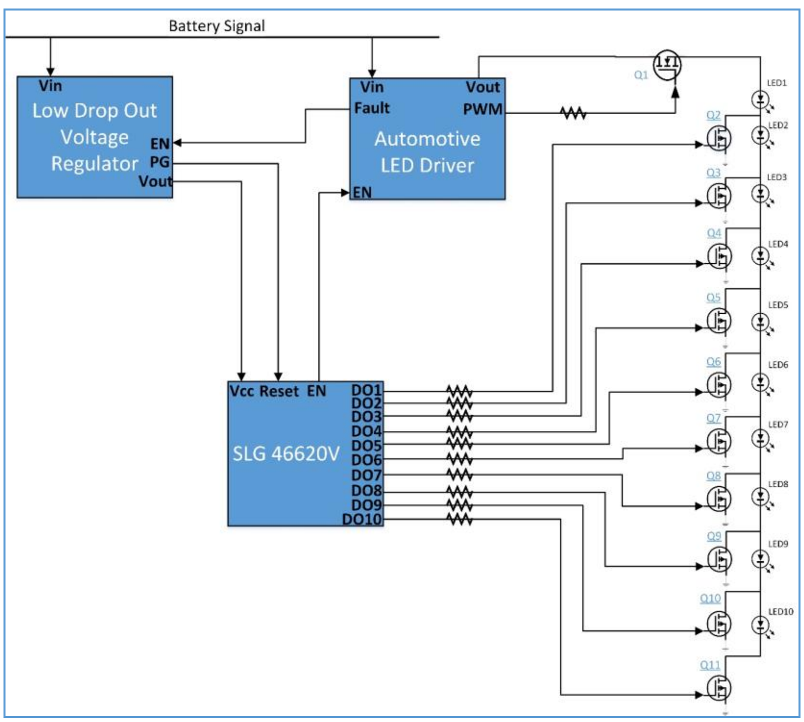

Fig. 1. Scheme 1 for LED turn-signal control circuitry with single-channel driver.

A variant of this scheme is also possible in which two multi-channel drivers are employed, as shown in Fig. 2. In this option, the driving current of each channel reduces as compared to the single-channel driver.

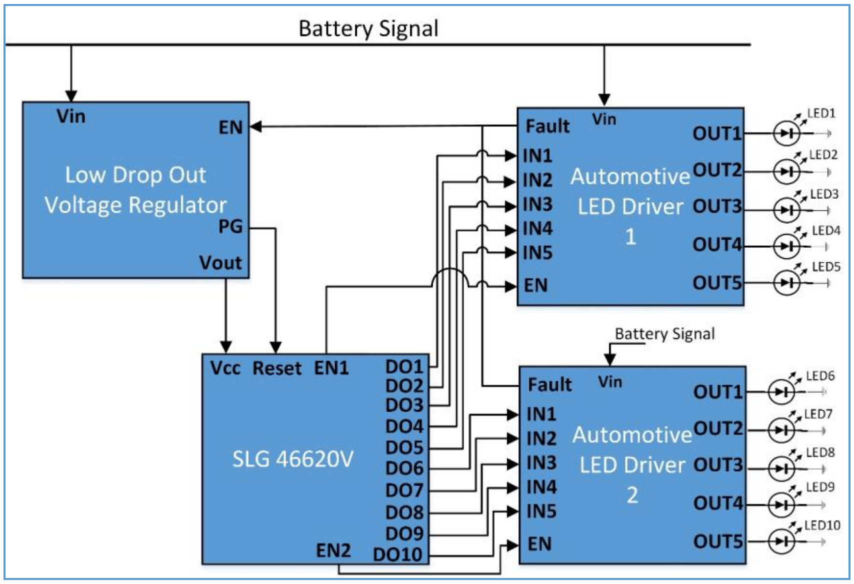

Fig. 2. Scheme 2 for LED turn-signal control circuitry with two multi-channel drivers.

CMIC Design

A suitable way to achieve the goal of flexible indicator LED patterns is to use a finite state machine (FSM) concept. Dialog Semiconductor provides several CMICs that contain a built-in ASM (asynchronous state machine) block. Unfortunately all those CMICs are only available in QFN packages, which are not recommended for harsh environments. So the SLG46620, which does not have the ASM, is chosen because it is available in both the QFN and the TSSOP package which is preferred in automotive applications. (We'll explain shortly how to get around the lack of the ASM.)

Three examples are presented for three different LED animations. For the first two examples, we consider a single-channel driver as shown in Fig. 1. For the third example, we assume that multi-channel drivers are available, as shown in Fig. 2 and each channel is used to drive a separate LED. Other patterns can also be obtained using the same concept.

In the first example design, LEDs 1-10 are sequentially turned on one after the other once a certain programmable time period expires as shown in Fig. 3.

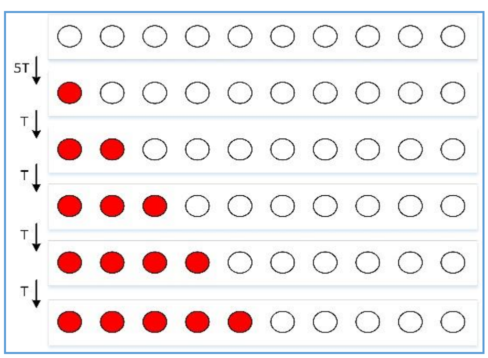

Fig. 3. Example 1 pattern.

In the second example design, two LEDs are sequentially added in the pattern as shown in Fig. 4.

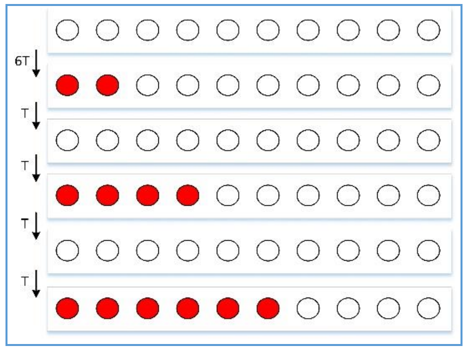

Fig. 4. Example 2 pattern.

Fig. 5 depicts how alternate LEDs are sequentially added in the pattern in the third proposed design.

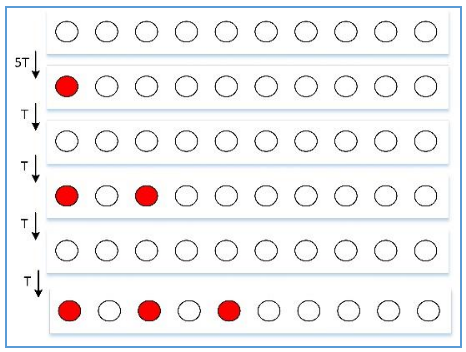

Fig. 5. Example 3 pattern.

Since there is no built-in block of ASM available in the SLG46620, a finite state Moore machine is developed using the available blocks, namely a counter, DFFs and (look-up tables) LUTs. A 16-state Moore machine is developed using Table 2 for the three examples.

In Table 2, all the bits of the present state and the next state are given. Moreover, the bits for all the output signals are also provided. From Table 2 the equations of the next state and all the outputs are evaluated in terms of the present state bits.

Table 2.16-state Moore machine for the three example pattern designs.

Indicator Signal	Present State ABCD	Next State ABCD	En (Ex. 1)	LEDs On (Ex. 1)	En (Ex. 2)	LEDs On (Ex. 2)	En1 (Ex. 3)	En2 (Ex. 3)	LEDs On (Ex. 3)
0	0000	1111	0	0	0	0	0	0	0
0	0001	1111	0	0	0	0	0	0	0
0	0010	1111	0	0	0	0	0	0	0
0	0011	1111	0	0	0	0	0	0	0
0	0100	1111	0	0	0	0	0	0	0
0	0101	1111	1	1	0	0	1	0	1

Indicator Signal	Present State ABCD	Next State ABCD	En (Ex. 1)	LEDs On (Ex. 1)	En (Ex. 2)	LEDs On (Ex. 2)	En1 (Ex. 3)	En2 (Ex. 3)	LEDs On (Ex. 3)
0	0110	1111	1	1+2	1	1+2	0	0	0
0	0111	1111	1	1+2+3	0	0	1	0	1+3
0	1000	1111	1	1+2+3+4	1	1+2+3+4	0	0	0
0	1001	1111	1	1+2+3+4+5	0	0	1	0	1+3+5
0	1010	1111	1	1+2+3+4+5+6	1	1+2+3+4+5+6	0	0	0
0	1011	1111	1	1+2+3+4+5+6+7	0	0	0	1	1+3+5+7
0	1100	1111	1	1+2+3+4+5+6+7+8	1	1+2+3+4+5+6+7+8	0	0	0
0	1101	1111	1	1+2+3+4+5+6+7+8+9	0	0	0	1	1+3+5+7+9
0	1110	1111	1	1+2+3+4+5+6+7+8+9+10	1	1+2+3+4+5+6+7+8+9+10	0	0	0
1	0000	0001	0	0	0	0	0	0	0
1	0001	0010	0	0	0	0	0	0	0
1	0010	0011	0	0	0	0	0	0	0
1	0011	0100	0	0	0	0	0	0	0
1	0100	0101	0	0	0	0	0	0	0
1	0101	0110	1	1	0	0	1	0	1
1	0110	0111	1	1+2	1	1+2	0	0	0
1	0111	1000	1	1+2+3	0	0	1	0	1+3
1	1000	1001	1	1+2+3+4	1	1+2+3+4	0	0	0
1	1001	1010	1	1+2+3+4+5	0	0	1	0	1+3+5
1	1010	1011	1	1+2+3+4+5+6	1	1+2+3+4+5+6	0	0	0
1	1011	1100	1	1+2+3+4+5+6+7	0	0	0	1	1+3+5+7
1	1100	1101	1	1+2+3+4+5+6+7+8	1	1+2+3+4+5+6+7+8	0	0	0

Indicator Signal	Present State ABCD	Next State ABCD	En (Ex. 1)	LEDs On (Ex. 1)	En (Ex. 2)	LEDs On (Ex. 2)	En1 (Ex. 3)	En2 (Ex. 3)	LEDs On (Ex. 3)
1	1101	1110	1	1+2+3+4+5+6+7+8+9	0	0	0	1	1+3+5+7+9
1	1110	0000	1	1+2+3+4+5+6+7+8+9+10	1	1+2+3+4+5+6+7+8+9+10	0	0	0
1	1111	0000	1	1+2+3+4+5+6+7+8+9+10	1	1+2+3+4+5+6+7+8+9+10	1	1	1+2+3+4+5+6+7+8+9+10

Note: The first three columns are common for each example.

At the core of the development of the 4-bit Moore machine are four DFF blocks. Each DFF block functionally represents one bit of the four bits: ABCD. When the indicator signal is high (corresponding to an on indicator switch), a transition from one state to the next is required at each clock pulse, thus generating different LED patterns as a result. On the other hand, when the indicator signal is low, a stationary pattern, having all the LEDs on in each design example is the goal.

Fig. 6 shows the functionality of the developed 4-bit (ABCD) Moore machine for each example. The basic idea of the development of such an FSM is to represent each bit of the next state, the enable signal and each output pin signal (assigned for the LEDs) in terms of the present state. This is where the LUTs contribute. All four bits of the present state are fed to different LUTs to basically achieve the required signal in the next state at the edge of a clock pulse. For the clock pulse, a counter is configured to provide a pulse train with a suitable period.

For each example, each bit of the next state is evaluated in terms of the present state using the following equations derived from K-Maps:

A = D' (C' + C (A B)') & IND + IND'

B = C' D + C D' (A B)' & IND + IND'

C = B' C D + B (C' + A' D') & IND + IND'

D = A B' + A' B C D + A B C' & IND + IND'

where IND represents the indicator signal.

Further details of each of the three examples are given below.

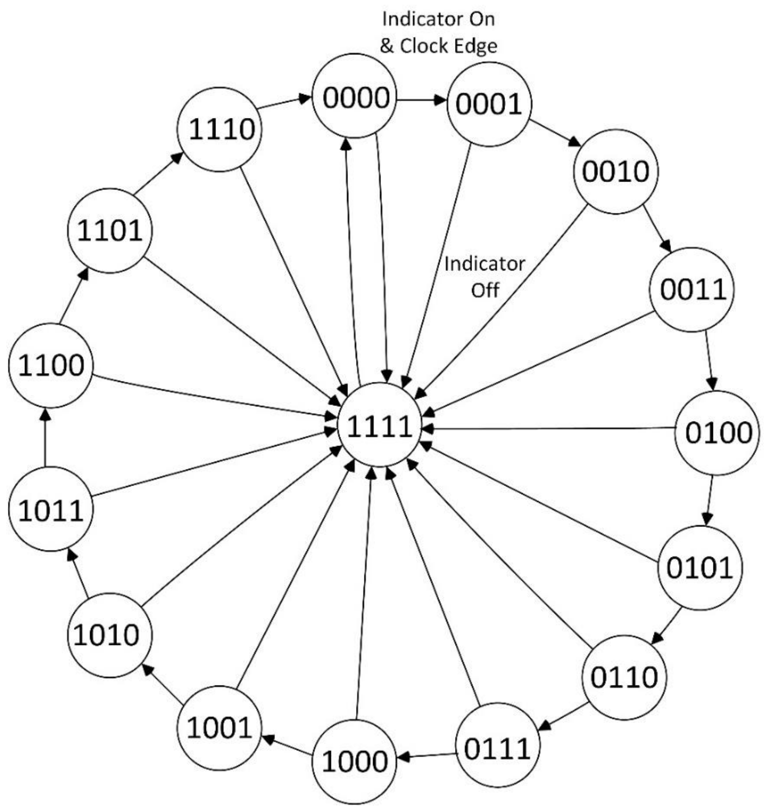

Fig. 6. 4-bit Moore state machine.

Design Example 1

The equations of the enable signal and the LED driving signals for the first example, with each LED turning on sequentially using the scheme in Fig. 1, are as shown below.

En = A + A' B (C+D)

DO1 = A' B C' D	DO2 = A' B C D'	DO3 = A' B C D	DO4 = A B' C' D'
DO5 = A B' C' D	DO6 = A B' C D'	DO7 = A B' C D	DO8 = A B C' D'
DO9 = A B C' D	DO10 = A B C		

In Fig. 7, the Matrix-0 GreenPAK design of example 1 is shown. Four DFFs are used to develop the 4-bit Moore machine. DFFs with reset option (three from Matrix-0 and one from Matrix-1) are selected so that the Moore machine can be reset conveniently. A counter, with a suitable time period of 72 ms, is configured to change the state of the machine after each period. LUTs with appropriate configurations are used to derive functions for the DFFs inputs, driver enable signal (En), and the output pins: DO1-DO10.

In Matrix-1 shown in Fig. 8, the rest of the GreenPAK resources are utilized to complete the design using the methodology described earlier. The figures are appropriately labeled for clarity.

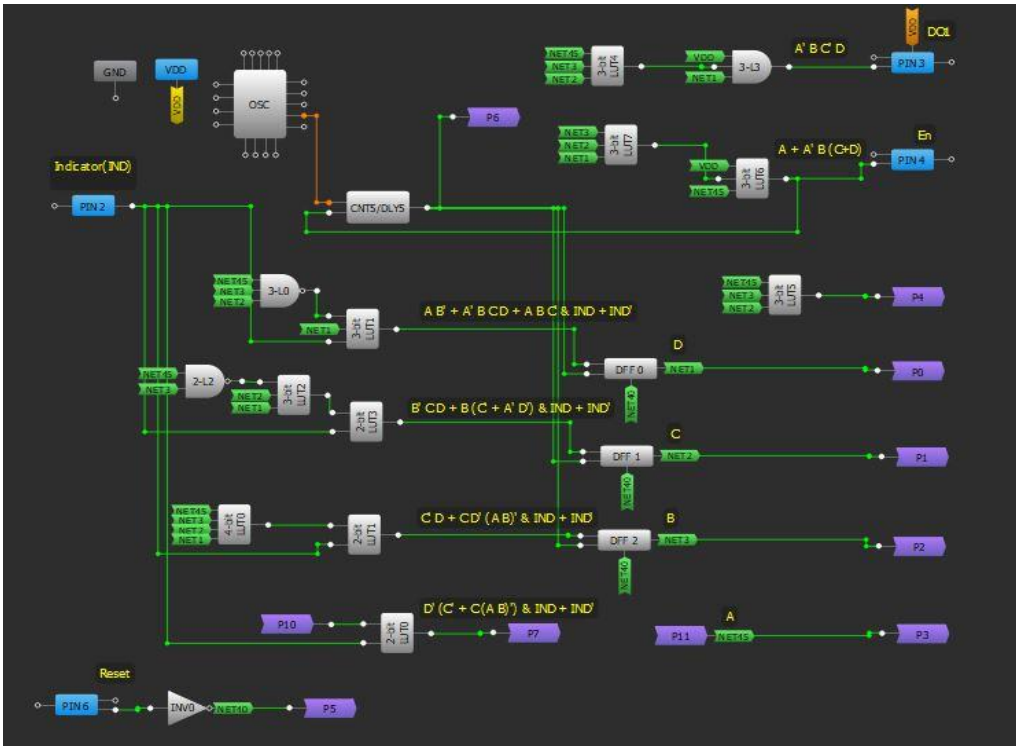

Fig. 7. Example 1 GreenPAK Design Matrix-0.

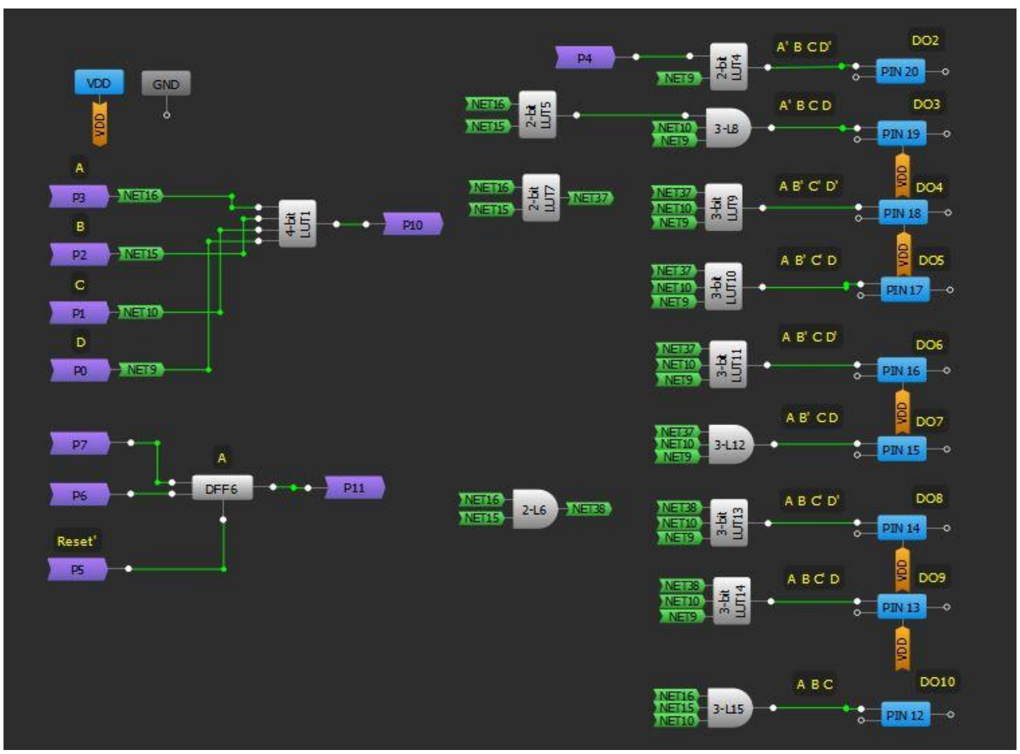

Fig. 8. Example 1 GreenPAK Design Matrix-1.

Design Example 2

The equations of the enable signal and the LED driving signals for the second example, with two LEDs adding in the sequential pattern using the scheme in Fig. 1, are as shown below.

En = D' (A' B C + A B' C' + A B' C + A B) + A B C

DO1 = 0	DO2 = A' B C D'	DO3 = 0	DO4 = A B' C' D'
DO5 = 0	DO6 = A B' C D'	DO7 = 0	DO8 = A B C' D'
DO9 = 0	DO10 = A B C		

In Figs. 9 and 10, the Matrix-0 and 1 GreenPAK designs of example 2 are presented. The basic design is similar to the example 1 design. The major differences, in comparison, are in the driver enable (En) function and the elimination of connections of DO1, DO3, DO5, DO7 and DO10, which are pulled down in this design.

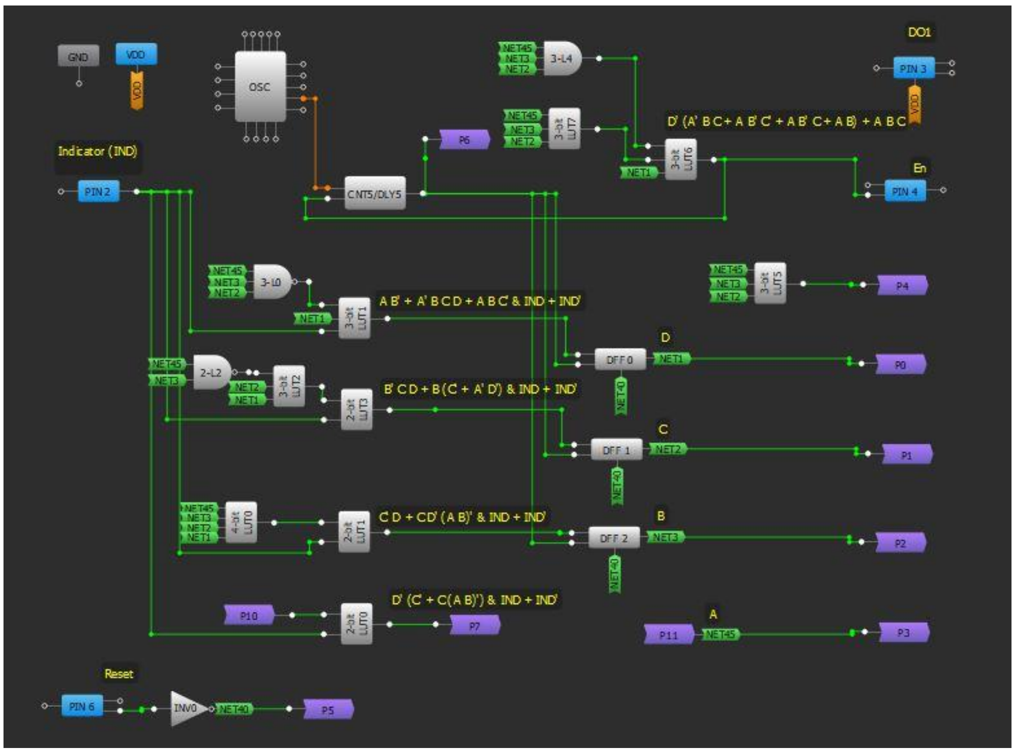

Fig. 9. Example 2 GreenPAK Design Matrix-0.

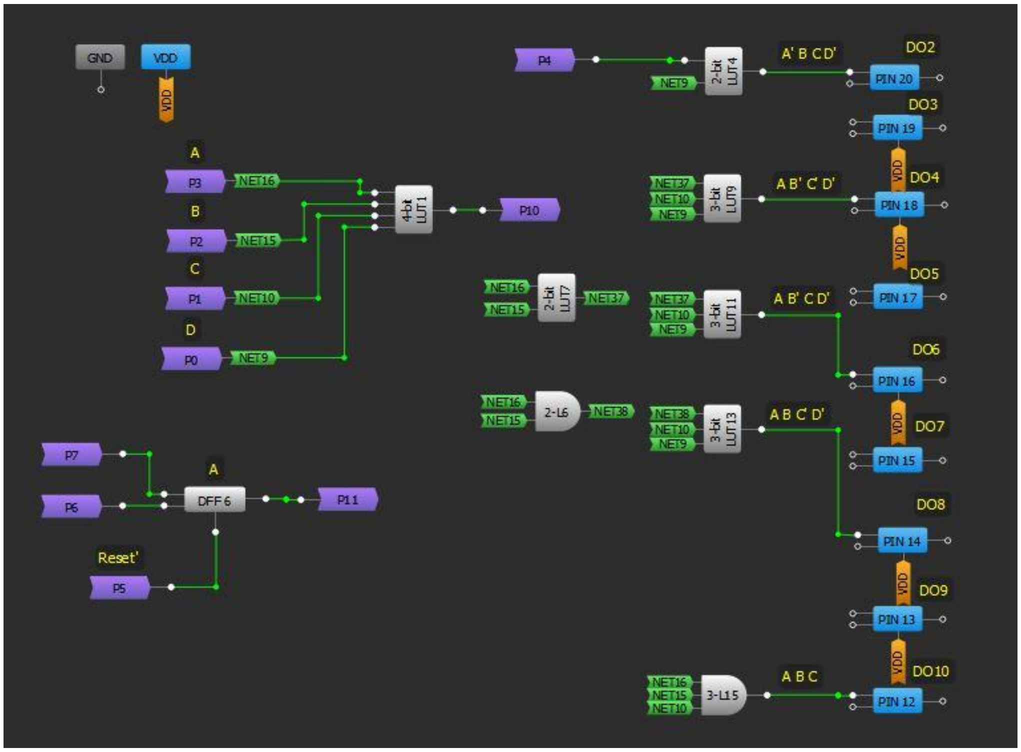

Fig. 10. Example 2 GreenPAK Design Matrix-1.

Design Example 3

The equations of the enable signal and the LED driving signals for the third example, generating the alternate LED sequential addition pattern using the scheme in Fig. 2, are given below.

En1 = (A' B C' + A B' C' + B C) D

En2 = (A B' C + A B) D

DO1 = D (A+B)	DO2 = A B C D	DO3 = D (A+ C B)	DO4 = A B C D
DO5 = D A	DO6 = A B C D	DO7 = D A (C' B + C)	DO8 = A B C D
DO9 = D A B	DO10 = A B C D		

In Figs. 11 and 12, the Matrix-0 & 1 GreenPAK designs of example 3 are presented. In this design, there two separate driver enable signals (En1 & En2) for driver 1 and 2. Moreover, the output pins are connected to the outputs of appropriately configured LUTs.

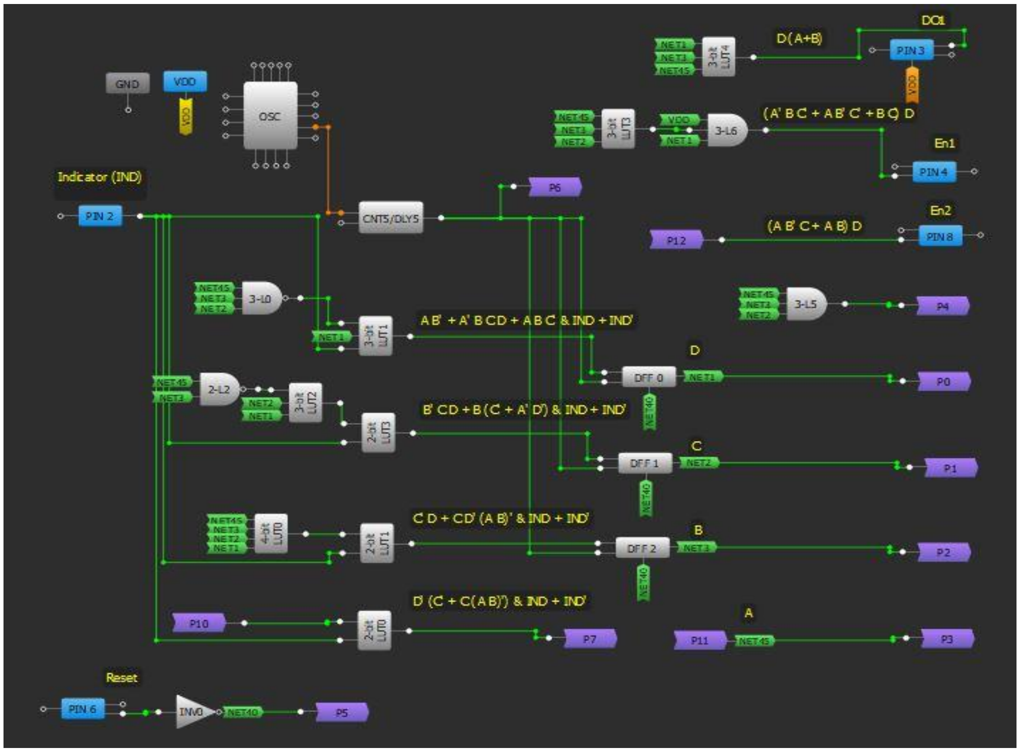

Fig. 11. Example 3 GreenPAK Design Matrix-0.

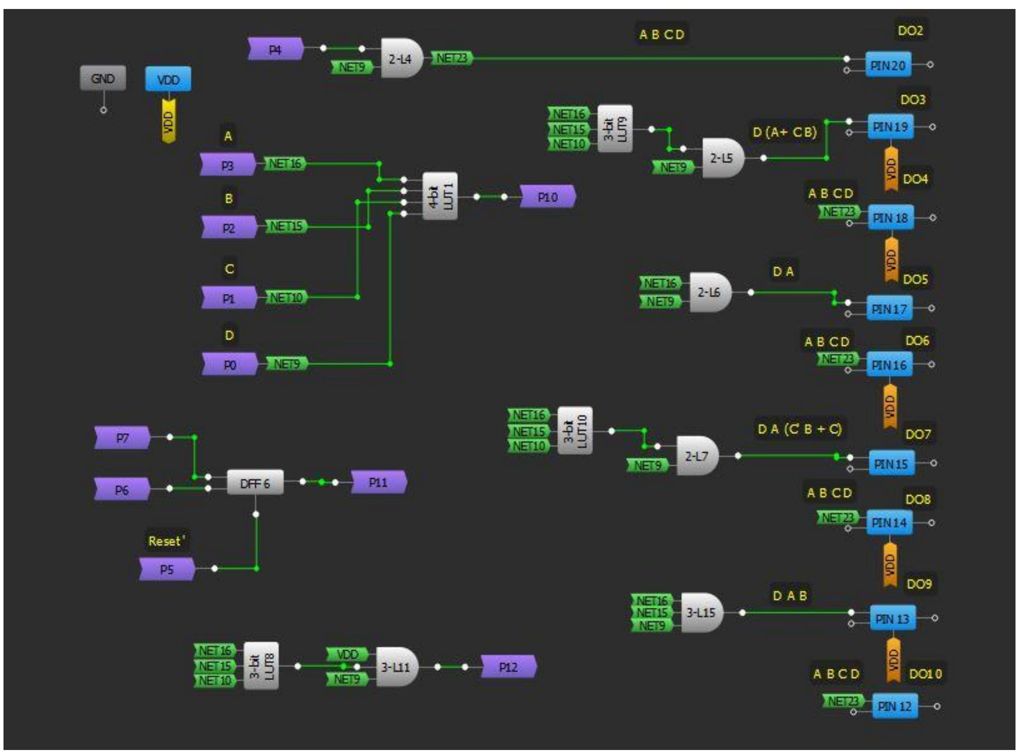

Fig. 12. Example 3 GreenPAK Design Matrix-1.

Experimental Results

A convenient way to test the designs of examples 1, 2 and 3 is through experimentation and visual inspection. The temporal behavior of each scheme is analyzed using a logic analyzer and the results are presented in this section.

Fig. 13 shows the temporal behavior of different output signals for example 1 whenever the indicator is turned on (IND = 1). It can be observed that the signals for the output pins DO1-DO5 sequentially turn on after the other after a set time period expires in accordance with Table 2. The pattern of the signals provided to the pins DO6-DO10 is also similar.

The driver enable (En) signal turns on when any of the signals DO1-DO10 are turned on and otherwise it is off. During the animation, whenever the indicator signal goes low (IND = 0), the En and DO10 signals turn on and remain logical high. In short, the results meet the requirements and validate the theoretical proposals for example 1.

In Fig. 14, the timing diagram of different output signals for example 2, with the indicator signal turned on (IND = 1), is depicted. It is observed that the signals for the output pins DO1-DO5 are turned on alternately in a sequence after some time period in agreement with Table 2. The pins DO1, DO3 and DO5 remain low, whereas the signals for the DO2 and DO4 alternately turn on sequentially. The same patterns for DO6-DO10 are also observed (not shown in the figure due to limited number of analyzer inputs).

Whenever any of the signals DO1-DO10 are on, the driver enable (En) signal also turns on which otherwise remains off. Throughout the animation, whenever the indicator signal goes low (IND = 0), the En and DO10 signals turn on and remain logical high. The results meet the requirements and agree exactly with the expected performance for example 2.

Fig. 15 shows, the timing diagram of different output signals for Example 3, with the indicator signal turned on (IND = 1). It can be observed that the signals for the output pins DO1-DO7 turn on as shown in Table 2. Moreover, the pin DO9 signal also behaves according to Table 2 (not shown in the figure). Pins DO2, DO4, DO6, DO8, and DO10 remain low.

The En1 turns logical high whenever a signal from DO1, DO3 and DO5 is on and En2 turns logical high whenever a signal from DO7 and DO9 goes high. During the entire animation, whenever the indicator signal is goes low (IND = 0), all the output signals: En1, En2 and DO1-DO10 turn on and remain logical high. Therefore, it can be concluded that the results fulfill the requirements and agree with the predicted results for example 3.

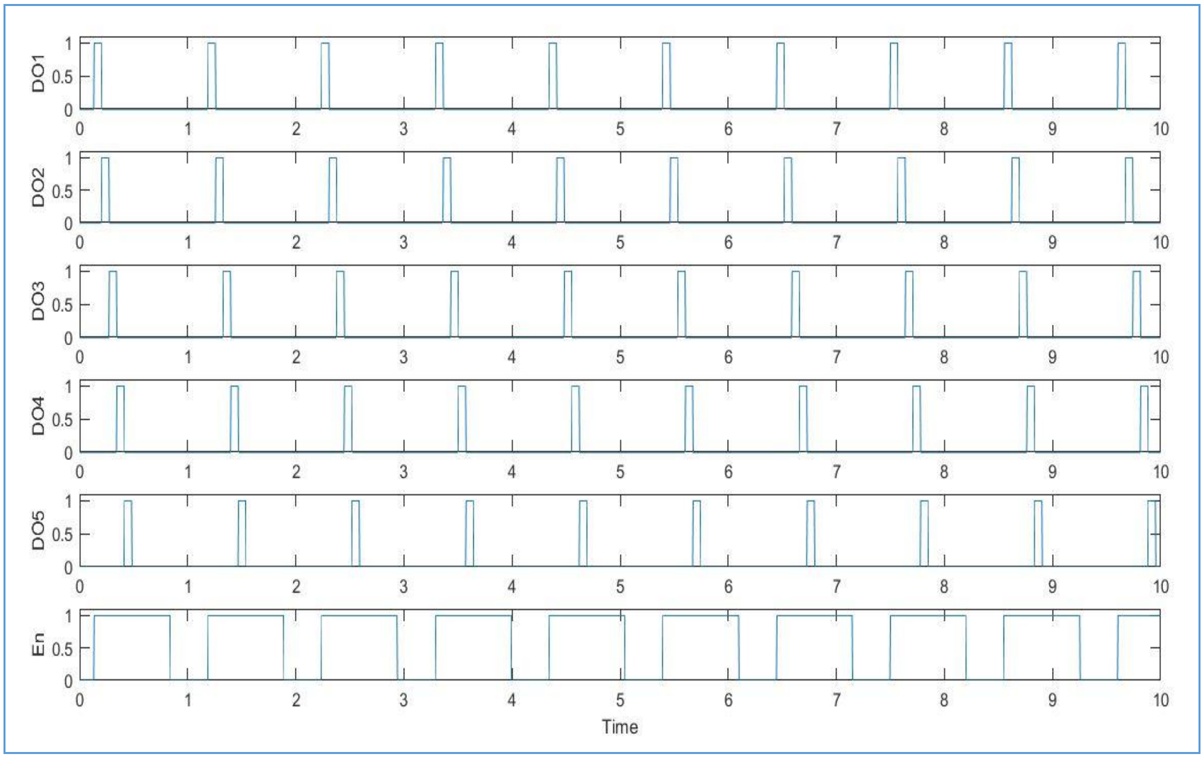

Fig 13. Temporal behavior of output signals for example 1 with IND = 1.

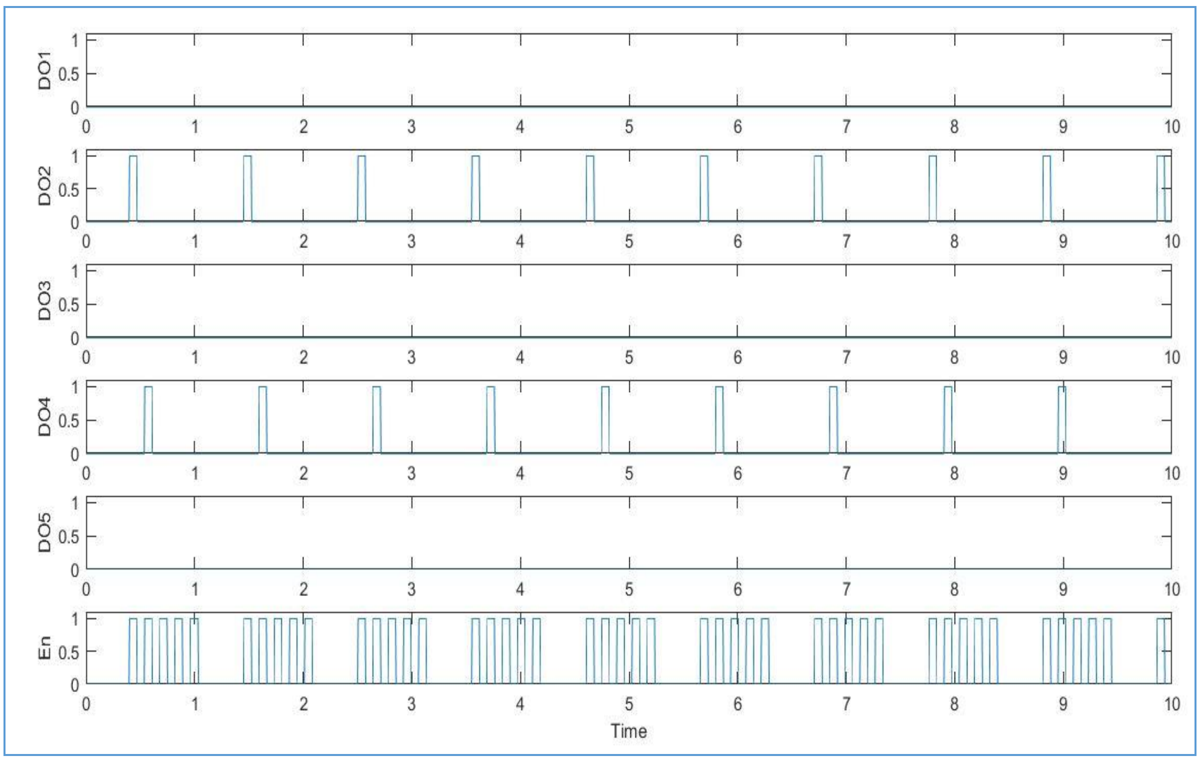

Fig. 14. Temporal behavior of output signals for example 2 with IND = 1.

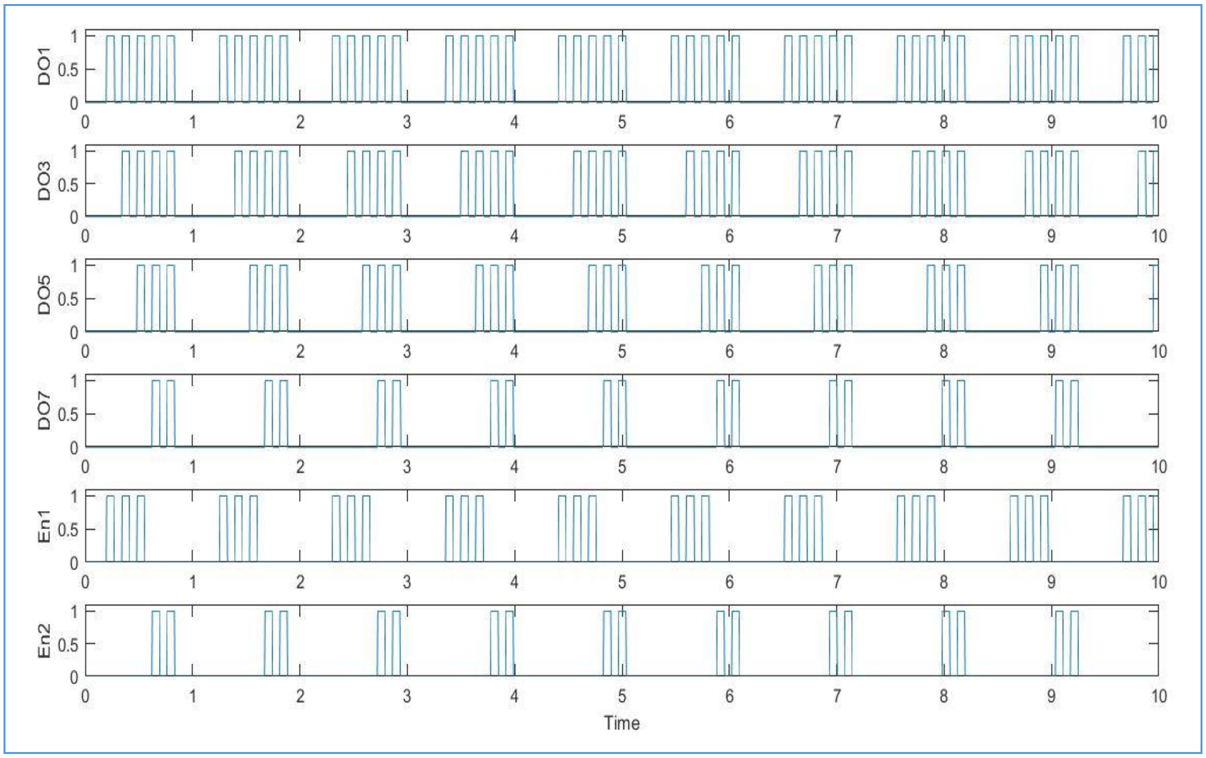

Fig. 15. Temporal behavior of output signals for example 3 with IND = 1.

Conclusion

A detailed description of various automotive turn-signal schemes with animation has been presented. A suitable Dialog CMIC SLG46620 was chosen for this application since it is available in the TSSOP package which is recommended for harsh environmental applications such as automotive.

Two major schemes, using single- and multi-channel automotive drivers, were presented to develop flexible sequential LED animation models. Appropriate finite state Moore machine models were developed to generate the desired animations. For validation of the developed model, convenient experiments have been carried out. The experimental results confirmed that the functionality of the developed models agreed with the theoretical design.

For related documents and software, see the GreenPAK landing page.[1] To download the free GreenPAK Designer software see reference 2 and to open the .gp files, see reference 3 and view the proposed circuit design. Use the GreenPAK development tools[4] to freeze the design into your own customized IC in a matter of minutes. Dialog Semiconductor provides a complete library of application notes[5] featuring design examples as well as explanations of features and blocks within the Dialog IC.

References

1. GreenPAK Programmable Mixed-signal Matrix page.
2. GreenPAK Designer Software, Software Download and User Guide, Dialog Semiconductor
3. AN-CM-xxx Automotive Turn Signal with Animation.gp, GreenPAK Design File, Dialog Semiconductor
4. GreenPAK Development Tools, GreenPAK Development Tools Webpage, Dialog Semiconductor
5. GreenPAK Application Notes, GreenPAK Application Notes Webpage, Dialog Semiconductor

6. SLG 46620 datasheet.

About The Authors

Muhammad Saqib's work focuses on power electronics, embedded systems and instrumentation. He also works on CMICs and develops app notes for Dialog Semiconductor. Muhammad received the B.S and M.S degrees in electrical engineering from National University of Computer and Emerging Sciences (NUCES), Lahore. Muhammad can be contacted at saqibfastian@gmail.com.

Aamir Hussain Chughtai is currently a Ph.D. student specializing in electrical engineering at LUMS, Lahore. His interests include signal processing, artificial intelligence and power electronics. He also provides services for app notes development to Dialog Semiconductor. Aamir received the B.Sc. degree in electrical engineering from UET, Lahore and M.Sc. degree in electrical engineering from LUMS, Lahore. Aamir can be contacted at chughtaiah@gmail.com.

For more information on LED driver design, see How2Power's Design Guide, locate the Power Supply Function category and click on "Lamp ballasts and LED drivers".