Application Note

LED-based Multiplexed Display Implementation Using TI Programmable Logic Devices

Gerardo Leyva-Hernandez

ABSTRACT

Alphanumeric or pixel-oriented multiplexed displays can be handled by an electronic device with small quantity of GPIOs. Each of the columns of such display driven one at a time, so the combination of high frequency and persistence of vision both together create the viewer's illusion that the full display is active all the time. The TI Programmable Logic Devices™ (TPLD) have the resources needed to be configured to generate multiple blocks needed in the use case such the character generator, the CLK and synchronization signals and the multiplexer column selector as well.

This application note shows how the use State Machines and true – tables without any logic equations. The TI Programmable Logic Devices tool provide friendly-user options to implement designs by filling tables (FSM) and chose options (LUTs) instead of designing logic equations and Logic Algebra, which can be a time-consuming task and error prone. Texas Instruments PLD is an effective design to replace dozens of registers and logic gates in a given design, minimizing the Bill of materials, space and power consumption.

Table of Contents

Trademarks

TI Programmable Logic Devices™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

When several alphanumeric display or pixel-oriented displays must be managed by a controller, the number of input-outputs (IOs) and the power needed for such controller can be a challenge. This takes 56 IOs to control 8-digits, 7-segment display if all the digits are active simultaneously. However, if a multiplexed display is implemented in the same size, just 15 IOs can accomplish the same job and, moreover, with less power.

In a Multiplexed Display, the LEDs in the segments or dots are organized in such way that the rows and the columns provide the data information (asserted high) and digit control (asserted low), respectively, as the Display Block diagram in Figure 1-1 shows.

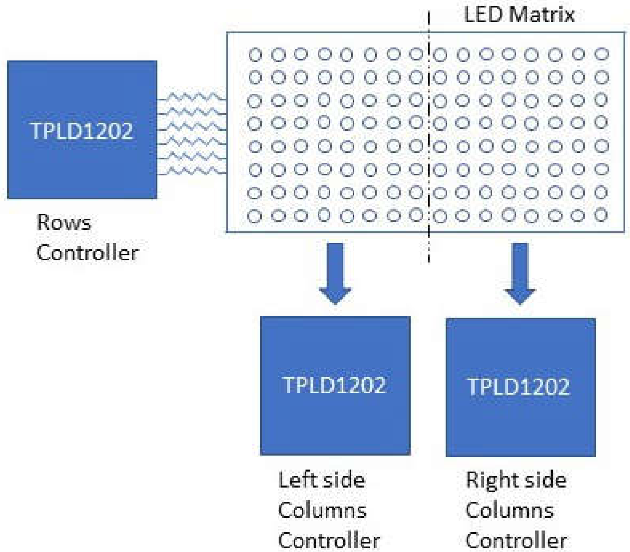


Figure 1-1. Multiplexed Display Block Diagram

Each of the columns of a multiplexed display are driven one at time, the combination of high frequency and persistence of vision both together create the viewer's illusion that the full display is active all the time. The activation sequence of of each of the rows is typically made more than 50 times for second, as is shown in the Figure 1-2.

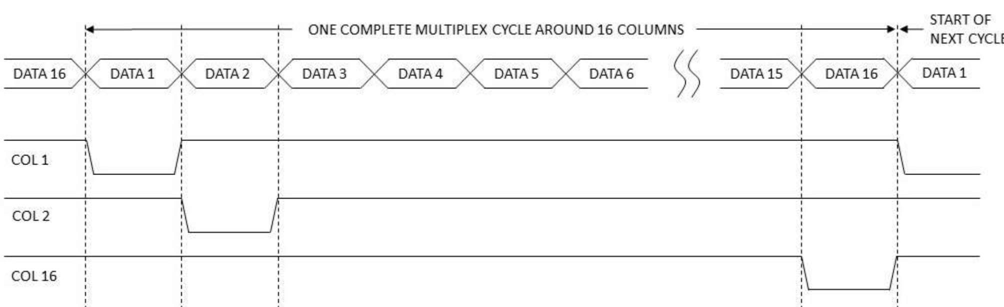


Figure 1-2. Multiplexed Display Time Diagram

The TI programmable Logic Device (TPLD) can be configured to generate multiple block needed in the use case such the character generator, the CLK and synchronization signals and the multiplexer column selector as well. The LED-based monochrome Multiplexed Display in this example drives up to 16-columns 6-dots each, or this can be used for an 8-digits 7-segment display for alphanumeric messages or numbers, respectively. In the first case the design is using 3 TPLD and in the last case just 2 TPLD devices are needed. In both cases one of the TPLD is in charge of generate the data to be shown in the displays and the synchronization signals as well. The remaining TPLD devices are in charge of generating the columns multiplexer.

2 Implementing the Multiplexing Display Controller in TPLD

TPLD devices have a lot and diverse micro – cells to implement all the required blocks for this application, such as oscillator, Finite-states Machine (FSM), True – Tables (LUTs), input - output pins (IOs) and more. In the next sections all the details about each of the blocks needed are described.

2.1 Data and Synchronization Generator

In this design, a pre-defined message is going to be show in a 16x6 graphic display, this means that such information needs to be saved somewhere inside the TPLD device. Additionally, each of the rows must be activated one at time, so, a synchrony system must also be included to keep control of the internal elements in this device as well as the external TPLD column's controllers.

Regarding the data that can be displayed, a 16-states finite state machine can be a good way to generate each of the values at time required for each of the rows of the dot-matrix display. The TPLD1202 includes an 8-states FSM built-in in hardware, so 16-states FSM can be implemented using this FSM plus a secondary FSM built with true – tables and D flip-flops. In addition, the secondary FSM can also generate the synchronization signal required.

Figure 2-1 shows the message that can be displayed and the row data involved as well. Notice that the left section is handled by the FSM and the right one is handled by the sequencer plus LUTs as well.

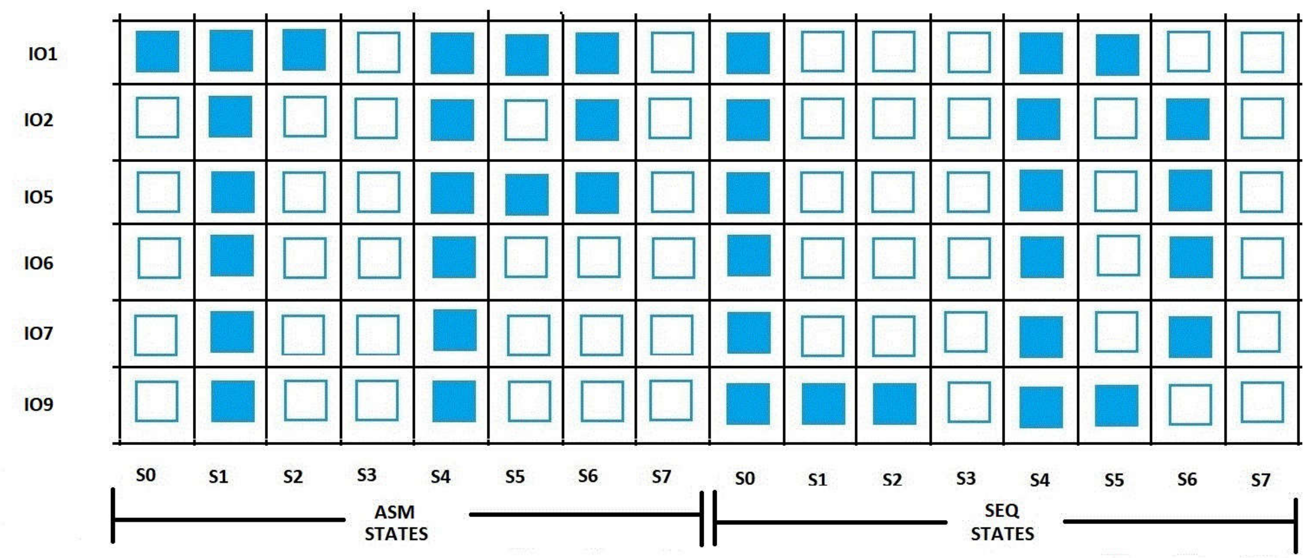


Figure 2-1. Message Shown in the Multiplexed Display

2.2 Configuring the FSM in Interconnect Studio

The process of configure the FSM in interconnect Studio is straightforward. Once the FSM macro-cell is put in the edition area and clicked on, the configuration options are shown including the number of states needed, the output's value for the state and where is transitioning for. The user must configure all of the states needed. Figure 2-2 shows a given state with the features already defined.

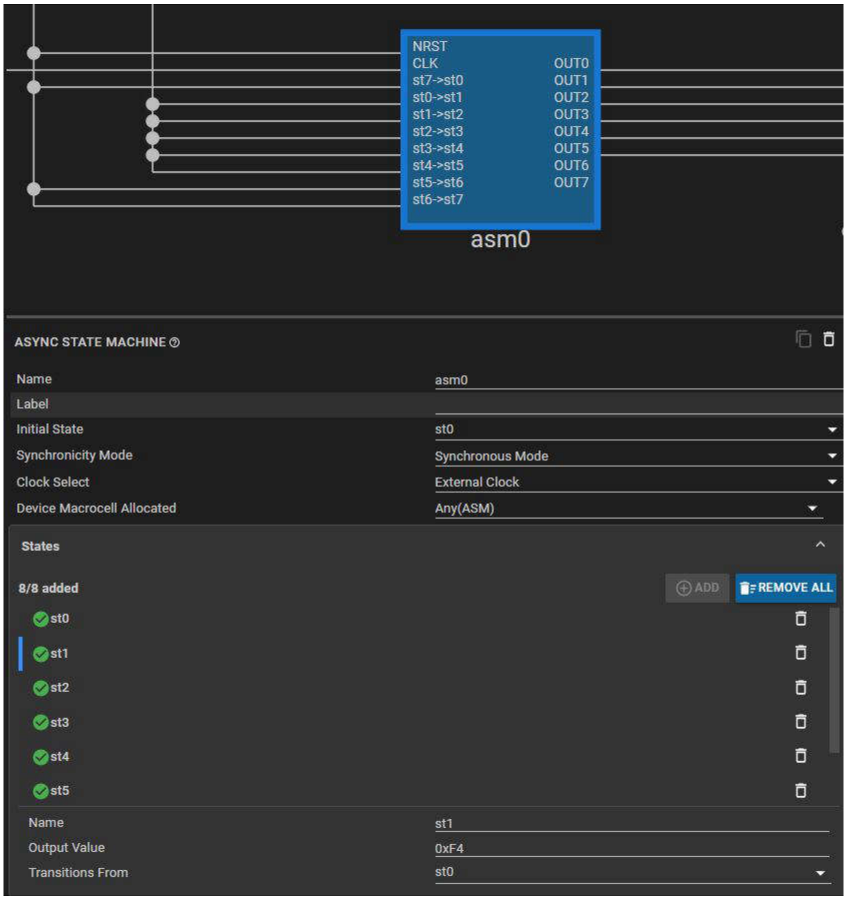
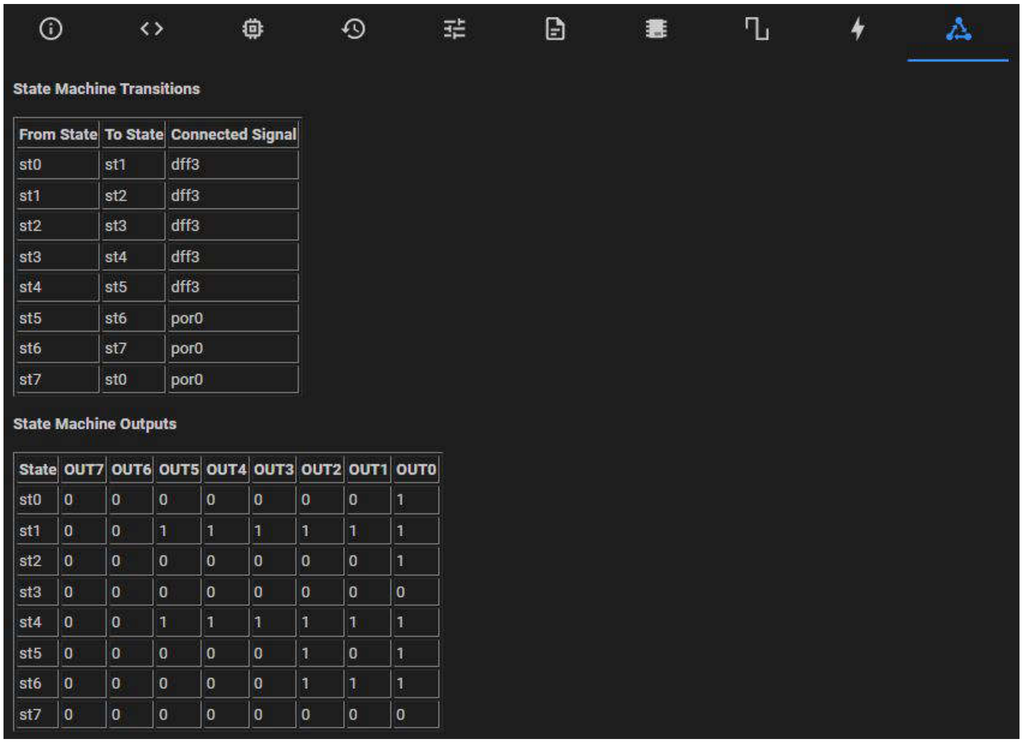


Figure 2-2. Finite State Machine Definition

One additional helpful feature provided by ICS is the ability of generate both the transitions and the outputs for the FSM previously defined, in the way the user can always make a double check if the FSM definition is capturing the expected behavior. To get that, the user can click in the State Machine icon located at the bottom right of the screen, then ICS can present a couple of tables showing the State Machine both Transitions and Outputs as well as can be seen in the Figure 2-3. The message “TP” can be seen 90 degrees rotated clockwise in the State Machine Outputs table.



State Machine Transitions

From State	To State	Connected Signal
st0	st1	dff3
st1	st2	dff3
st2	st3	dff3
st3	st4	dff3
st4	st5	dff3
st5	st6	por0
st6	st7	por0
st7	st0	por0

State Machine Outputs

State	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
st0	0	0	0	0	0	0	0	1
st1	0	0	1	1	1	1	1	1
st2	0	0	0	0	0	0	0	1
st3	0	0	0	0	0	0	0	0
st4	0	0	1	1	1	1	1	1
st5	0	0	0	0	0	1	0	1
st6	0	0	0	0	0	1	1	1
st7	0	0	0	0	0	0	0	0

Figure 2-3. Finite State Machine Summary

2.3 Configuring TYPE-D Flip-Flops (DFF) in Interconnect Studio

The next section of the Multiplexed display's data generator can be implemented with a classical Moore FSM including next-state logic, memory an output logic section. But, because the FSM is going to run all the time form ST0 to ST7, we can implement a count-down counter with DFF saving the next-state logic in some kind of sequencer. So finally, we can add the output logic based on true – tables fashion, and implemented in LUTs macrocells in order to generate the given data needed accordingly to the state of the sequencer, as is shown in the Figure 2-4. Regarding the sequencer, the same figure is showing how a DFF can be defined to operate as Toggle FF, and therefore, all TFFs working together as a countdown counter.

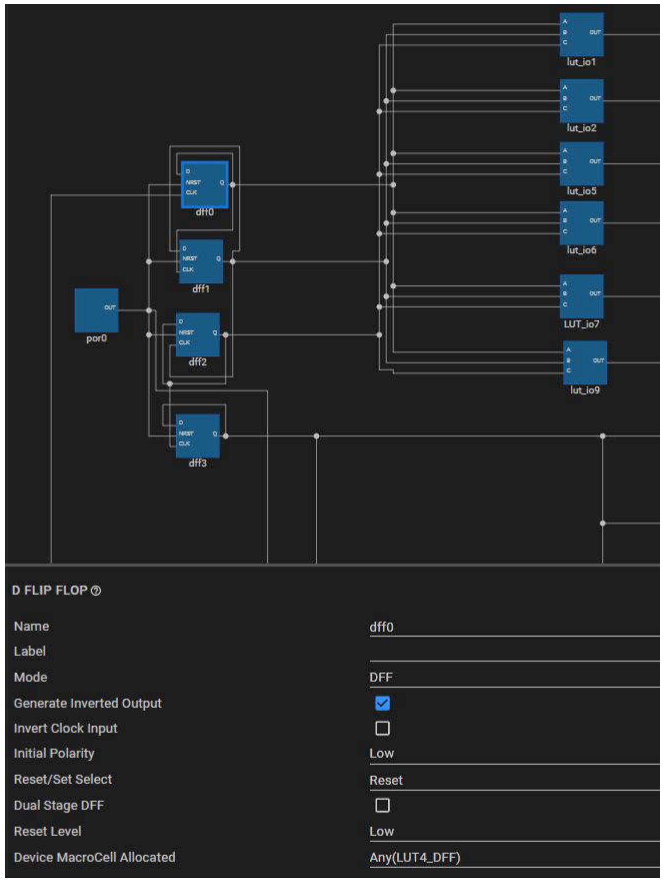


Figure 2-4. Countdown Counter Implemented With T-FFs

2.4 Configuring the True – Tables in Interconnect Studio

ICS let the user handle the combinational logic in three different ways: using standard logic gates, introducing logic equations or defining the outputs of true tables, independently if the logic function is using 2, 3 or 4 inputs.

The Figure 2-5 is shown a 3-input true table for one of the outputs in the current design. The only thing the user must take care about is to checked outputs indicating the ones that generate logical high given the values of the inputs (C, B and A in the figure), instead of use Logic Algebra to minimize the logic equations. This feature let the tool in charge of generate the proper configuration on the logic macro-cell, helping to the user avoid the use of logic equations, that in some cases is a time-consuming task. Of course, the user has the chance of use his own logic equations if that is preferred.

Each of the LUTs in Figure 2-4 have assigned an IO. Figure 2-5 is showing the true – table configuration for the LUT_IO1. The first row in the sequencer section must generate the values “10001100” as is shown in the Figure 2-1. The same values are present at the Figure 2-5, but from bottom to top, in other words, from input’s combination *111* to *000*.

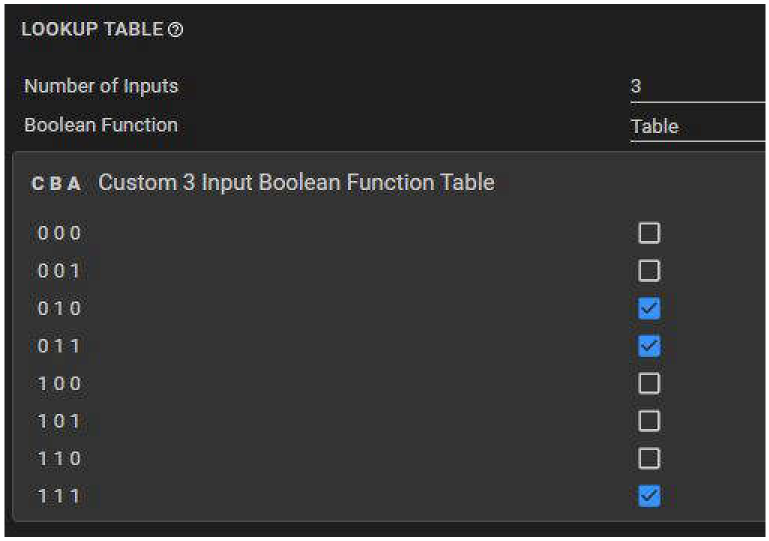


Figure 2-5. ICS Handling Look-up Tables

Notice that the current design is handling 6 segments or rows instead of 7 or 8. However, depending of the data that can be displayed, more than one outputs can share the same logic function, allowing to the user expand the rows to 8 instead of just 6.

2.5 Configuring the Oscillator in Interconnect Studio

The oscillator's dividers can be utilized to generate a wide variety of frequencies. The circuit shown in Figure 2-6, configured in InterConnect Studio (ICS), shows an example of using the 2KHz oscillator in the TPLD1202 to generate a 62.5Hz square wave with 15/16 duty cycle. To achieve this, the oscillator pre-divider is set to divide by 2, changing the base frequency to 1KHz. The serial receivers are generating a duty cycle of 15/16 of that signal, resulting in a 16mS period low asserted signal for each of the columns.

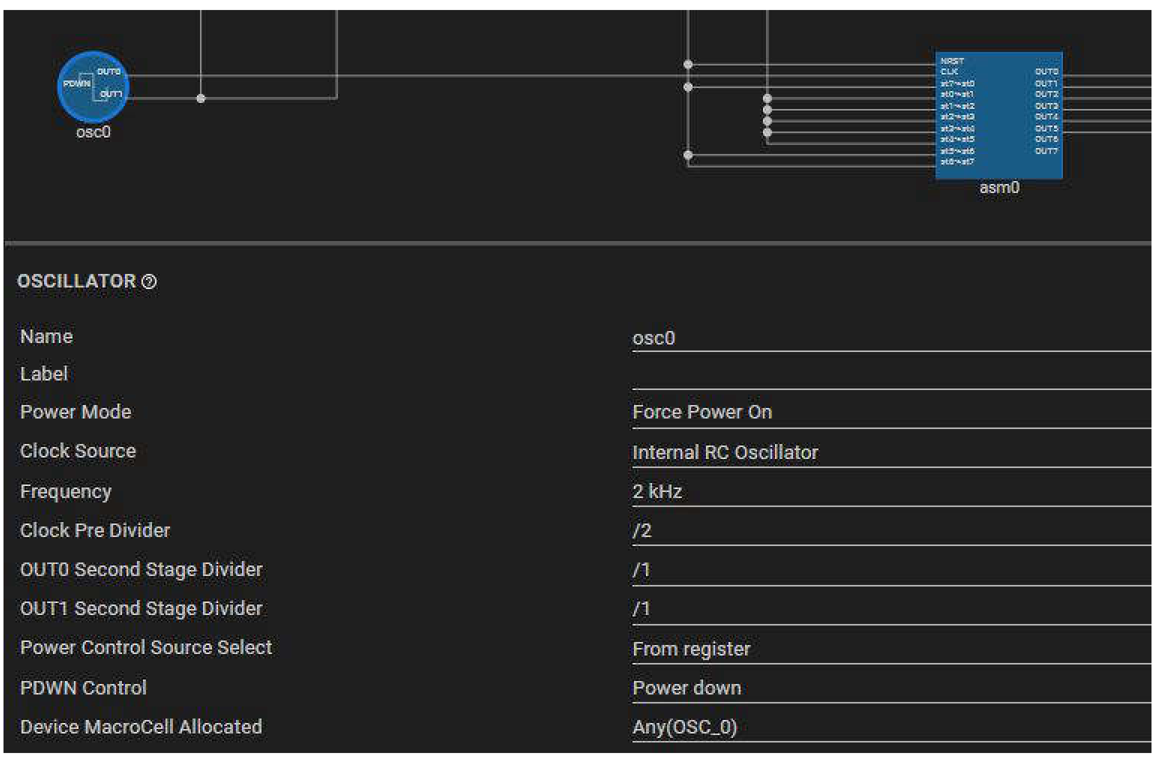


Figure 2-6. OSC Configuration in ICS

3 Summary

Digital design can be easily implemented in TPLD devices. In this Application note was shown how the use State Machines and true – tables without any logic equations. The ICS tool provide friendly-user options to implement designs by filling tables (FSM) and chose options (LUTs) instead of designing logic equations and Logic Algebra, both of them activities that use to be time consuming task and error prone.

Texas Instruments PLD is an effective design to replace dozens of registers and logic gates in a given design, minimizing the bill of materials, space and power consumption. By offering several types if analog and digital resources in the same package such as OSC, DFFs, FSM, GPIOs, TPLD can help to get a bigger integration and a cost-effective design.

4 References

- Texas Instruments, *Programmable Logic Devices (PLD)*.