MBI5050 Application Note

Foreword

In contrast to the conventional LED driver which uses an external PWM signal, MBI5050 uses the embedded PWM signal to control grayscale output and LED current, which makes it more outstanding in grayscale performance and suitable for LED full-color display. Besides, it also features 4K-bit SRAM which is applicable in LED full-color scan display, providing higher LED utilization rate, higher color level and refresh rate.

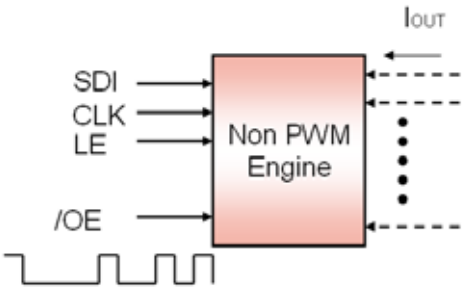

Fig.1 The traditional LED driver

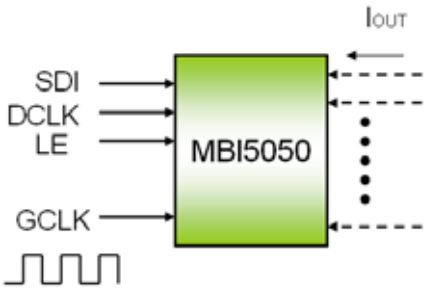

Fig. 2 PWM-embedded LED driver

Figure 1 shows the operation scheme of the traditional LED driver. It controls color levels or LED brightness by sending different PWM pulse width through $\overline{OE}$ pin. The PWM signals will suffer distortion and decay in long range transmission, causing deteriorated high color level.

Figure 2 shows the scheme of PWM-embedded LED driver. It controls LED color gradation and brightness by triggering the internal PWM counter to send PWM pulse by means of GCLK signal. Like the traditional LED driver, the GCLK signal also suffers distortion and decay in long range transmission, but the distorted and decayed GCLK signal will not affect the quality of high color gradation.

Since GCLK signal is the clock of PWM counter, its pulse width does not affect the internal PWM engine. In addition, only one group of grayscale data needs to be sent to MBI5050 each frame, so there is no image data transmission delay issue, dramatically enhancing the LED utilization and screen refresh rates.

The operations of MBI5050 and the traditional MBI5026 are quite different, such as the input methodology of image data and the setting of grayscale data. The detail operation is listed in the following chapters.

Floor 6-4, No. 18, Pu-Ting Rd., Hsinchu, Taiwan 30077, ROC.

Principle of Operations

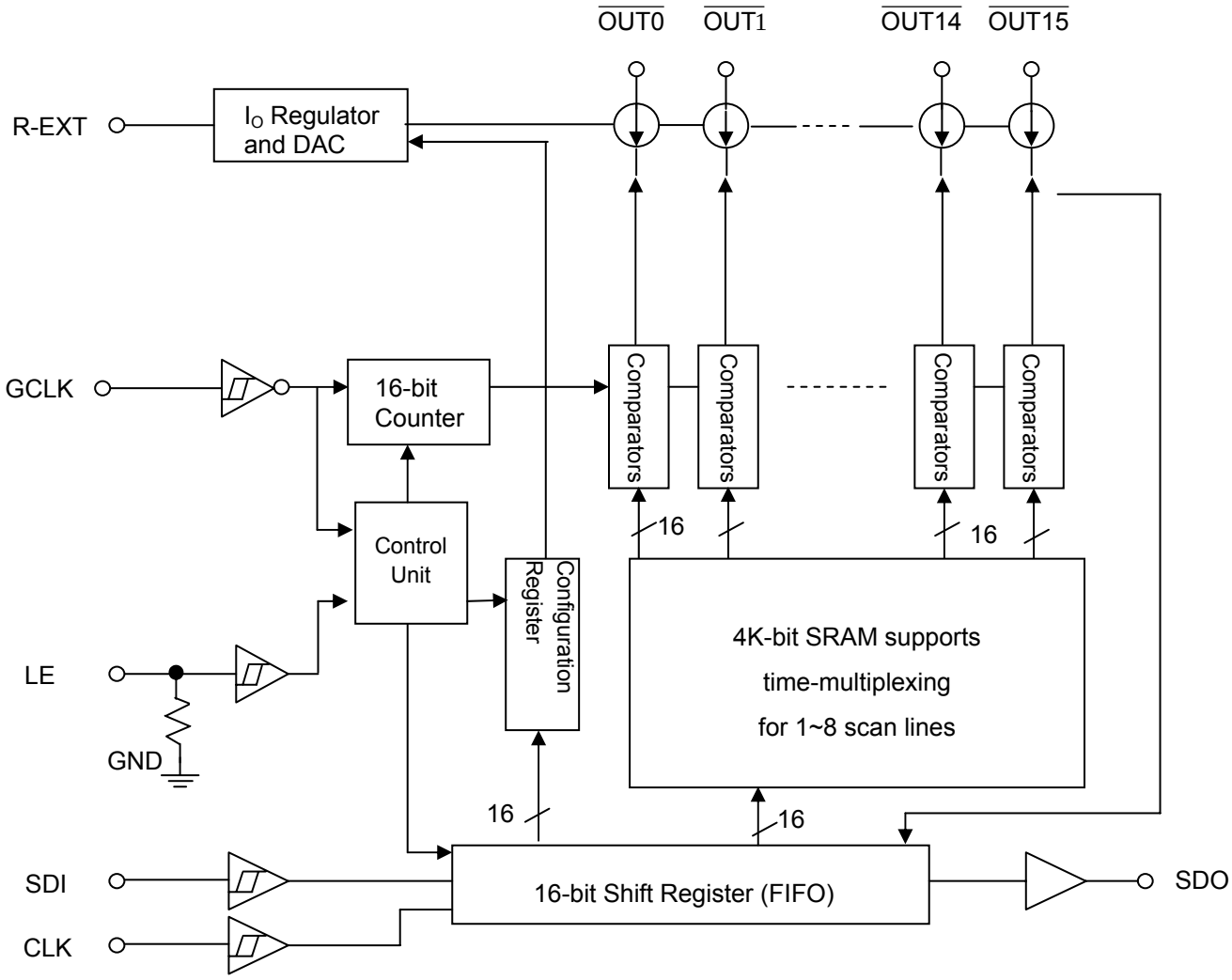

Fig.3 MBI5050 Block Diagram

Fig.3 shows the block diagram of MBI5050, the function of each pin is described below

Input Pins of Control Signals

DCLK (Data Clock Input pin): DCLK samples the signals of SDI and LE at its rising edge

SDI (Grayscale Data Input pin): SDI functions with DCLK

LE (Latch Enable): LE functions with DCLK

The grayscale data can be transmitted to MBI5050 by means of DCLK, SDI and LE. At the rising edge of DCLK, MBI5050 reads one bit of data into the internal 16-bit shift register, and LE controls the data latch of the internal 16-bit shift register.

Output Pin of Control Signal

SDO(Serial Data Output pin): The SDO of prior stage MBI5050 is connected to the SDI of latter stage, and then the grayscale data can be transmitted to the next MBI5050.

Input Pin of PWM Counter Clock

GCLK(PWM Counter Clock Input pin): The frequency of GCLK determines the counting speed of PWM counter, who is the output frequency of $\overline{OUT0}$ ~ $\overline{OUT15}$.

Pin for Setting Output Current

R-EXT: An external resistor is connected to R-EXT and the resistance determines I_{OUT}

Output Pins of Constant Current

$\overline{OUT0}$ ~ $\overline{OUT15}$: I_{OUT} output channel pins. Every channel can be connected to one or more LEDs depending on the circuit design.

The Setting of configuration register when Cascading MBI5050

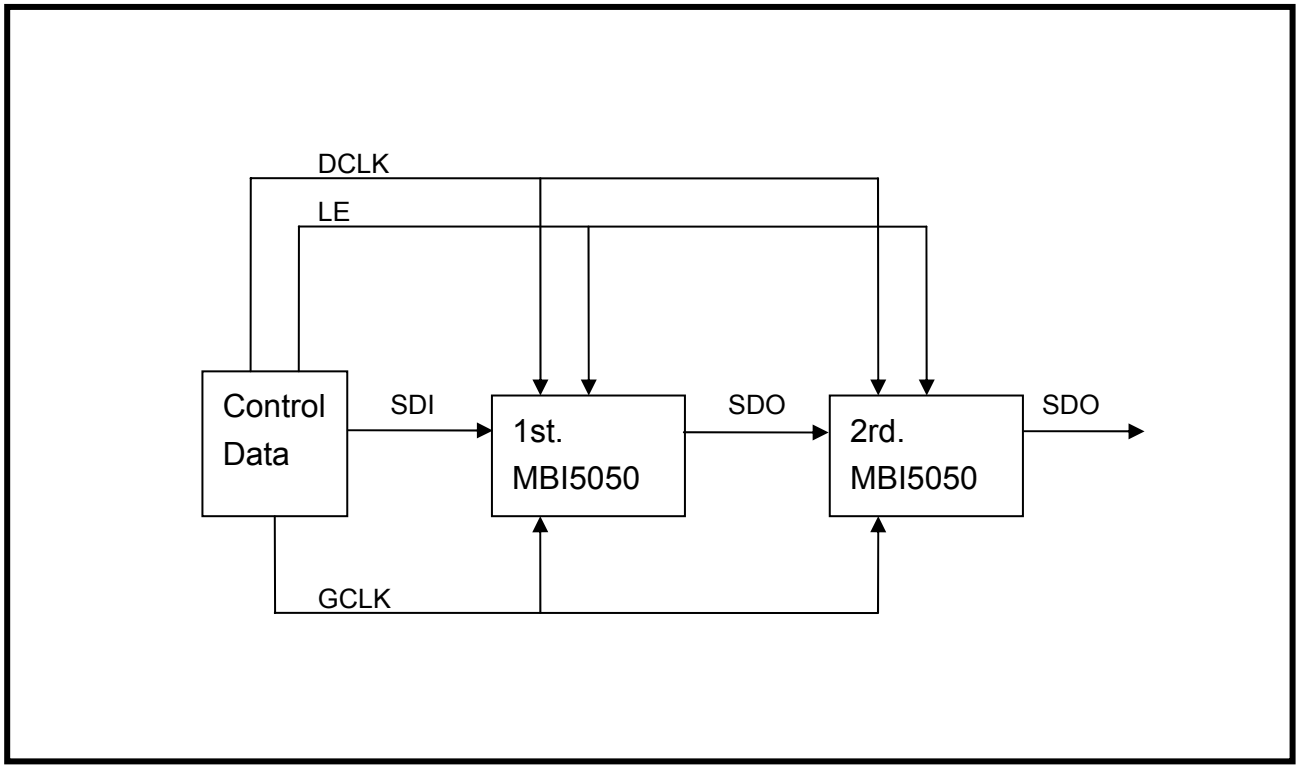

Fig.4 The cascade configuration of two MBI5050s

The configuration register embedded in MBI5050 is a readable and writable register. The register value determines the operation mode of MBI5050. For example, the selection of GCLK signal source, current gain setting and the scan line number are all achieved by altering the register value.

Fig.5 illustrates how to control the configuration register based on fig 4.

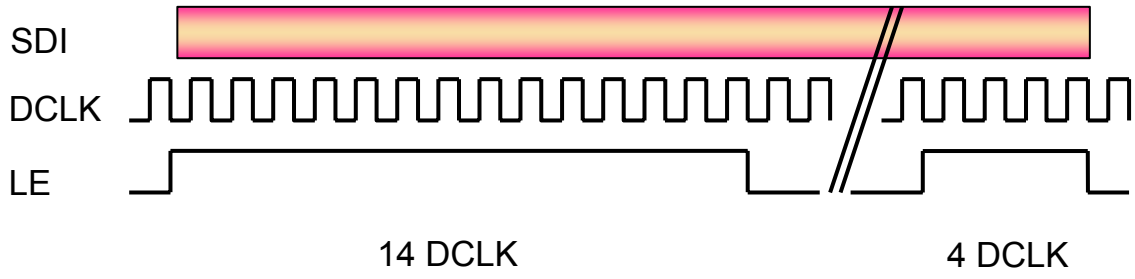

Fig.5 A completed “write configuration command” includes a “pre-active command” of LE asserts 14 DCLKs rising edges and a “write configuration command” of LE asserts 4 DCLKs rising edges.

First, to avoid the unstable signal or external noise makes MBI5050 get error message during controller enables, users must add a “pre-active command” before the commands of “write configuration”, “Enable all outputs” and “Disable all outputs”. The “pre-active command” is LE goes to high and asserts 14 DCLKs rising edges. When LE goes to low, it’s recommended to add a DCLK to trigger LE low level. This is very important and will be emphasized constantly in this application note.

Write Configuration

At the last 4 bits in SDI data string, the LE goes to high and asserts 4 DCLKs rising edges. The data string of each MBI5050 is 16 bits. If there are two MBI5050 in cascaded, as fig.4 shows, the data string is 32 bits, whose LE is low at previous 28 DCLKs rising edges, and then goes to high at last 4 DCLKs rising edges. After that, LE goes to low again, and please add a DCLK to trigger LE low level.

Data sequence is from the second MBI5050 to the first MBI5050. Each grayscale data sequence is from MSB to LSB. The effective trigger point of MBI5050 happens in LE falling edge. So, the effective data is from LE falling edge and shift back to 32 DCLKs. More detail information about “Configuration register” parameter please refers to the section of “Definition of Configuration register” in MBI5050 Datasheet.

Read Configuration

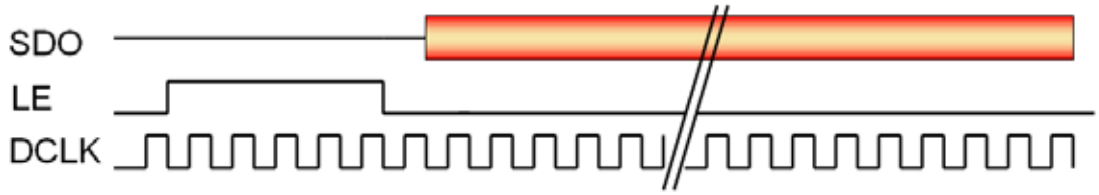

Fig.6 A completed ”Read Configuration Command”

Fig.6 shows the timing diagram of read configuration. Once the command of LE keeps high during 5 of DLCK rising edges, and then goes to low at 6th rising edge is existed, the command of read configuration will be enabled. The embedded 16-bit shift register will load the data in configuration register. The sequence of output data is from MSB (bit F) to LSB (bit 0) with the rising edge of DCLK.

When enter “Read Configuration” command, the data string of SDI will not change the internal settings and configuration register data of MBI5050.

The Setting of Grayscale Data when Cascading MBI5050

The length of grayscale data is related to the scan line number of multiplexing application. Thus, please set the configuration register first before input the grayscale data.
If there are N piece of MBI5050 in cascaded, the grayscale data is 16 bits x N when each data latch.

For example:

1 pcs of MBI5050 needs 16 bits (=16 bits x 1) when each data latch.
2 pcs of MBI5050 need 32 bits (=16 bits x 2) when each data latch.
3 pcs of MBI5050 need 48 bits (=16 bits x 3) when each data latch, and etc.

No matter how many MBI5050 in cascaded, the data can be latched until all the input data gets ready.

Fig.7 illustrates the different grayscale data of two cascaded MBI5050 in 8-line scan application.

Take the circuit diagram of figure 4 as example, in 8-line scan application, serial connected two MBI5050 ICs by different grayscale data.

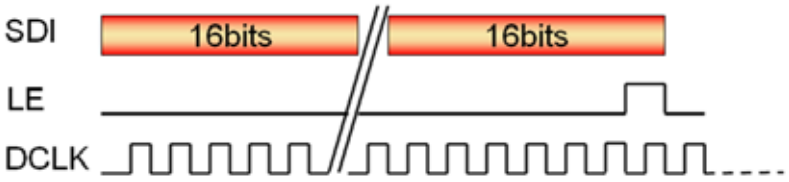

Fig.7 Each data latch inputs 32 bits data to each channel of two cascaded MBI5050. one data latch is input data for one channel. Take 2 serial connected MBI5050 as example. Input data will be 32 bits.

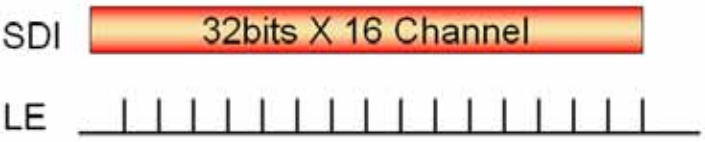

Fig.8 To input the grayscale data of two cascaded MBI5050, it needs 16 times of data latches.

Fig.9 For 8-scan lines application, the total grayscale is 32x16x8 bits.

The length of grayscale data is 16 bits for one output channel. So, it is 256bits (=16 bits x 16) for 16 output channels in one MBI5050. And it is 512bits (=32 bits x16) for 16 output channels in two serial connected MBI5050. The procedure of input data is from OUT15, OUT14, OUT13, …, to OUT0. After the first scan line has been completed, input the data of the 2nd scan line to the 8th scan line one by one.

The completed data length for this example is 4096 bits (=32 bits x16 x 8). The sequence when every data inputs, is from MSB to LSB. Each input data are sent from the most significant bits (MSB) and the procedure is from bit15, bit14, bit13, …, to bit0.

No matter the counting mode is 16bits or 14bits, the input data length for each single scan is 256bits.

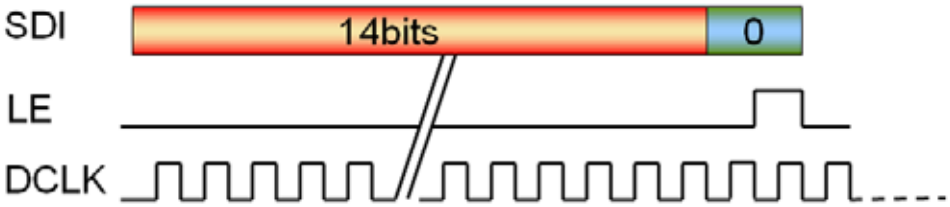

Fig.10 Timing diagram of 14 bits counting mode

In 14bits counting mode, bit1 and bit0 are insignificant, but since the internal shift register of MBI5050 is fixed to 16 bits, these two bits have to be set to “0” to make sure the next MBI5050 can receive the correct grayscale data, as fig.10 shows. The operation of LE is same as 16 bits counting mode.

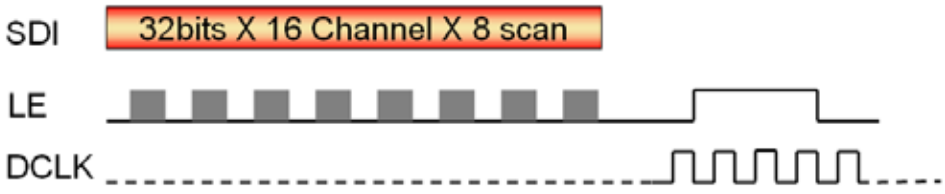

Fig.11：Vertical sync command is that LE is asserted three of DCLK rising edge. Suggest at least one more DCLK before and after the Vertical Sync command.

Use “Vsync” command to update the frame. Vsync command is that LE is asserted 3 DCLKs rising edges. Data latch is just to input the data to MBI5050, and these data which saved in SRAM will not be updated to output immediately. Till to receive the “Vsync” command, the frame data will be replaced, as fig.11 shows.

Users should send Vsync command after 50 GCLKs of the last “Data Latch” command. The display data will not start until Vsync command is ready. The GCLK must be stopped before Vsync command is set, till to the end of Vsync command.

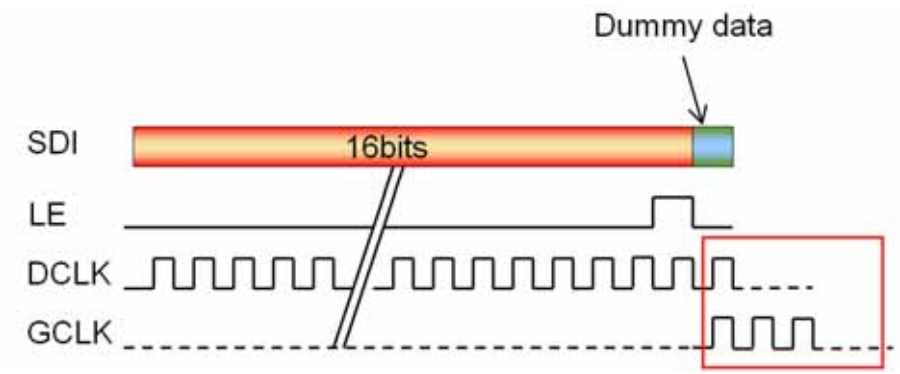

Fig.12 At least one more DCLK after data latch is required, and at least 3 GCLKs between the two data latch is required.

After data latch, at least one DCLK is necessary, also the data can't be inputted to MBI5050 when GCLK is stopped. 3 GCLKs at least is needed between each data latch to write the grayscale data in shift register to SRAM correctly, refer to fig.12.

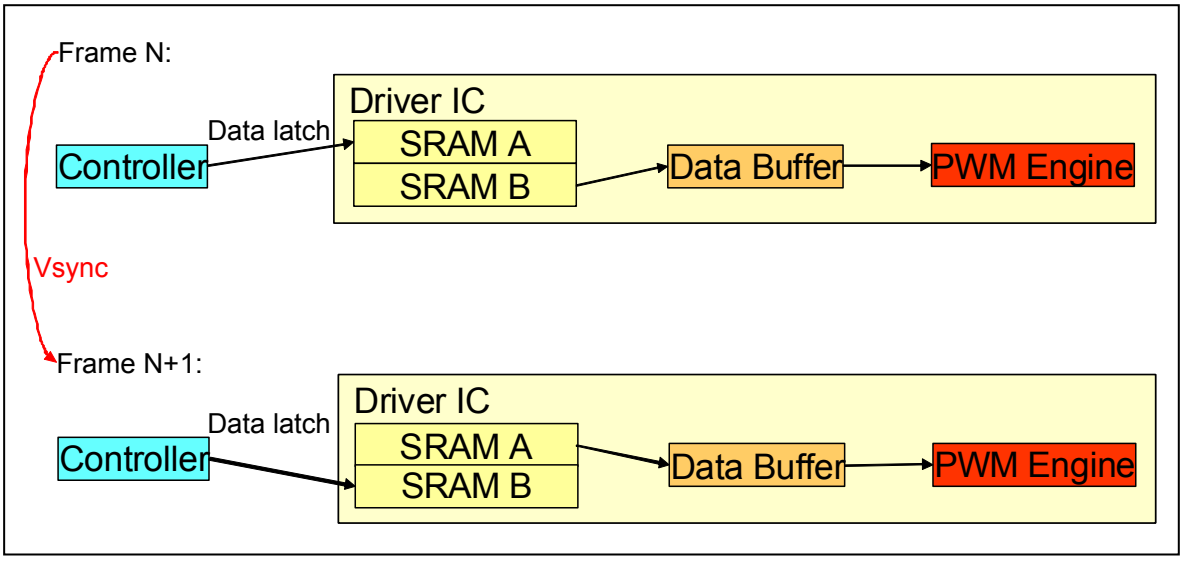

Fig.13 MBI5050 transmission and architecture

MBI5050 stores the gray scale data in a 4k-bit SRAM. The SRAM has two banks for interleaving reading and writing data frame, refer to fig.13. Since the SRAM is composed of two-bank structure, users can send the gray scale data of next frame while current frame is playing. These two banks are in charge of writing and reading data, and starts from scan line 0, channel 15. After Vsync command, the SRAM will switch the function of these two banks to reading and writing.

Control Signal in Time-multiplexing Application

Following take an example by using six MBI5050s in cascaded and 8 scan lines time-multiplexing application, the circuit is shown as fig.14. Eight rows of LEDs share the same LED supply voltage, V_{LED}. By changing the MOSFET switch, the voltage of V_{LED} takes turns lighting up eight rows of LEDs.

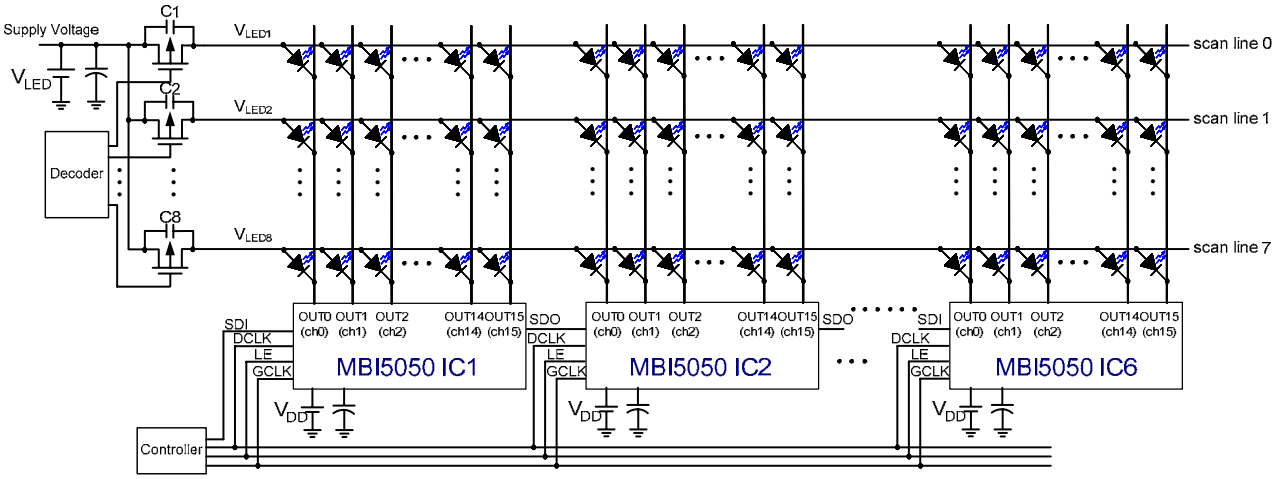

Fig.14 Time-multiplexing circuit diagram

In this case, it needs users need 128(=16 x 8) of “Data Latch” to input the gray scale data; after the last “Data Latch”, 50 GCLKs is necessary to read the gray scale data into SRAM, and use the “Vsync” command to update the frame. The GCLK must be stopped before Vsync command, and there are some timing limitations which will be described detail in the following section.

Step 1. Data Input Sequence

First, input the gray scale data to six drivers with 8 scan lines, refer to fig.15. If only one LED driver is used, it needs 16 bits for each channel. Therefore, six drivers need 16x6 (=96) bits for each channel. The first 16 bits (bit95~bit80) is for the $\overline{OUT15}$ of 6th LED driver, and the last 16 bits (bit15~bit0) is for the $\overline{OUT15}$ of 1st LED driver. The MSB should be inputted first. User should follow this sequence to input the gray scale data for the remained $\overline{OUTn}$ (n=14~0). After these processes, it only finishes the gray scale data of 1st scan line. Other scan line’s gray scale data also have to repeat these processes.

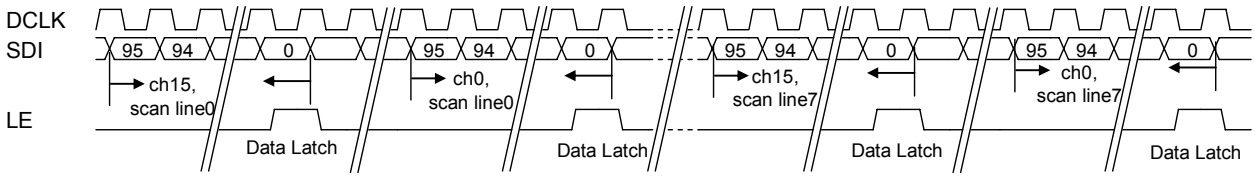

Fig.15 Gray Scale Data Input Sequence

Step 2. Vsync Command Operation to update the image frame (GCLK≠DCLK)

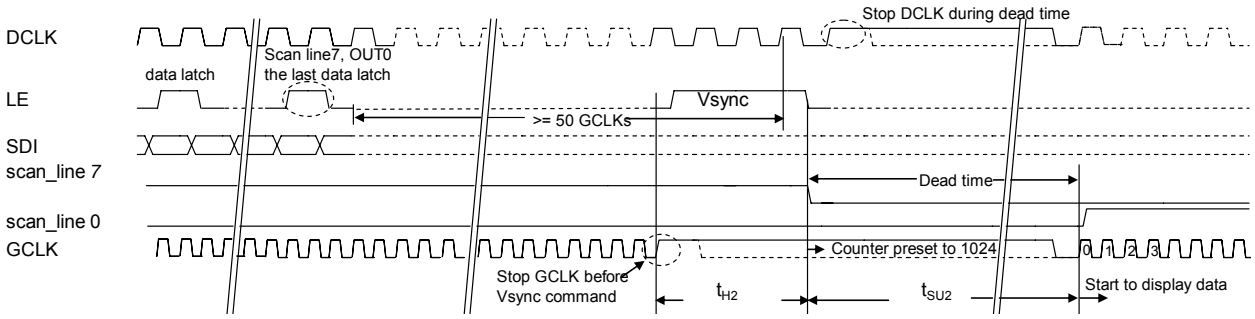

Fig.16 Vsync Command Operation (GCLK≠DCLK)

Since the gray scale data needs time to pre-read from SRAM to internal display buffer after last Data Latch command, there should be at least 50 GCLKs before the Vsync command is sent. After that, "Vsync" command is to update the frame. Refer to fig.16 and following shows the limitations:

a.) The GCLK should stop before Vsync command is sent. The hold time (between GCLK's rising edge and LE's falling edge) must meet the t_{H2}. The setup time (between LE's falling edge and GCLK's rising edge) must meet the t_{SU2}.

b.) The dead time is the time interval between scan lines, and is controlled by stopping GCLK. When Vsync command is set, the frame will be updated. The scan line needs to be switched (by controller) from 7 to 0, too.

c.) During dead time, please ensure all the output channels of driver ICs are turned off. Please don't send DCLK and any other commands, too.

d.) When no command inputs, the DCLK is don't care.

e.) The new data will be loaded to internal display buffer at Vsync command. But it will start to display after dead time is finished.

Vsync Command Operation to update the image frame (GCLK=DCLK)

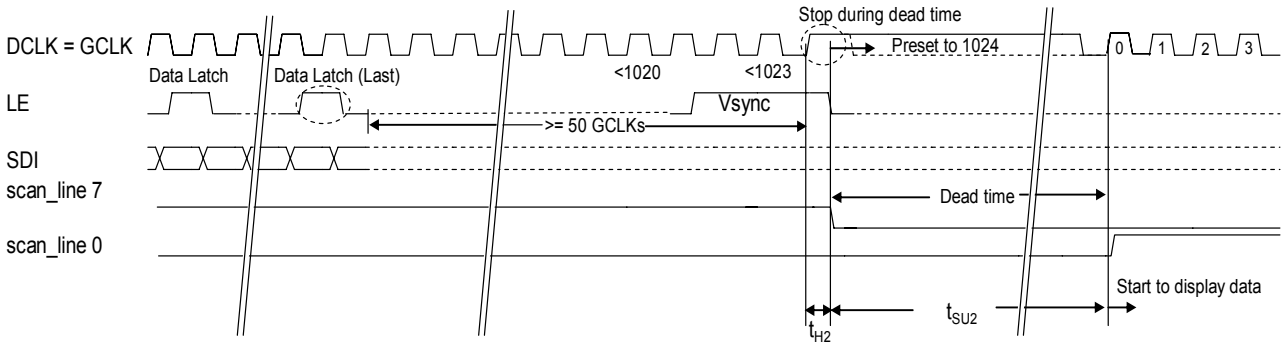

Fig.17 Vsync Command Operation (GCLK=DCLK)

If GCLK equals to DCLK, the control behavior is a little different from above:

a.) The DCLK cannot stop after command is delivered, due to DCLK also acts as GCLK.

b.) When the Vsync command is delivered, the DCLK has to stop for a dead time. Same as the limitation of GCLK=DCLK, the hold time and setup time must meet the t_{H2} and t_{SU2}, too.

c.) The stop point of DCLK for Vsync command can only be set before GCLK counter is < 1023.

Step 3. Scan Line Switching

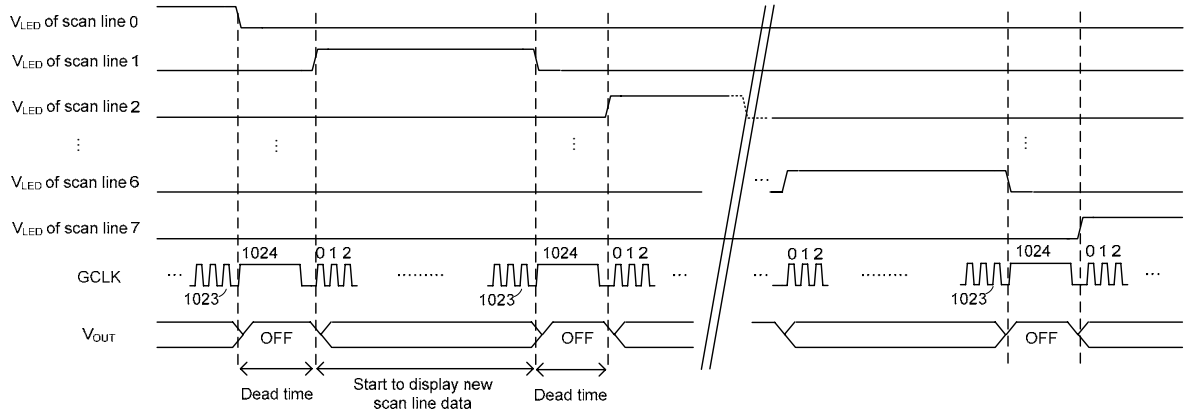

Fig.18 Time-multiplexing circuit diagram

MBI5050 uses the S-PWM technique to scramble the frame data from SRAM to increase the visual refresh rate. After GCLK counts from 0 to 1023, an extra GCLK is needed to switch the scan line; the time interval between the extra GCLK to next depends on the dead time of switched scan lines, refer to fig.18.

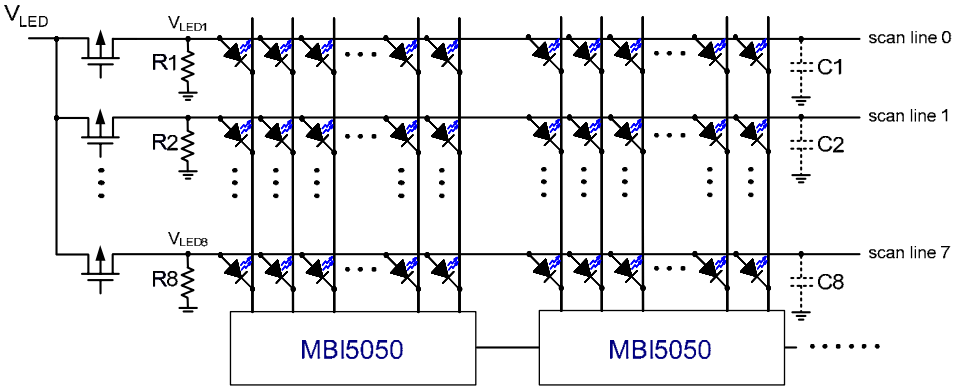

Fig.19 PCB layout Time-multiplexing circuit diagram

The dead time is to make sure supply voltage of previous LED row can completely discharge to 0V; otherwise, the residual charge stored in parasitic capacitor (C1 to C8) of previous LED row will cause the ghosting problem. The resisters, R1 to R8 in fig 19, are recommended to discharge the remaining VLED. Typically, the resistance is 5.1KΩ and it can be adjusted based on the actual electric circuit situation. When users design the PCB layout, please reserve the placements of these resistors to avoid ghosting problem.

During dead time, please ensure all the output channels of driver ICs are turned off. Please don't send DCLK and any other commands, too. It is also suggested for controller to keep one GCLK counter (from 0~1024), which will preset to 1024 at the falling edge of LE of Vsync command and restart from 0 at next GCLK.

Floor 6-4, No. 18, Pu-Ting Rd., Hsinchu, Taiwan 30077, ROC.

Other Control Command

Software Reset

"Software Reset" function includes

1. GCLK counter: GCLK counter will reset to zero.
2. Latch counter: Data latched counter and error status latched counter will reset to zero.
3. Enable/ disable all output: IC will return to normal status if it stays in the command of enable / disable all outputs.

"Software reset" function does not include

1. SRAM: The data in SRAM will keep the same, and it does not reset to zero.
2. Data in latched or in the register: The data, which includes error code in error status and code in configuration register, does not be changed to default value.
3. The parameter such as current gain code does not be changed.

Enable/Disable all outputs

Enable all Outputs: All channels will be turned on until receive the data latch command or software reset command. During the status, GCLK must keep running, and its frequency can not be lower than 20% of DCLK frequency. The "Enable all Outputs" command shows in fig 20. The command will stop when encountering data latch command.

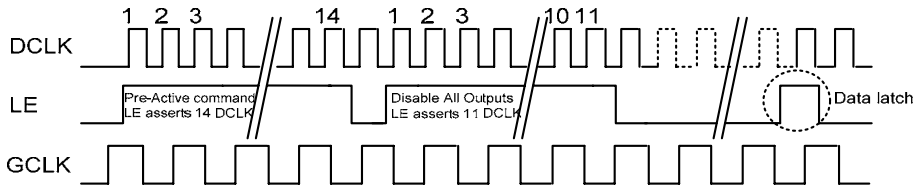

Fig.20 The waveform of "Enable all Outputs" command

Disable all Outputs: All channels will be turned off until receive the data latch command or software reset command. During the status, GCLK must keep running, and its frequency can not be lower than 20% of DCLK frequency. The "Disable all Outputs" command shows in fig 21. The command will stop when encountering data latch command.

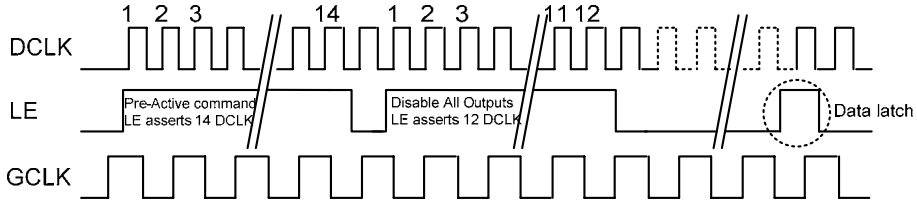

Fig.21 The waveform of "Disable all Outputs" command

Current Gain

It needs approximately several μsec to tens of μsec to change the current gain code in configuration register. The spent time depends on the range of varied output current. Though the current gain can be adjusted from 12.5% to 200%, the output current range of MBI5050 still have to limit in the range of 3mA to 45mA. Output current over design will make MBI5050 unstable moreover shorten the lifetime.

Conclusion

MBI5050 designs the embedded SRAM to support the time-multiplexing application. Once the frame data is sent in, it will be stored in the SRAM, and needn't to send in again when the scan line is changed. This application note provides the detailed instructions of MBI5050 for user. User can follow this article to make sure MBI5050 works correctly.