DLPC3439 Software Programmer’s Guide

User's Guide

Literature Number: DLPU035
August 2015

Contents

List of Figures

List of Tables

Introduction

1.1 Software Programmer's Guide Overview

This guide details the software interface requirements for a DLPC3439 DUAL ASIC based system. It defines all applicable communication protocols including I²C, initialization, default settings and timing. The DLPC3439 system can be used in accessory products with LED controller DLPA3000 or DLPA3005 in Figure 1-1 and Figure 1-2.

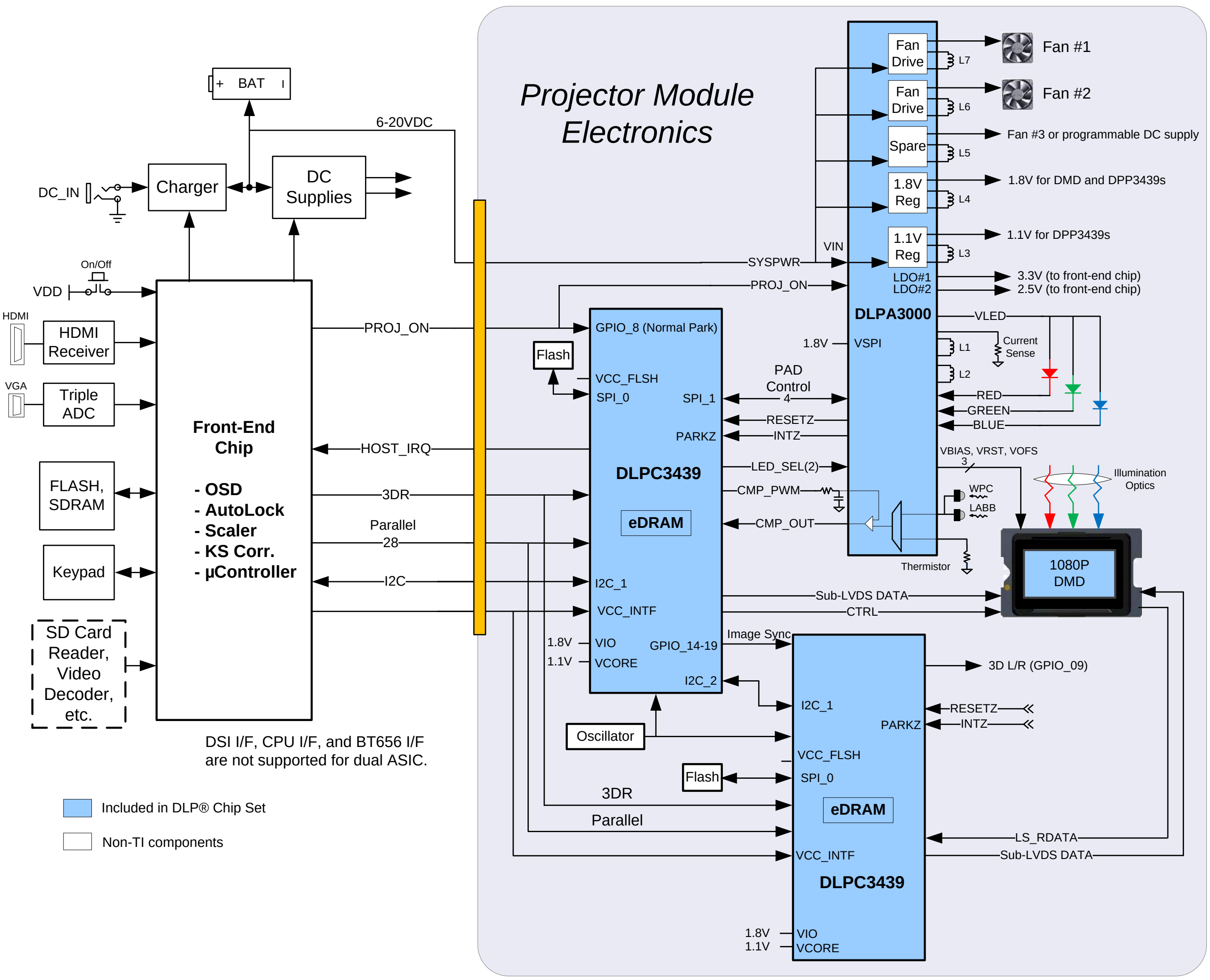

Figure 1-1. DLPC3439 Accessory Configuration with DLPA3000

Composer is a trademark of Texas Instruments Inc..

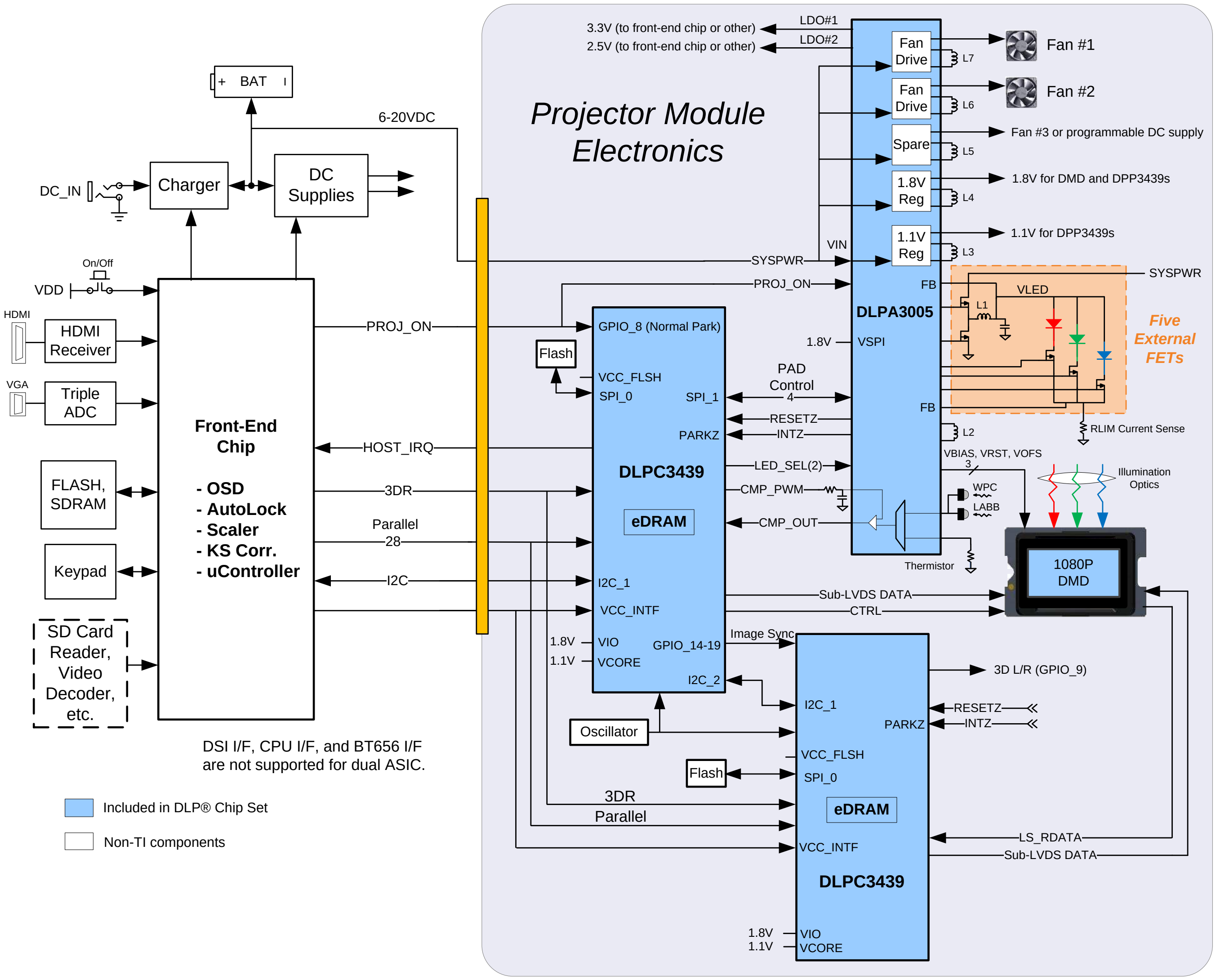

Figure 1-2. DLPC3439 Accessory Configuration with DLPA3005

1.1.1 I²C-Based Command Data Interface

The legacy interface configurations make use of an I²C interface for commands (conforming to the Philips I²C specification, up to 400 KHz) and a 24-bit parallel interface.

Note: Currently, we only support I²C speed of up to 100 kHz.

Interface Specification

2.1 Electrical Interface

This section discusses the requirements for a number of interface signals that are not command or data busses. These signals are used for different boot options.

2.1.1 System Power-up Associated Signals

2.1.1.1 EXT-BOOT-EN

The EXT-BOOT-EN signal is used by the ASIC hardware at system power-up to determine whether the internal boot application, or an external boot application (located in FLASH), is to be used during the ASIC initialization process. This is discussed further in Section 2.2.

2.1.1.2 DIS-PGM-LD

The DIS-PGM-LD signal is used by the boot application during system power-up to direct the function of the system boot application during the ASIC initialization process. This is discussed further in Section 2.2.

2.1.1.3 SPI-FLS-EN

The SPI-FLS-EN signal is used by the boot application during system power-up to direct the function of the system boot application during the ASIC initialization process. This is discussed further in Section 2.2.

2.1.1.4 High-Level Definition

As noted, a more detailed discussion of these signals is provided in Section 2.2; however, a brief summary is in Table 2-1.

Table 2-1. Summary of Settings for Power up Associated Signals(1)

EXT-BOOT-EN	DIS-PGM-LD	SPI-FLS-EN	Use/Definition
0	0	0	Normal Flash Operation
0	0	1	SPI Flashless Operation
0	1	0	Bad Flash Flashless Operation
0	1	1	N/A
1	0	0	TI Debug
1	0	1	N/A
1	1	0	TI Debug
1	1	1	N/A

(1) TI only supports Normal Flash Operation for DLPC3439.

2.2 System Initialization

This section discusses the methodologies used for system initialization.

2.2.1 Boot ROM Concept

In the DLPC3439, a boot ROM, with associated boot software, will be employed. This *resident* boot code will consist of the minimum code needed to complete the various tasks required based on the state of the DIS-PGM-LD (Disable Program Load) pin and the SPI-FLS-EN (SPI Flashless Enable) pin.

2.2.2 Internal vs External Boot Software

In the DLPC3439, the state of the EXT-BOOT-EN (External Boot Enable) pin allows the external user to specify whether the hardware points the microprocessor to the internal boot ROM for the boot application (EXT-BOOT-EN = *0*), or points it to an external FLASH for the boot application (EXT-BOOT-EN = *1*). Allowing for the use of an external boot program in FLASH is to provide for debug and boot code development purposes only (since uP code execution out of serial flash will be extremely slow).

2.2.3 Flash and Flashless Product Configurations

For most DLPC3439 product configurations, an external FLASH device will be used to store the main application code, along with all of the other configuration and operational data required by the system for normal operation.

In certain applications it may be desirable to eliminate this external FLASH part (for cost reasons). In these *Flashless* configurations, the expectation is that the main application code will be downloaded (by command) to iRAM by the *Boot Application* via the SPI port. All other configuration and operational data normally obtained from Flash will be obtained by the *Main Application* code via the SPI port.

For all discussions in this document, unless stated otherwise, it is assumed that an external FLASH device will be used.

2.2.4 Resident Boot Software (EXT-BOOT-EN = 0)

As noted previously, an internal boot ROM, with associated boot software, will be employed. This resident boot code will consist of the minimum code needed to complete the various tasks required based on the state of the DIS-PGM_LD and SPI-FLS_EN pins (with EXT-BOOT-EN = ‘0’). An overview of these tasks is shown in Figure 2-1.

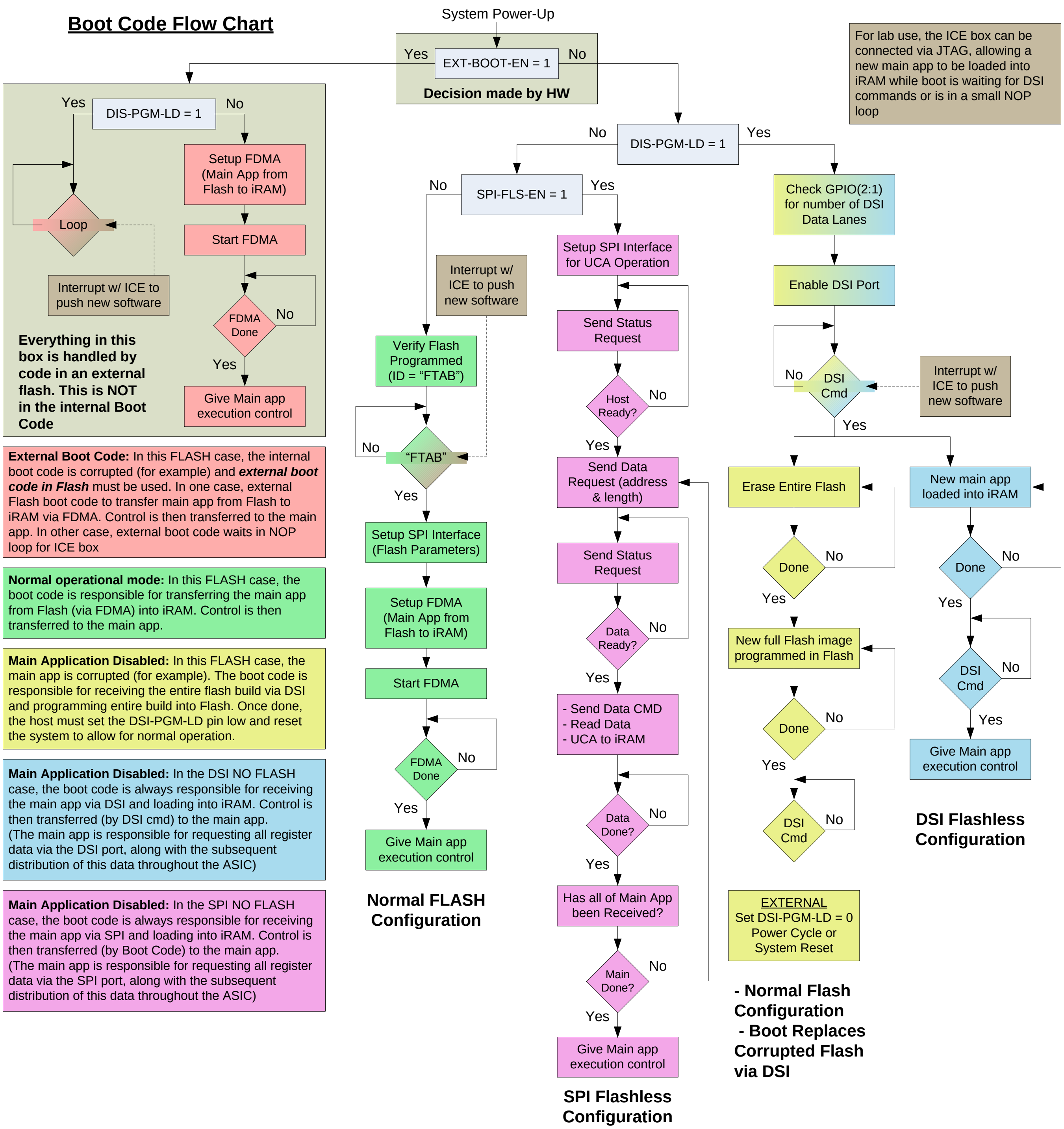

Figure 2-1. Boot Code Flow Chart

2.2.4.1 DIS-PGM-LD = 0 (With EXT-BOOT-EN = 0)

2.2.4.1.1 SPI-FLS-EN = 0

This should be the normal operational mode of the boot application for a Flash based product configuration during normal ASIC use.

2.2.4.1.2 SPI-FLS-EN = 1

This should be the normal operational mode of the boot application for an SPI Flashless based product configuration during normal ASIC use. In this case, the boot application will expect to get the main application from the host via the SPI port in response to TI command requests. The only SPI interface instructions that will be supported by the boot code are associated with requesting and read data from the host via this port.

2.3 Software Interface

In general, there will be one set of software commands supported by the DLPC3439 DUAL ASIC. This custom set of TI specific commands will be applicable for use on I^2C command interface.

2.3.1 Software Command Philosophy

With DLPC3439, *all* commands via I^2C will be processed by software. As such, no commands will directly address or access ASIC registers, ASIC mailboxes, or any attached flash parts. All commands will be of a high level, more abstract nature, decoupling the OEM from the internal hardware of the ASIC.

2.3.2 I^2C Considerations

2.3.2.1 I^2C Transactions

Since all I^2C commands will be processed by software, there is just one type of I^2C transaction to be supported. This transaction type is shown in Table 2-2 for both writes and reads. It should be noted that the I^2C interface is able to support variably sized transactions (that is, a one byte transaction, a nine byte transaction) to match the TI commands discussed later in this document.

Table 2-2. I^2C Write and Read Transactions

Transaction	Address(1)	Sub-Address(2)	Remaining Data Bytes(3)
Write	8-bits	8-bits	8-bit parameter bytes (0 -> N)
	36h (or 3Ah)	Command Value	Parameter Values
Read Request	8-bits	8-bits	8-bit parameter bytes (0 -> N)
	36h (or 3Ah)	Command Value	Parameter Values
Read Response	8-bits		8-bit parameter bytes (0 -> N)
	37h (or 3Bh)		Parameter Values

(1) The address corresponds to the chip address of the ASIC.
(2) The sub-address will correspond to a TI command.
(3) The data (if present) will correspond to any required command parameters.

2.3.2.2 Data Flow Control

While the I^2C interface inherently supports flow control by holding the clock, this will likely not be sufficient for all transactions (sequence and CMT updates for example). In this case, the host software should make use of the Read Short Status to determine if the system is busy.

2.3.3 List of System Write/Read Software Commands

The commands supported by the IC interfaces are discussed in the following sections.

2.3.3.1 List of TI-Specific Generic Commands

Table 2-3. Supported TI Generic Commands

Command Type	Command Description	Reset Value	OpCode (hex)	Default Action	Page #
General Operation					
Write	Write Input Source Select	1	05	Test Pattern	21
Read	Read Input Source Select		06		25
Write	Write External Video Source Format Select	43h	07	RGB888	26
Read	Read External Video Source Format Select		08		27
Write	Write External Video Chroma Processing Select	0	09	Chroma Interpolation	28
Read	Read External Video Chroma Processing Select		0A		30
Write	Write Test Pattern Select	7000h	0B	White Solid Field	30
Read	Read Test Pattern Select		0C		43
Write	Write Splash Screen Select		0D	Composer Specified	44
Read	Read Splash Screen Select		0E		46
Read	Read Splash Screen Header		0F		47
Write	Write Image Crop	ffffffff00000000h	10	No Crop	49
Read	Read Image Crop		11		50
Write	Write Display Size	DMD Res	12		51
Read	Read Display Size		13		53
Write	Write Display Image Orientation		14	Composer Specified	54
Read	Read Display Image Orientation		15		56
Write	Write Display Image Curtain	1	16	Black	57
Read	Read Display Image Curtain		17		58
	Unused		*18-19*		
Write	Write Image Freeze	0	1A	No Freeze	59
Read	Read Image Freeze		1B		62
Write	Write 3-D Control	0	20	Automatic	63
Read	Read 3-D Control		21		65
Write	Write LOOK Select		22	Composer Specified	66
Read	Read LOOK Select		23		67
Read	Read Sequence Header Attributes		26		68
Write	Write Degamma/CMT Select		27	Composer Specified	70
Read	Read Degamma/CMT Select		28		71
Write	Write CCA Select		29	Composer Specified	72
Read	Read CCA Select		2A		73
Write	Write Execute Batch File	0	2D		74
Write	Write External Input Image Size	DMD Res	2E		76
Read	Read External Input Image Size		2F		78
Write	Write 3-D Reference	0	30	Next Frame Left	79
Write	Write GPIO[19:00] Control		31	Composer Specified	80
Read	Read GPIO[19:00] Control		32		83
Write	Write GPIO[19:00] Outputs		33	Composer Specified	86
Read	Read GPIO[19:00] Outputs		34		90

Table 2-3. Supported TI Generic Commands (continued)

Command Type	Command Description	Reset Value	OpCode (hex)	Default Action	Page #
Write	Write Splash Screen Execute		35		92
Read	Read GPIO[19:00] Inputs		36		93
Write	Write External Parallel I/F Data Mask Control	0	37		94
Read	Read External Parallel I/F Data Mask Control		38		96
	Unused		*39-4F*		
Illumination Control					
Write	Write LED Output Control Method		50	Composer Specified	97
Read	Read LED Output Control Method		51		99
Write	Write RGB LED Enable	7h	52	Enabled	100
Read	Read RGB LED Enable		53		101
Write	Write RGB LED Current		54	Composer Specified	102
Read	Read RGB LED Current		55		104
Read	Read CAIC LED Max Available Power		57		105
Write	Write RGB LED Max Current		5C	Composer Specified	106
Read	Read RGB LED Max Current		5D		107
Read	Read Measured LED Parameters		5E		107
Read	Read CAIC RGB LED Current		5F		109
Image Processing Control					
Write	Write Local Area Brightness Boost Control	1	80	Manual Strength Control	110
Read	Read Local Area Brightness Boost Control		81		111
Write	Write CAIC Image Processing Control		84	Composer Specified	112
Read	Read CAIC Image Processing Control		85		115
Write	Write CCA Control	1	86	Enabled	116
Read	Read CCA Control		87		117
Write	Write Border Color	0	B2	Black	118
Read	Read Border Color		B3		120
Write	Write External Parallel I/F SYNC Polarity	0	B6	0	121
Read	Read External Parallel I/F SYNC Polarity		B7		122
Write	Write External Parallel I/F Manual Image Framing	0	B8	Disabled	123
Read	Read External Parallel I/F Manual Image Framing		B9		124
Read	Read Auto Framing Information		BA		125
Administrative Commands					
Read	Read Short Status		D0		126
Read	Read System Status		D1		128
Read	Read System Software Version		D2		133
Read	Read Communication Status		D3		134
Read	Read ASIC Device ID		D4		137
Read	Read DMD Device ID		D5		138
Read	Read System Temperature		D6		139
Read	Read Flash Build Version		D9		140
Write	Write Batch File Delay		DB	Composer Specified	141
Read	*Read DMD I/F Training Data*		*DC*		141
Read	Flash Update PreCheck		DD		145
Write	Flash Data Type Select	0	DE	Entire Flash	147
Write	Flash Data Length	0	DF		151

Table 2-3. Supported TI Generic Commands (continued)

Command Type	Command Description	Reset Value	OpCode (hex)	Default Action	Page #
Write	Erase Flash Data		E0		152
Write	Write Flash Start		E1		153
Write	Write Flash Continue		E2		153
Read	Read Flash Start		E3		154
Read	Read Flash Continue		E4		156
Write	Write Internal Register Address	0	E5		157
Write	Write Internal Register		E6		158
Read	Read Internal Register		E7		159
Write	Write Internal Mailbox Address	0	E8		160
Write	Write Internal Mailbox		E9		163
Read	Read Internal Mailbox		EA		164
Write	Write External PAD Address	0	EB		165
Write	Write External PAD Data		EC		167
Read	Read External PAD Data		ED		168
	Reserved		*F8-FF*		

2.3.3.2 Write Input Source Select (05h)

2.3.3.2.1 Write

This command is used to select the image input source for the display module.

2.3.3.2.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1:0) – Input Source

0h: External Video Port
1h: Test Pattern Generator
2h: Splash Screen
3h: *Reserved*

Default: 01h

Note 1: When selecting the External Video Port, there is a set of associated commands that are only applicable to this source selection. These associated commands are the *Write External Input Image Size* (Section 2.3.3.33), the *Write External Video Source Format Select* (Section 2.3.3.4), the *Write External Video Chroma Processing Select* (Section 2.3.3.6), the *Write External Parallel I/F Manual Image Framing* (Section 2.3.3.42), and *Write External CPU I/F Video Sync Method* commands.

When selecting the Test Pattern Generator, there is one associated command that is only applicable to this source selection. This associated command is the *Write Test Pattern Select* (Section 2.3.3.8) command.

When selecting the Splash Screen, there are two associated commands that are only applicable to this source selection. These associated commands are the *Write Splash Screen Select* (Section 2.3.3.10) and *Write Splash Screen Execute* (Section 2.3.3.40) commands.

These associations are also shown in Table 2-4.

Table 2-4. Source-Specific Associated Commands

	Input Source Select Options		
Source Specific Associated Commands	**External Video Port**	**Test Pattern Generator**	**Splash Screen** [(1)]
Write External Video Source Format Select	Only	N/A	N/A
Write External Video Chroma Processing Select	Only	N/A	N/A
Write External Input Image Size	Only	N/A	N/A
Write External Parallel I/F Manual Image Framing	Only	N/A	N/A
Write Test Pattern Select	N/A	Only	N/A
Write Splash Screen Select	N/A	N/A	Only
Write Splash Screen Execute	N/A	N/A	Special

(1) The Write Splash Screen Execute command is special in that there is no maintained state or history. Thus, this command has no *settings* to be stored and reused by the system.

These commands (except for *Write Splash Screen Execute*) describe the unique characteristics of their associated source, and once these settings have been defined, the system will store them. Then, each time an input source selection is made (using the *Write Input Source Select* command), the system will *remember* the settings described by the commands associated with the selected source and automatically apply them. As such, the user only needs to send these associated commands when the source first needs to be defined, or when the source characteristics for that port need to be changed. It is important to note that the appropriate associated commands must be updated when source characteristics do change.

If the user wants to send source associated commands each time an input source selection is made, this is allowed. In this case, it is recommended that the source associated commands be sent prior to sending the *Write Input Source Select* command. When source associated commands are sent when that source is not active, the ASIC software will save the new settings, but will not execute these commands. When that source becomes active (via the *Write Input Source Select* command), the ASIC will apply these new settings. An example is shown below:

1. User sends the following commands (active Input Source = Test Pattern Generator)
   (a) Write Image Freeze = Freeze
   (b) Write External Video Source Format Select (settings stored, command not executed)
   (c) Write External Video Chroma Processing Select (settings stored, command not executed)
   (d) Write External Input Image Size (settings stored, command not executed)
   (e) Write Input Source Select = External Port (See *b*, below)
   (f) Write Image Freeze = Unfreeze
2. When the Write Input Source Select command is received, software will apply the settings from these External Video Port associated commands.
   (a) External Video Source Format Select
   (b) External Video Chroma Processing Select
   (c) External Input Image Size
   (d) External Parallel Manual Image Framing (as appropriate – that is, if parallel port selected)
   (e) External CPU Video Sync Mode (as appropriate – that is, if CPU port selected)

If source associated commands are sent for a source that is already active, the ASIC software will execute these commands when received. An example is shown below:

1. User sends the following commands (active Input Source = External Video Port)
   (a) Write Image Freeze = Freeze
   (b) Write External Video Source Format Select (command executed)
   (c) Write External Video Chroma Processing Select (command executed)
   (d) Write Image Freeze = Unfreeze

Note 2: The rest of the commands that apply to image setup are those commands whose settings are applicable across all source selections, and indeed, these command settings would typically remain the same across the three Input Source selections. A few examples are Write Display Size and Write Display Image Orientation. A more representative list of these commands is shown in Table 2-5.

Table 2-5. Common Commands

	Input Source Select Options		
Common Commands	**External Video Port**	**Test Pattern Generator**	**Splash Screen**
Write Image Crop	Common	Common	Common
Write Display Image Size	Common	Common	Common
Write Display Image Orientation	Common	Common	Common
Write Display Image Curtain	Common	Common	Common
Write Look Select	Common	Common	Common

Table 2-5. Common Commands (continued)

	Input Source Select Options		
Common Commands	**External Video Port**	**Test Pattern Generator**	**Splash Screen**
Write Sequence Select	Common	Common	Common
Write Local Area Brightness Boost Control	Common	Common	Common
Write CAIC Image Processing Control	Common	Common	Common

It is important to note that while the values for these commands may be the same across the different input source types, this does not mean that hardware settings will not change (Here is one example: Display Image Size = 1080p = DMD size – The external port input source size is WXGA which is scaled up to the display size of 1080p. If the user changes to the TPG Input Source, our rule is that the size of the test pattern is to match the size of the DMD. Therefore, the scaler settings would have to be changed). The ASIC software is responsible for managing the underlying hardware settings. This also applies to those commands which specify *automatic* operation (for example, Write Idle Mode Select = Auto Idle Mode Enable). While the setting of automatic would remain the same, the actual underlying algorithm might change its settings based on the characteristic of the selected source.

Note 3: The user is required to specify the active data size for all external input sources using the *Write Input Image Size* (Section 2.3.3.33) command. In addition, for input image data on the Parallel bus that does not provide data framing information, the user is required to provide manual framing data using the *Parallel I/F Manual Image Framing* command (Section 2.3.3.42).

Note 4: When a test pattern is selected, it will be generated at the resolution of the DMD, modified by the settings specified by the *Write Image Crop* command (Section 2.3.3.13), and displayed at the resolution specified by the *Write Display Size* command (Section 2.3.3.15).

Note 5: The user should see the *Write Image Freeze* command (Section 2.3.3.21) for information on hiding on-screen artifacts when selecting an input source

2.3.3.3 Read Input Source Select (06h)

2.3.3.3.1 Read

This command is used to read the state of the image input source for the display module.

2.3.3.3.2 Read Parameters

This command has no command parameters.

2.3.3.3.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1:0) – Input Source
0h: External Video Port
1h: Test Pattern Generator
2h: Splash Screen
3h: *Reserved*

2.3.3.4 Write External Video Source Format Select (07h)

2.3.3.4.1 Write

This command is used to specify the active external video port and the source data type for the display module.

2.3.3.4.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See CMD Parameter Below

CMD Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Note (3)
Parallel Port User Selection						
40h	Parallel	16	RGB565	16	1	Auto-select RGB CSC
41h	Parallel	18	RGB 666	18	1	Auto-select RGB CSC
42h	Parallel	24	RGB 888	8	3	Auto-select RGB CSC
43h	Parallel	24	RGB 888	24	1	Auto-select RGB CSC
50h	Parallel	18	YCbCr 666	18	1	Auto-select YCbCr CSC
51h	Parallel	24	YCbCr 888	24	1	Auto-select YCbCr CSC
60h	Parallel	16	YCbCr 4:2:2 88	8	2	Auto-select YCbCr CSC Auto-select 4:2:2 -> 4:4:4
61h	Parallel	16	YCbCr 4:2:2 88	16	1	Auto-select YCbCr CSC Auto-select 4:2:2 -> 4:4:4

Default: 43h

Note 1: This command is used in conjunction with the *Write Input Source Select* command (Section 2.3.3.2). This command specifies which input port is to be displayed when the *Write Input Source Select* command selects External Video Port as the image source. The settings for this command will be retained until changed using this command. These settings will be automatically applied each time the External Video Port is selected.

Note 2: When the external video port is selected as the input source, software will automatically select and load the proper CSC based on the selected parameter of this command (appropriate matrix for RGB, selected matrix for YCbCr including offset). It will also automatically select the appropriate data path for 4:2:2 vs. 4:4:4 processing. It should be noted that the OEM is responsible for ensuring the appropriate source Chroma parameters are set using the *Write External Video Chroma Processing Select* command (Section 2.3.3.6).

Note 3: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note that this command is a source associated command.*

2.3.3.5 Read External Video Source Format Select (08h)

2.3.3.5.1 Read

This command is used to read the state of the active external video port and the source data type for the display module.

2.3.3.5.2 Read Parameters

This command has no command parameters.

2.3.3.5.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See CMD Parameter Below

CMD Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Notes (3)
Parallel Port User Selection						
40h	Parallel	16	RGB565	16	1	Auto-select RGB CSC
41h	Parallel	18	RGB 666	18	1	Auto-select RGB CSC
42h	Parallel	24	RGB 888	8	3	Auto-select RGB CSC
43h	Parallel	24	RGB 888	24	1	Auto-select RGB CSC
50h	Parallel	18	YCbCr 666	18	1	Auto-select YCbCr CSC
51h	Parallel	24	YCbCr 888	24	1	Auto-select YCbCr CSC
60h	Parallel	16	YCbCr 4:2:2 88	8	2	Auto-select YCbCr CSC Auto-select 4:2:2 -> 4:4:4
61h	Parallel	16	YCbCr 4:2:2 88	16	1	Auto-select YCbCr CSC Auto-select 4:2:2 -> 4:4:4

2.3.3.6 Write External Video Chroma Processing Select (09h)

2.3.3.6.1 Write

This command is used to specify the characteristics of the selected YCbCr source, as well as specifying the type of chroma processing to be used for this YCbCr source by the display module.

2.3.3.6.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below
Byte 2	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:5) – *Reserved*

b(4) – Chroma Interpolation Method
0h: Chroma Interpolation
1h: Chroma Copy
2h: Splash Screen
3h: *Reserved*

b(3) – *Reserved*

b(2) – Chroma Channel Swap
0h: CbCr
1h: CrCb

b(1:0) – *Reserved*

Byte 1 Default: 00h

msb	***Byte 2***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*
b(1:0) – CSC Coefficient Set (Color Space)

Byte 2 Default: 00h

Note 1: This command is used in conjunction with the *Write Input Source Select* command (Section 2.3.3.2). The settings for this command will be retained until changed using this command. These settings will be automatically applied each time the External Video Port is selected.

Note 2: The system will assume RGB sources have a dynamic range of 0 to 255, with an offset of 0.

Note 3: Bits 3:0 for Byte 1 are used to specify the characteristics for the current YCbCr source. Bits 7:4 for Byte 1, as well as Byte 2, are used to specify the type of processing to be done on the current YCbCr source.

Note 4: CSC coefficient sets are stored in Flash until needed.

Note 5: CSC coefficient sets are specified in Byte 2 by an enumerated value (such as 0, 1, 2, or 3). The set stored in ‘0' is ITU-R BT. Rec. 601. The other three sets are customer definable via Composer™.

It should be noted that regardless of the setting for this command, set *0* will always be used for the conversion of Splash Screen images stored as YCbCr (since this is the CSC that is used by Composer to convert from RGB to YCbCr). This will be done internally by TI software, and the setting of this command will not be changed.

Note 6: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note that this command is a source associated command.*

Note 7: Luma Offset and YCbCr Dynamic Range are to be specified at the beginning of the CSC coefficient set by Composer.

2.3.3.7 Read External Video Chroma Processing Select (0Ah)

2.3.3.7.1 Read

This command is used to read the specified characteristics for the selected YCbCr source, as well as the type of chroma processing being used for this YCbCr source by the display module.

2.3.3.7.2 Read Parameters

This command has no command parameters.

2.3.3.7.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below
Byte 2	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:5) – *Reserved*

b(4) – Chroma Interpolation Method
0h: Chroma Interpolation
1h: Chroma Copy

b(3) – *Reserved*

b(2) – Chroma Channel Swap
0h: CbCr
1h: CrCb

b(1:0) – *Reserved*

msb	*Byte 2*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – CSC Coefficient Set

2.3.3.8 Write Test Pattern Select (0Bh)

2.3.3.8.1 Write

This command is used to specify an internal test pattern for display on the display module.

2.3.3.8.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*	
Byte 1	TPG Pattern Select	(See Below)
Byte 2	Foreground / Background Color	(See Below and Table 2-6)
Byte 3	Parameter 1	(See Table 2-7)
Byte 4	Parameter 2	(See Table 2-7)
Byte 5	Parameter 3	(See Table 2-7)
Byte 6	Parameter 4	(See Table 2-7)

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – Test Pattern Border
00h: Disabled
01h: Enabled

b(6:4) – *Reserved*

b(3:0) – Left Pattern Select
00h: Solid Field
01h: Fixed Step Horizontal Ramp
02h: Fixed Step Vertical Ramp
03h: Horizontal Lines
04h: Diagonal Lines
05h: Vertical Lines
06h: Horizontal and Vertical Grid
07h: Checkerboard
08h: Color Bars
09h-0Fh: Reserved

Byte 1 Default: 00h

msb	*Byte 2*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – *Reserved*

b(6:4) – Foreground Color
0h: Black
1h: Red
2h: Green
3h: Blue

4h: Cyan
5h: Magenta
6h: Yellow
7h: White

b(3) – *Reserved*

b(2:0) – Background Color
0h: Black
1h: Red
2h: Green
3h: Blue
4h: Cyan
5h: Magenta
6h: Yellow
7h: White

Table 2-6. Foreground and Background Color Use

Pattern	Byte 2	
	Foreground Color	Background Color
Solid Field	Yes	No
Fixed Step Horizontal Ramp	Yes	No
Fixed Step Vertical Ramp	Yes	No
Horizontal Lines	Yes	Yes
Vertical Lines	Yes	Yes
Diagonal Lines	Yes	Yes
Grid Lines	Yes	Yes
Checkerboard	Yes	Yes
Color Bars	No	No

Byte 2 Default: 70h

Table 2-7. Descriptions and Bit Assignments for Parameters 1-4

	Byte 6 (Parameter 4)		Byte 5 (Parameter 3)		Byte 4 (Parameter 2)		Byte 3 (Parameter 1)	
Pattern	**Description**	**Bits**	**Description**	**Bits**	**Description**	**Bits**	**Description**	**Bits**
Solid Field	n/a		n/a		n/a		n/a	
Fixed Step Horizontal Ramp	n/a		n/a		End Value	8	Start Value	8
Fixed Step Vertical Ramp	n/a		n/a		End Value	8	Start Value	8
Horizontal Lines	n/a		n/a		Background Line Width	8	Foreground Line Width	8
Vertical Lines	n/a		n/a		Background Line Width	8	Foreground Line Width	8
Diagonal Lines	n/a		n/a		Vertical Spacing	8	Horizontal Spacing	8
Grid Lines	Vertical Background Line Width	8	Vertical Foreground Line Width	8	Horizontal Background Line Width	8	Horizontal Foreground Line Width	8
Checkerboard	Number of Vertical Checkers	3	Number of Vertical Checkers	8	Number of Horizontal Checkers	3	Number of Horizontal Checkers	8
Color Bars	n/a		n/a		n/a		n/a	

Note 1: This command is used in conjunction with the *Write Input Source Select* command (Section 2.3.3.2). This command specifies which test pattern is to be displayed when the *Write Input Source Select* command selects Test Pattern Generator as the image source. The settings for this command are to be retained until changed using this command. These settings will be automatically applied each time the Test Pattern Generator is selected.

Note 2: Write Execute Batch files (Section 2.3.3.32) can be created and stored in Flash and used to recall the settings for predefined test patterns.

Note 3: Test Patterns will be created at the resolution of the display (DMD), however, they can be modified by the *Write Image Crop* command (Section 2.3.3.13), and will be displayed at the resolution specified by the *Write Display Size* command (Section 2.3.3.15).

Note 4: Test Patterns will be displayed at the frame rate of 60 Hz.

Note 5: The Test Pattern border selection creates a single pixel wide/tall white border around the specified test pattern.

Note 6: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note that this command is a source associated command.*

Note 7: When a Foreground or Background Color is not used, the bit values will be ignored (See Table 2-6). If both Foreground and Background Color are not used, or when a Parameter Byte (Bytes 3 thru 6) is not used, the byte should not be sent. This is shown in Table 2-8, which shows the number of bytes required based on the specified pattern.

Table 2-8. Number of Bytes Required based on Pattern Selection

Specified Pattern	Solid Field	Fixed Step Horz Ramp	Fixed Step Vert Ramp	Horz Lines	Vert Lines	Diag Lines	Grid Lines	Checker board	Color Bars
Number of Bytes Required	2	4	4	4	4	4	6	6	1

Note 8: As noted in Table 2-6 the color for the Solid Field pattern is specified using the Foreground color. An example of a Solid Field pattern is shown in Table 2-8.

Figure 2-2. Example of Solid Field Test Pattern (Red)

Note 9: As noted in Table 2-6, the color for the Fixed Step Horizontal Ramp pattern is specified using the Foreground color. As noted in Table 2-7, the user specifies the start value and the stop value for the ramp. For this pattern, the system will automatically determine the step size based on the start/stop values and the size of the display (DMD). The minimum start value = 0, the maximum stop value = 255, and the start value must always be smaller than the stop value. As an example, if the start value = 0, the stop value = 255, and the DMD resolution is 1280 wide – the step size would be 5 (1280 pixels / 256 values = 5). Thus every gray shade value from 0 to 255 would have a step size of 5 pixels (that is, each step would have 5 columns of pixels with the same gray scale value). The gray scale value always increments by 1 for each step between the start and stop values. An example of a Fixed Step Horizontal Ramp pattern is shown in Table 2-6.

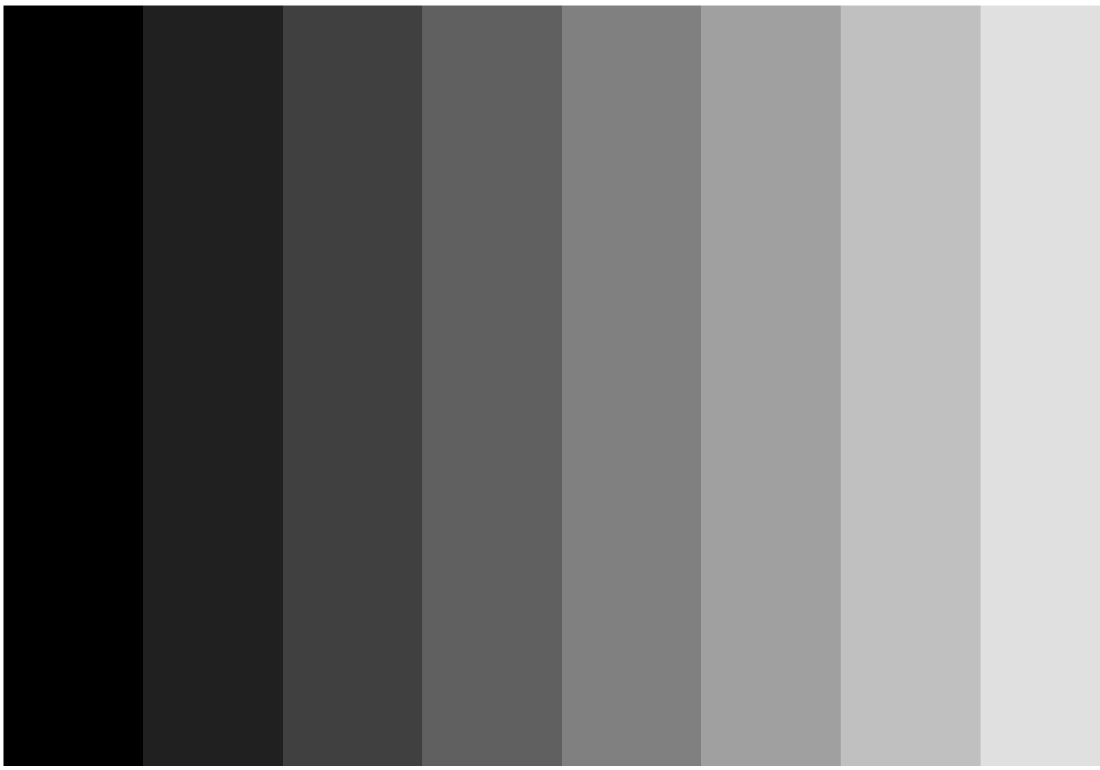

Figure 2-3. Example of Fixed Step Horizontal Ramp Test Pattern

Note 10: As noted in Table 2-6, the color for the Fixed Step Vertical Ramp pattern is specified using the Foreground color. As noted in Table 2-7, the user specifies the start value and the stop value for the ramp. For this pattern, the system will automatically determine the step size based on the start/stop values and the size of the display (DMD). The minimum start value = 0, the maximum stop value = 255, and the start value must always be smaller than the stop value. As an example, if the start value = 0, the stop value = 255, and the DMD resolution is 768 tall – the step size would be 3 (768 pixels / 256 values = 3). Thus every value from 0 to 255 would have a step size of 3 pixels (that is, each step would have 3 rows of pixels with the same gray scale value). The gray scale value always increments by 1 for each step between the start and stop values. An example of a Fixed Step Vertical Ramp pattern is shown in Figure 2-4.

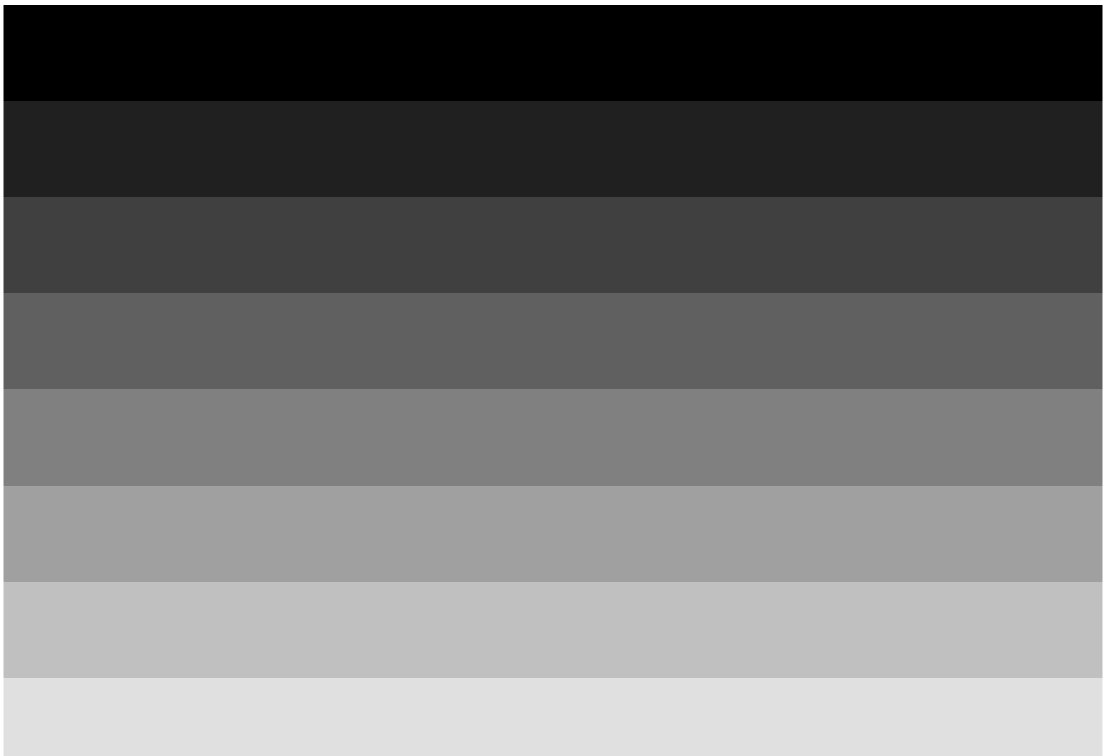

Figure 2-4. Example of Fixed Step Vertical Ramp Test Pattern

Note 11: As noted in Table 2-6, the colors for the Horizontal Lines pattern are specified using both the Foreground and Background colors. The foreground color is used for the horizontal lines, and the background color is used for the space between the lines. As noted in Table 2-7, the user specifies the Foreground Line Width, as well as the Background Line Width. It is up to the user to determine the line spacing that will meet their needs for each resolution display. As an example, if the foreground line width = 1, and the background line width = 9, there would be a single pixel horizontal line on every 10th line. An example of a Horizontal Lines pattern is shown in Figure 2-5.

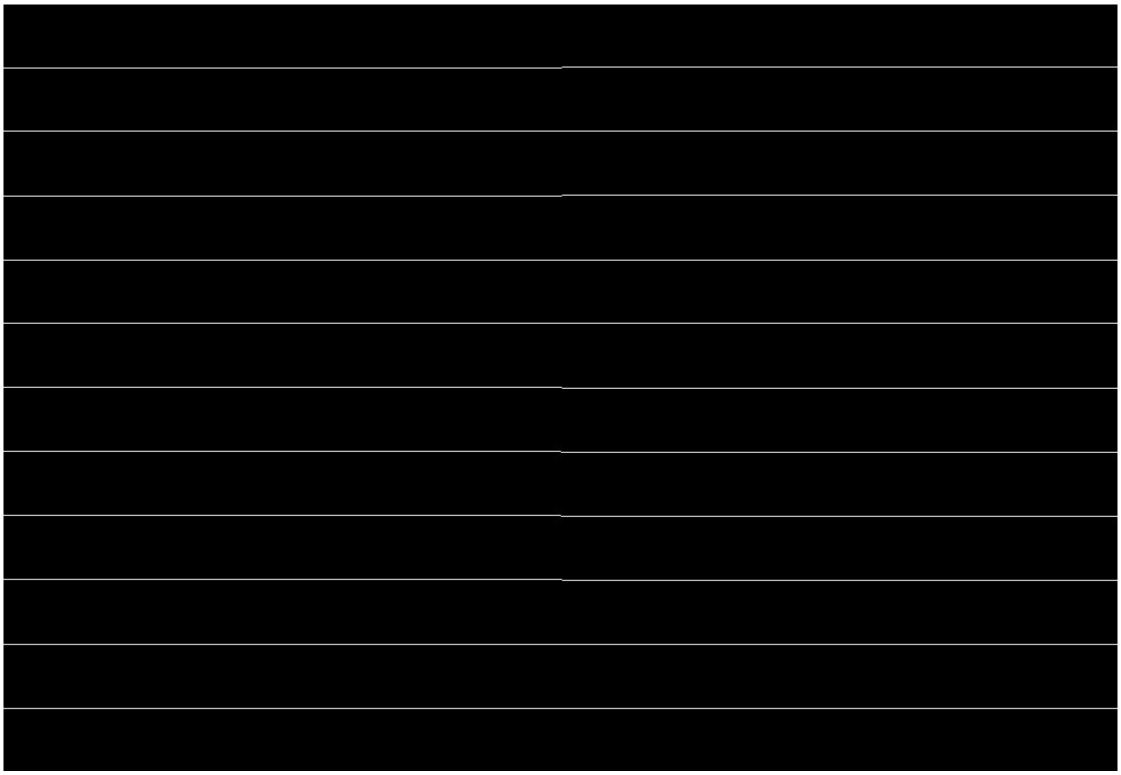

Figure 2-5. Example of Horizontal Lines Test Pattern

Note 12: As noted in Table 2-6, the colors for the Vertical Lines pattern are specified using both the Foreground and Background colors. The foreground color is used for the vertical lines, and the background color is used for the space between the lines. As noted in Table 2-7, the user specifies the Foreground Line Width, as well as the Background Line Width. It is up to the user to determine the line spacing that will meet their needs for each resolution display. As an example, if the foreground line width = 1, and the background line width = 9, there would be a single pixel vertical line on every 10 th line. An example of a Vertical Lines pattern is shown in Figure 2-6.

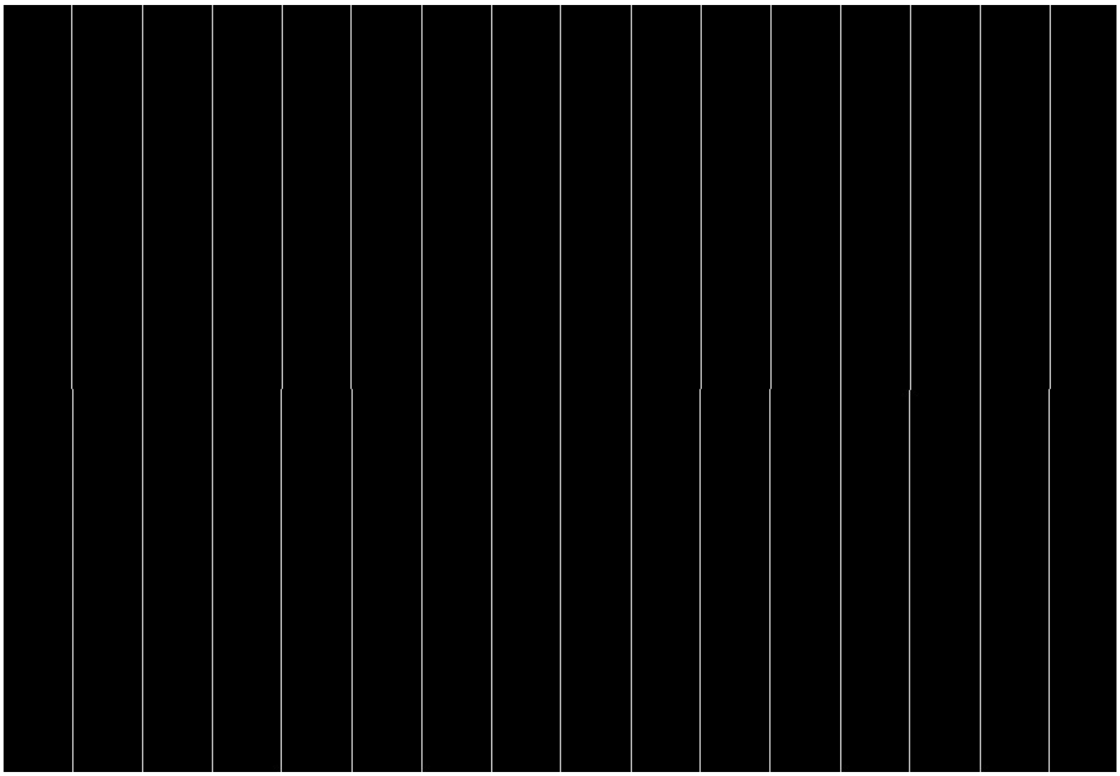

Figure 2-6. Example of Vertical Lines Test Pattern

Note 13: As noted in Table 2-6, the colors for the Diagonal Lines pattern are specified using both the Foreground and Background colors. The foreground color is used for the diagonal lines, and the background color is used for the space between the lines. As noted in Table 2-7, the user specifies the Horizontal and Vertical Line Spacing. The line width will always be one pixel. It is up to the user to determine the line spacing that will meet their needs for each resolution display. It should be noted that both horizontal and vertical line spacing must use the same value, and are limited to values of 3, 7, 15, 31, 63, 127, 255. Invalid values will result in a communication error (invalid command parameter). An example of a Diagonal Lines pattern is shown in Figure 2-7.

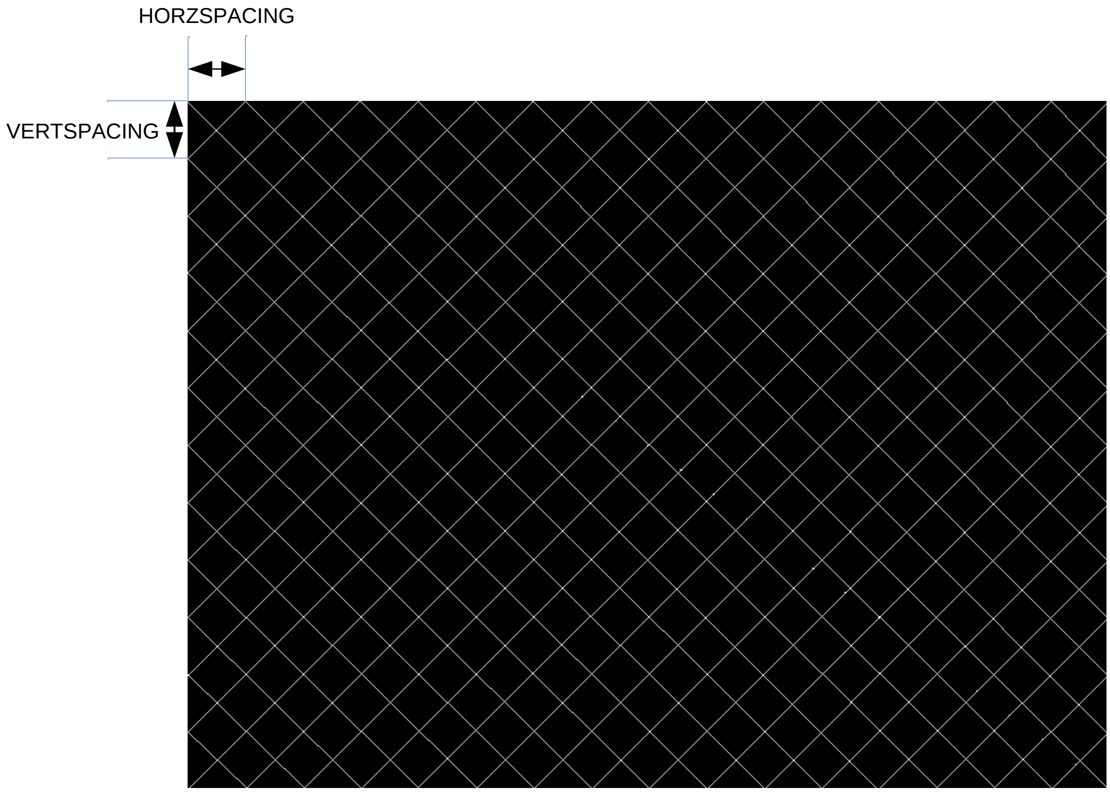

Figure 2-7. Example of Diagonal Lines Test Pattern

Note 14: As noted in Table 2-6, the colors for the Grid Lines pattern are specified using both the Foreground and Background colors. The foreground color is used for the grid lines, and the background color is used for the space between the lines. As noted in Table 2-7, the user specifies the Horizontal Foreground and Background Line Width, as well as the Vertical Foreground and Background Line Width. It is up to the user to determine the line spacing that will meet their needs for each resolution display. As an example, if the horizontal foreground line width = 1, and background line width = 9, there would be a single pixel horizontal line on every 10th line. And if the vertical foreground line width = 1, and background line width = 9, there would be a single pixel vertical line on every 10 th line. An example of a Grid Lines pattern is shown in Figure 2-8.

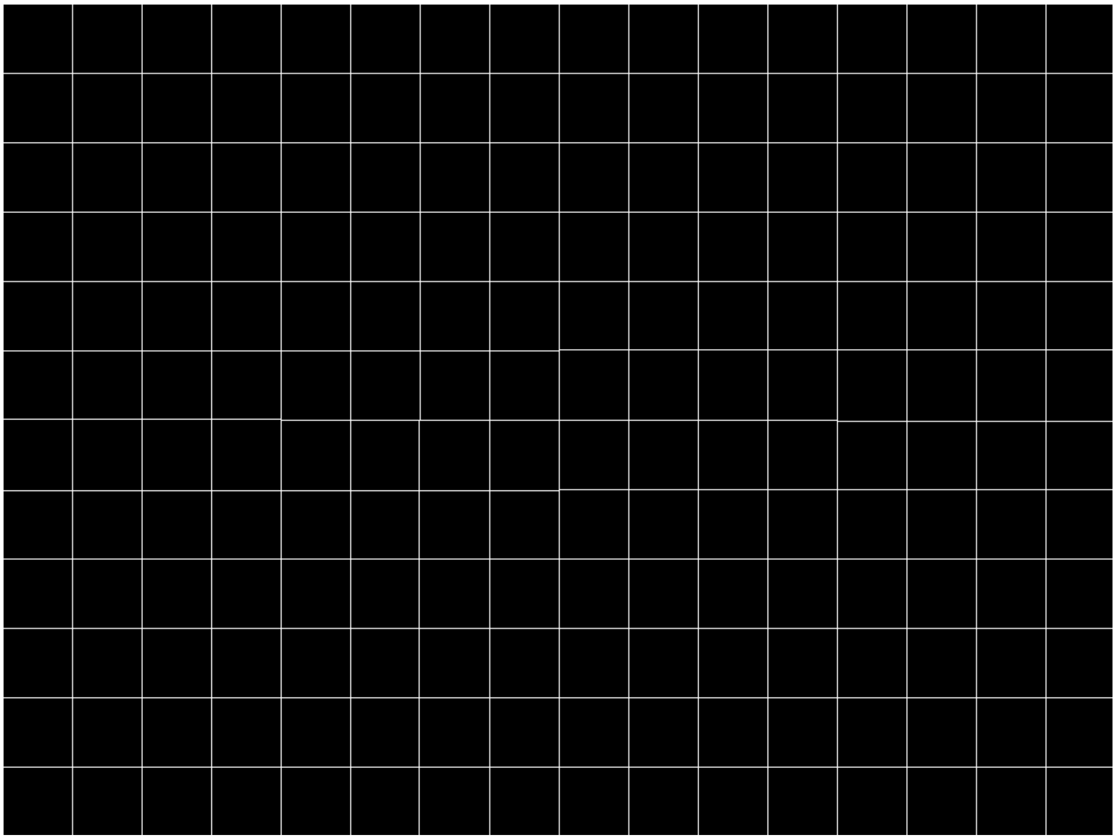

Figure 2-8. Example of Grid Lines Test Pattern

Note 15: As noted in Table 2-6, the colors for the Checkerboard pattern are specified using both the Foreground and Background colors. The foreground color is used for one of the checkers, and the background color is used for the alternating checker. As noted in Table 2-7, the user specifies the Number of Horizontal Checkers and the Number of Vertical Checkers. For this pattern, the system will automatically determine the checker size in each direction based on the number of checkers and the size of the display (DMD). As an example, if the number of horizontal checkers = 4, the number of vertical checkers = 4, and the DMD resolution is 1280x720, the size of the horizontal checkers would be 320 pixels, and the size of the vertical checkers would be 180 pixels (1280 pixels / 4 checkers = 320 pixels: 720 pixels / 4 checkers = 180 pixels). An example of a Checkerboard pattern (16 checker by 12 checker) is shown in Figure 2-9.

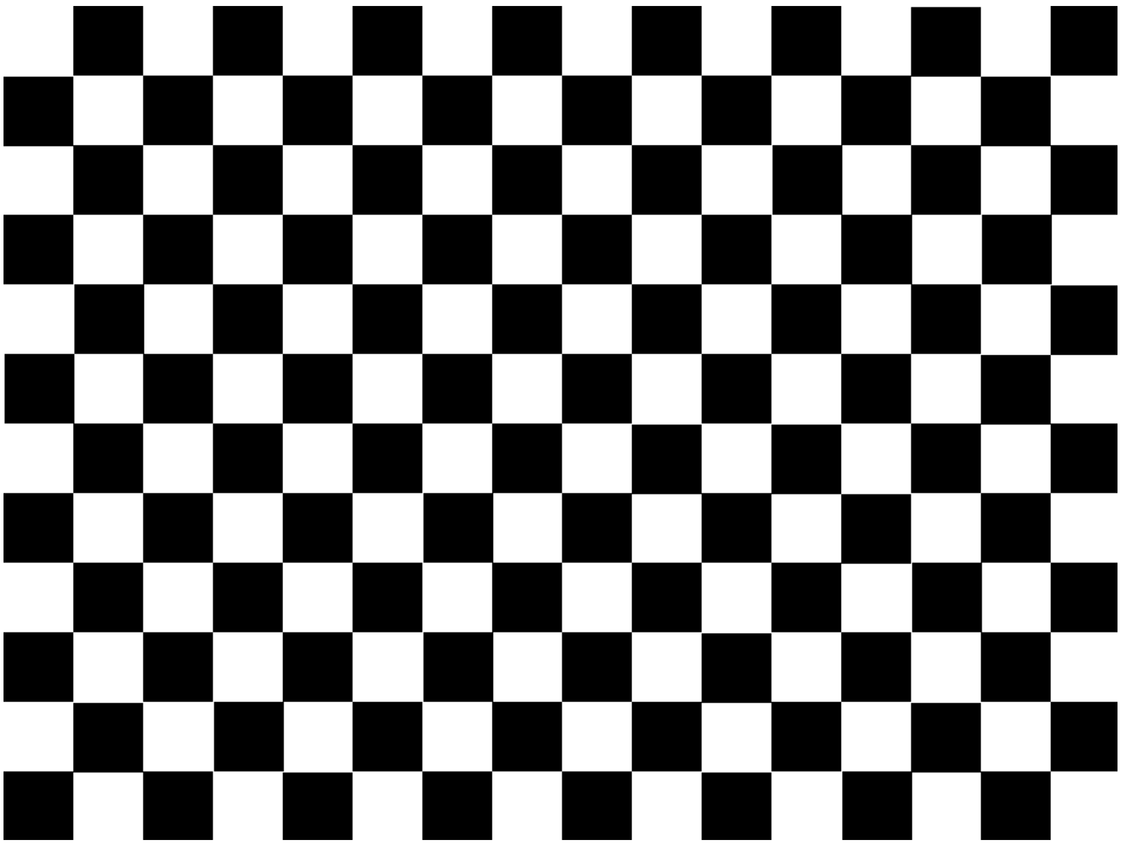

Figure 2-9. Example of Checkerboard Test Pattern

Note 16: As noted in Table 2-6 and Table 2-7, there is no user programmability associated the Color Bars test pattern. This pattern is made up of eight vertical color bars, where the colors are white, yellow, cyan, green, magenta, red, blue, and black. For this pattern, the system will automatically determine the width for each color bar based on the size of the display (DMD). An example of the Color Bars pattern is shown in Figure 2-10.

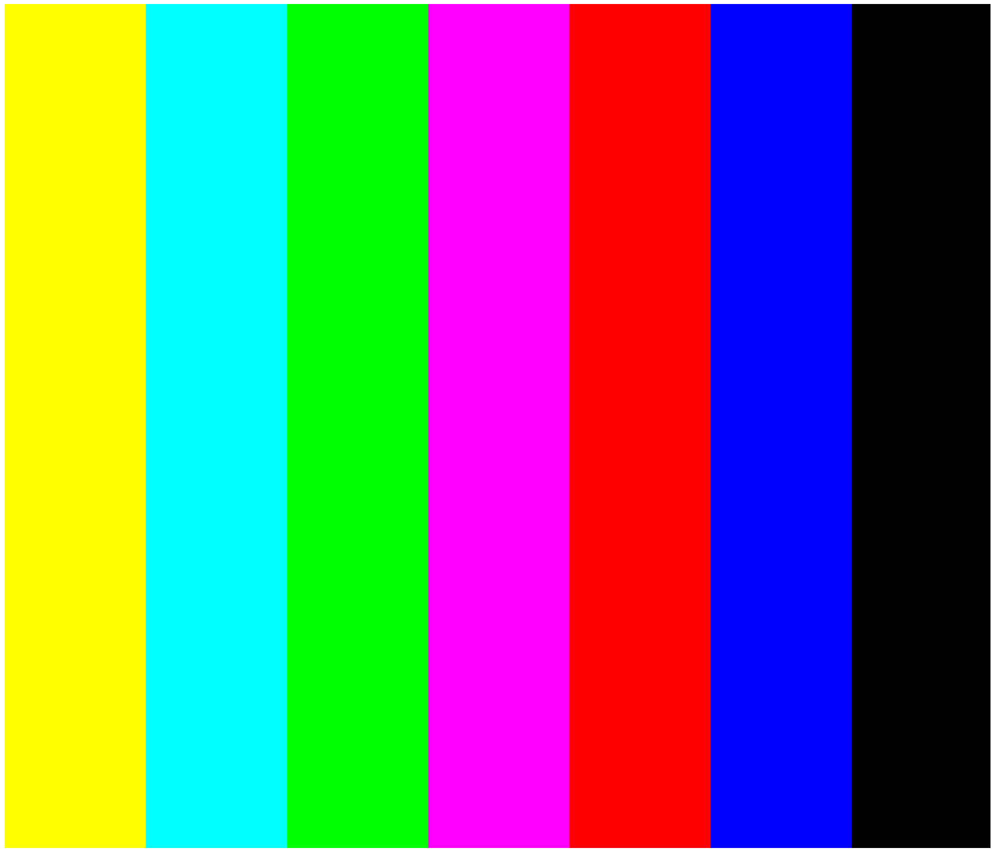

Figure 2-10. Example of Color Bars Test Pattern

2.3.3.9 Read Test Pattern Select (0Ch)

2.3.3.9.1 Read

This command is used to read the state of the test pattern select command for the display module.

2.3.3.9.2 Read Parameters

This command has no command parameters.

2.3.3.9.3 Return Parameters

The return parameters are described below.

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

Parameter Bytes	***Description***	
Byte 1	TPG Pattern Select	(See Below)
Byte 2	Foreground / Background Color	(See Table 2-6)
Byte 3	Parameter 1	(See Table 2-7)
Byte 4	Parameter 2	(See Table 2-7)
Byte 5	Parameter 3	(See Table 2-7)
Byte 6	Parameter 4	(See Table 2-7)

Note 1: This command will always return six bytes, since the host will not know how many bytes will be valid until they know which pattern has been selected. All unneeded bytes (See Table 2-8) will be set to *0*.

Note 2: If a batch file was used to specify the parameters of the test pattern generator, those are the parameters that will be returned by this command.

2.3.3.10 Write Splash Screen Select (0Dh)

2.3.3.10.1 Write

This command is used to select a stored splash screen to be displayed on the display module.

2.3.3.10.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Splash screen reference number (integer)

Default: Composer specified

Note 1: This command is used in conjunction with the *Write Input Source Select* (Section 2.3.3.2) and the *Write Splash Screen Execute* (Section 2.3.3.40) commands. It specifies which splash screen is to be displayed when the *Input Source Select* command selects splash screen as the image source. The settings for this command will be retained until changed using this command.

Note 2: The steps required to display a splash screen are: select the desired splash screen (this command), change the input source to splash screen (using *Write Input Source Select*), and start the splash screen retrieval process (using *Write Splash Screen Execute*).

Note 3: The Splash Screen is a unique source since it is read from Flash and sent down the processing path of the ASIC one time, to be stored in memory for display at the end of the processing path. As such, *all image processing settings* (for example, image crop, image orientation, display size, splash screen select, splash screen as input source, and so forth) *should be set appropriately by the user before executing the Write Splash Screen Execute command*.

Note 4: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note that this command is a source associated command.*

Note 5: The availability of splash screens is limited by the available space in flash memory.

Note 6: All splash screens must be *landscape* oriented.

Note 7: For single ASIC applications which support DMD resolutions up to 1280 × 720, the minimum splash image size allowed for flash storage is 427 × 240, with the maximum being the resolution of the product DMD. Typical splash image sizes for flash are 427 × 240 and 640 × 360. The full resolution size is typically used to support an *Optical Test* splash screen.

Note 8: For dual ASIC applications which support DMD resolutions up to 1980 ×1080, the minimum splash image size allowed for flash storage is 854 × 480, with the maximum being the resolution of the product DMD. Typical splash image sizes for flash are 854 × 480. The full resolution size is typically used to support an *Optical Test* splash screen.

Note 9: The user is responsible for specifying how the splash image will be displayed on the screen. Key commands for this are *Write Image Crop* (Section 2.3.3.13) and *Write Display Size* (Section 2.3.3.15).

Note 10: When this command is received while Splash Screen is the active source, other than storing the specified splash screen value, the only action that will be taken by the ASIC software will be to obtain the header information from the selected splash screen and store this in internal memory. Then, when the *Write Splash Screen Execute* command is received, the ASIC software will use this stored information to set up the processing path prior to pulling the splash data from flash.

2.3.3.11 Read Splash Screen Select (0Eh)

2.3.3.11.1 Read

This command is used to read the state of the splash screen select command for the display module.

2.3.3.11.2 Read Parameters

This command has no command parameters.

2.3.3.11.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	Splash Screen Selected (integer)

Note 1: See Section 2.3.3.10 for more information on splash screens.

2.3.3.12 Read Splash Screen Header (0Fh)

2.3.3.12.1 Read

This command is used to read the splash screen header information for the selected splash screen for the display module.

2.3.3.12.2 Read Parameters

Parameter Bytes	Description
Byte 1	Splash screen reference number (integer)

Note 1: The read parameter is used to specify the splash screen for which the header parameters are to be returned. If a splash screen value is provided for which there is no splash screen available, this will be considered an error (invalid command parameter value – communication status) and the command will not be executed.

Note 2: See Section 2.3.3.10 for more information on splash screens.

2.3.3.12.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description	
Byte 1	Splash Image Width in Pixels	(LSByte)
Byte 2	Splash Image Width in Pixels	(MSByte)
Byte 3	Splash Image Height in Pixels	(LSByte)
Byte 4	Splash Image Height in Pixels	(MSByte)
Byte 5	Splash Image Size in Bytes	(LSByte)
Byte 6	Splash Image Size in Bytes	
Byte 7	Splash Image Size in Bytes	
Byte 8	Splash Image Size in Bytes	(MSByte)
Byte 9	Pixel Format	(See below)
Byte 10	Compression Type	(See below)
Byte 11	Color Order	(See below)
Byte 12	Chroma Order	(See below)
Byte 13	Byte Order	(See below)

Note 1: Parameter definitions referenced are in Table 2-9.

Table 2-9. Splash Screen Header Definitions

Parameter	Values	Parameter	Values
Pixel Format	‘0h' = 24-bit RGB Unpacked (not used) ‘1h' = 24-bit RGB Packed (not used) ‘2h' = 16-bit RGB 5-6-5 ‘3h' = 16-bit YCbCr 4:2:2	Chroma Order	‘0h' = Cr is first pixel ‘1h' = Cb is first pixel
Compression Type	‘0h' = Uncompressed ‘1h' = RGB RLE Compressed ‘2h' = User Defined (not used) ‘3h' = YUV RLE Compressed	Bytes Order	‘0h' = Little Endian ‘1h' = Big Endian
Color Order	‘0h' = 00RRGGBB ‘1h' = 00GGRRBB		

2.3.3.13 Write Image Crop (10h)

2.3.3.13.1 Write

This command image crop can only set non-crop operation, actual crop is not supported in DLPC3439 system.

2.3.3.14 Read Image Crop (11h)

This command Read Image Crop is not supported in DLPC3439.

2.3.3.15 Write Display Size (12h)

2.3.3.15.1 Write

This command is used to specify the size of the active image to be displayed on the display module.

2.3.3.15.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description	
Byte 1	Pixels per Line	(LSByte)
Byte 2	Pixels per Line	(MSByte)
Byte 3	Lines per Frame	(LSByte)
Byte 4	Lines per Frame	(MSByte)

Default: DMD Resolution

Note 1: This specifies the size of the image to be output from the scaler function, which will be the size of the active displayed image.

Note 2: The parameter values are to be ‘1' based. (in other words, a value of 1280 pixels will display 1280 pixels per line).

Note 3: All sub-images (images smaller than the DMD display) will be horizontally and vertically centered on the display (DMD)

Note 4: If the display size exceeds the resolution of the DMD, this will be considered an error (invalid command parameter value – communication status) and the command will not be executed. *Specifically*, the display size parameters will be checked against the DMD resolution, and if the DMD resolution is exceeded in *the* orientation, it will be considered an error. Note that the system will not check for proper image orientation setup.

DMD resolution = 854 × 480:

Example 1: Display size parameter = 480 × 854 (not an error)

Example 2: Display size parameter = 900 × 320 (error)

Example 3: Display size parameter = 500 × 600 (error)

Note 5: If the source, crop, and display parameter combinations exceed the capabilities of the scaler, the system will implement what was requested by the user as best it can, and the displayed image will be what it will be (in other words, a *broken* image may be displayed). It will be up to the user to provide updated parameters to *fix* the image.

2.3.3.16 Read Display Size (13h)

2.3.3.16.1 Read

This command is used to read the state of the display size command for the display module

2.3.3.16.2 Read Parameters

This command has no command parameters.

2.3.3.16.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description
Byte 1	Pixels per Line (LSByte)
Byte 2	Pixels per Line (MSByte)
Byte 3	Lines per Frame (LSByte)
Byte 4	Lines per Frame (MSByte)

Note 1: The parameter values are to be *1* based. (In other words, a value of 1920 pixels will display 1920 pixels per line.)

2.3.3.17 Write Display Image Orientation (14h)

2.3.3.17.1 Write

This command is used to specify the image orientation of the displayed image for the display module.

2.3.3.17.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – Short Axis Image Flip
0h: Image not flipped
1h: Image flipped

b(1) – Long Axis Image Flip
0h: Image not flipped
1h: Image flipped

b(0) – *Reserved*

Default: Composer specified

Note 1: Image rotation is not supported in DLPC3439.

Note 2: Short axis flip is as shown in Figure 2-11.

Figure 2-11. Short Axis Flip

2.3.3.18 Read Display Image Orientation (15h)

2.3.3.18.1 Read

This command is used to read the state of the displayed image orientation function for the display module.

2.3.3.18.2 Read Parameters

This command has no command parameters.

2.3.3.18.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – Short Axis Image Flip
0h: Image not flipped
1h: Image flipped

b(1) – Long Axis Image Flip
0h: Image not flipped
1h: Image flipped

b(0) – *Reserved*

2.3.3.19 Write Display Image Curtain (16h)

2.3.3.19.1 Write

This command is used to control the display image curtain for the display module.

2.3.3.19.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4) – *Reserved*

b(3:1) – Select Curtain Color
0h: Black
1h: Red
2h: Green
3h: Blue
4h; Cyan
5h: Magenta
6h: Yellow
7h: White

b(0) – Curtain Enable
0h: Curtain Disabled
1h: Curtain Enabled

Default: 01h

Note 1: The Image Curtain fills the entire display with a user specified color.

Note 2: The curtain color specified by this command is separate from the border color defined in the *Write Border Color* command (Section 2.3.3.61), even though they are both displayed using the curtain capability.

2.3.3.20 Read Display Image Curtain (17h)

2.3.3.20.1 Read

This command is used to read the state of the image curtain control function for the display module.

2.3.3.20.2 Read Parameters

This command has no command parameters.

2.3.3.20.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4) – *Reserved*

b(3:1) – Select Curtain Color
0h: Black
1h: Red
2h: Green
3h: Blue
4h; Cyan
5h: Magenta
6h: Yellow
7h: White

b(0) – Curtain Enable
0h: Curtain Disabled
1h: Curtain Enabled

2.3.3.21 Write Image Freeze (1Ah)

2.3.3.21.1 Write

This command is used to enable or disable the image freeze function for the display module.

2.3.3.21.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – Image Freeze
0h: Image Freeze Disabled
1h: Image Freeze Enabled

Default: 00h

Note 1: Normal use of the Image Freeze capability typically has two main functions. The first function is to allow the end user to freeze the current image on the screen for their own uses. The second function is to allow the user (host system/OEM) to reduce/prevent system changes from showing up on the display as visual artifacts. In this second case, the image would be frozen, system changes would be made, and when complete, the image is unfrozen. In all cases, when the image is unfrozen, the display starts showing the most resent input image. Thus input data between the freeze point and the unfreeze point is lost. Suggestions to the host system for the types of image changes likely to necessitate the use of the image freeze command to hide artifacts are discussed in Section 2.3.3.21.3.

Note 2: It should be noted that the ASIC software will never *automatically* or *under-the-hood* freeze or unfreeze the image. Basically, the ASIC software will not freeze or unfreeze the image for any reason except when explicitly commanded by the Write Image Freeze command.

Note 3: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note, Freeze command doesn't work on Splash screen on dual DLPC3439 system.*

Note 4: If the OEM chooses not to make use of Image Freeze, is recommended that they change the source itself before changing image parameters to minimize transition artifacts.

2.3.3.21.3 Use of Image Freeze to Reduce On-Screen Artifacts

Commands that take a long time to process, require a lot a data to be loaded from Flash, or change the frame timing of the system, have the potential to create on-screen artifacts. The *Write Image Freeze* command can be used in these cases to try and minimize, if not eliminate, these artifacts. The process would be:

1. Send *Write Image Freeze* command to enable freeze.
2. Send commands with the potential to create image artifacts.
3. Send *Write Image Freeze* command to disable freeze.

Since commands to the ASIC are processed serially, no special timing or delay is required between these commands. It is suggested that the number of commands placed between the freeze and unfreeze be kept small, as it is likely not desirable for the image to be frozen for a *long* period of time. A list of commands that may product image artifacts are listed in Table 2-10. This is not an all inclusive list, however, and the user is ultimately responsible for determining if and when use of the image freeze command will meet their product needs.

Table 2-10. Partial List of Commands that may Benefit from use of Image Freeze

Command (1) (2)	Command OpCode	Paragraph	Notes
Write Input Source Select	05h	Section 2.3.3.2	
Write External Video Source Format Select	07h	Section 2.3.3.4	
Write Test Pattern Select	0Bh	Section 2.3.3.8	
Write Look Select	22h	Section 2.3.3.25	

(1) If changed while this source is the active source.
(2) Freeze command does not work on Splash screen.

A few examples of how to use the image freeze command are shown in Table 2-11 and Table 2-12.

Table 2-11. TPG Example Using Image Freeze

Command	Notes
Write Display Image Curtain = Enable	May want to apply curtain if already displaying unwanted image (For example, a broken source)
Write Image Freeze = Freeze	
Write Image Crop, Write Display Size, Write Display Image Orientation.	Potential data processing commands that may be required for proper display of TPG
Write TPG Select Write Input Source Select = TPG	Set up TPG
Write Image Freeze = Unfreeze	

Table 2-12. Test Pattern Generator Example using Image Freeze

Command	Notes
Write Image Freeze = Freeze	
Write Image Crop, Write Display Size, Write Display Image Orientation, Write Test Pattern Select.	Potential data processing commands that may be required for proper display of test pattern image. These would be used as appropriate. It is recommended that these be set before the Write Input Source Select command.
Write Input Source Select = Test Pattern Generator	
Write Image Freeze = Unfreeze	

2.3.3.22 Read Image Freeze (1Bh)

2.3.3.22.1 Read

This command is used to read the state of the image freeze function for the display module.

2.3.3.22.2 Read Parameters

This command has no command parameters.

2.3.3.22.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – Image Freeze
0h: Image Freeze Disabled
1h: Image Freeze Enabled

2.3.3.23 Write 3-D Control (20h)

2.3.3.23.1 Write

This command is used to control 3-D functionality for the display module.

2.3.3.23.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – *Reserved*

b(6) – Polarity of 3-D Reference (External Only)
0h: Correct – No Inversion Required
1h: Incorrect – Inversion Required

b(5) – Frame Dominance
0h: Left Dom. (Data sent left eye first)
1h: Right Dom. (Data sent right eye first)

b(4:2) – *Reserved*

b(1) – Source of 3-D Reference
0h: Internal Reference Generator NOT supported
1h: External (SLT_3DR Pin)

b(0) – *Reserved*

Default: 00h

Note 1: The system will automatically enable 3-D operation when appropriate, basing this decision on the source frame rate and whether 3-D sequences are available to the system (that is, loaded in flash). The 3-D parameters specified by this command will take effect following the next VSYNC.

Note 2: 3-D image data must always be sent frame sequential (that is, syncs and blanking to be sent between every eye frame), at frame rates greater than approximately 94Hz (ASIC does not support frame rate multiplication). Internal Reference Generator is not supported in Dual ASIC system.

Note 3: Internal reference generator is not supported on dual ASIC DLPC3439.

Note 4: The 3-D Reference is used to specify whether a frame of data contains left eye data or right eye data. This 3-D reference can be provided to the display by an external hardware signal. Table 2-13 shows which 3-D Reference source can be used with which image data port.

When using the external hardware signal as the reference, it must be provided for every frame of data. If the external 3-D Reference is misaligned with the data, it can be corrected using the *Polarity of 3-D Reference (External Only)* parameter. As noted, the *Polarity of 3-D Reference* parameter is only applicable when the External Signal is selected as the 3-D Reference source.

Table 2-13. 3-D Reference Source Applicability for Display Data Ports

Display Data Port	3-D Reference Source [(1)]	Applicable	Notes
Parallel	External Hardware Signal	Yes	Recommended
Parallel	Internal Reference Generator	No	

(1) The *Write 3-D Reference* command should be use with this selection.

Note 5: For frame sequential 3-D, *Frame Dominance* determines which eye frames in the data stream go together to make up a single 3-D image. Left Dominance indicates that the first eye frame of a pair is Left, the second eye frame is Right. Right Dominance indicates that the first eye frame of a pair is Right, the second eye frame is Left). This is important for proper operation of display histograms (which span both eye frames of a single image), and when the image is frozen, as we want to be sure we display the correct two eye frames together. *The frame dominance control must not be used to attempt correction for misalignment of the 3-D reference signal to the image data.*

2.3.3.24 Read 3-D Control (21h)

2.3.3.24.1 Read

This command is used to read the state of the 3-D control function for the display module.

2.3.3.24.2 Read Parameters

This command has no command parameters.

2.3.3.24.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – *Reserved*

b(6) – Polarity of 3-D Reference (External Only)
0h: Correct – No Inversion Reqd.
1h: Incorrect – Inversion Reqd

b(5) – Frame Dominance
0h: Left Dom. (Data sent left eye first)
1h: Right Dom. (Data sent right eye first)

b(4:2) – *Reserved*

b(1) – Source of 3-D Reference
0h: Internal Reference Generator NOT supported
1h: External (SLT_3DR Pin)

b(0) – 3-D Mode Control
0h: 2-D Operation
1h: 3-D Operation

Note 1: The system automatically enables and disables 3-D operation. Bit(0) will indicate the state of 2-D/3-D operation.

2.3.3.25 Write LOOK Select (22h)

2.3.3.25.1 Write

This command is used to specify the LOOK for the image on the display module.

2.3.3.25.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – LOOK Number

Default: Composer specified

Note 1: In this product, a LOOK typically specifies a target white point.

Note 2: This command allows the host to select a LOOK (target white point) from a number of looks stored in flash. Based on the look selected, along with measured data obtained from an appropriate light sensor, software will automatically select and load the most appropriate sequence/duty cycle set available in the Look to get as close as possible to the target white point.

Note 3: The number of LOOKs available may be limited by the available space in flash memory.

Note 4: LOOKs are specified in this byte by an enumerated value (that is, 0, 1, 2, 3, and so forth.). There must always be at least one look, whose enumerated value will be *0*.

Note 5: The OEM may also specify the desired white point manually using the *Write Manual Duty Cycle Select* command. This manual command would most likely be used when there is no light sensor in the system, in which case, the host becomes responsible for sequence/duty cycle selection from the sequences available in the selected Look. Note that duty cycle sets are associated with sequences which are stored in Flash until needed.

Note 6: There are two other items that the host will likely want to specify in addition to the LOOK. These are:

- *A desired degamma curve*: This is achieved by selecting the appropriate Degamma/CMT which has the desired degamma curve and correct bit weights for the sequence selected. This is selected using the *Degamma/CMT Select* command discussed in Section 2.3.3.29.
- *The desired color points*: This is achieved by selected the appropriate CCA parameters, and is selected using the CCA Select command discussed in Section 2.3.3.30.

2.3.3.26 Read LOOK Select (23h)

2.3.3.26.1 Read

This command is used to read the state of the look select command for the display module.

2.3.3.26.2 Read Parameters

This command has no command parameters.

2.3.3.26.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description	
Byte 1	Look number	See Below
Byte 2	Sequence number	See Below
Byte 3	Current Sequence Frame Rate (lsb)	See Note 3
Byte 4	Current Sequence Frame Rate	
Byte 5	Current Sequence Frame Rate	
Byte 6	Current Sequence Frame Rate (msb)	

msb	Byte 1						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – LOOK Number

msb	Byte 2						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – Sequence Number

Note 1: LOOKs are specified by an enumerated value (that is, 0, 1, 2, 3, and so forth.)

Note 2: Sequences are specified by an enumerated value (that is, 0, 1, 2, 3, and so forth.), and the value returned by this command is the sequence currently selected by the LOOK algorithm when this command is received.

Note 3: The current sequence frame rate is returned as a count that is specified in units of 66.67ns (based on the internal 15MHz clock used to time between input frame syncs), and is valid regardless of whether ASIC software made the sequence/duty cycle selection, or the user made the selection. The frame rate is specified in this way to enable fast and simple compares to the frame count by software.

2.3.3.27 Read Sequence Header Attributes (26h)

2.3.3.27.1 Read

This command is used to read sequence header information for the active sequence of the display module.

2.3.3.27.2 Read Parameters

This command has no command parameters.

2.3.3.27.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description		
Byte 1	Red Duty Cycle	(LSByte)	(Look Structure)
Byte 2	Red Duty Cycle	(MSByte)	(Look Structure)
Byte 3	Green Duty Cycle	(LSByte)	(Look Structure)
Byte 4	Green Duty Cycle	(MSByte)	(Look Structure)
Byte 5	Blue Duty Cycle	(LSByte)	(Look Structure)
Byte 6	Blue Duty Cycle	(MSByte)	(Look Structure)
Byte 7	Maximum Frame Count	(LSByte)	(Look Structure)
Byte 8	Maximum Frame Count		(Look Structure)
Byte 9	Maximum Frame Count		(Look Structure)
Byte 10	Maximum Frame Count	(MSByte)	(Look Structure)
Byte 11	Minimum Frame Count	(LSByte)	(Look Structure)
Byte 12	Minimum Frame Count		(Look Structure)
Byte 13	Minimum Frame Count		(Look Structure)
Byte 14	Minimum Frame Count	(MSByte)	(Look Structure)
Byte 15	Max # of Seq Vectors	(See below)	(Look Structure)
Byte 16	Red Duty Cycle	(LSByte)	(Sequence Structure)
Byte 17	Red Duty Cycle	(MSByte)	(Sequence Structure)
Byte 18	Green Duty Cycle	(LSByte)	(Sequence Structure)
Byte 19	Green Duty Cycle	(MSByte)	(Sequence Structure)
Byte 20	Blue Duty Cycle	(LSByte)	(Sequence Structure)
Byte 21	Blue Duty Cycle	(MSByte)	(Sequence Structure)
Byte 22	Maximum Frame Count	(LSByte)	(Sequence Structure)
Byte 23	Maximum Frame Count		(Sequence Structure)
Byte 24	Maximum Frame Count		(Sequence Structure)
Byte 25	Maximum Frame Count	(MSByte)	(Sequence Structure)
Byte 26	Minimum Frame Count	(LSByte)	(Sequence Structure)
Byte 27	Minimum Frame Count		(Sequence Structure)
Byte 28	Minimum Frame Count		(Sequence Structure)
Byte 29	Minimum Frame Count	(MSByte)	(Sequence Structure)
Byte 30	Max # of Seq Vectors	(See below)	(Sequence Structure)

Note 1: The sequence header data is stored in two separate Flash data structures (the LOOK Structure and the Sequence Structure), and the values from each should match.

Note 2: The bit weight and bit order for the duty cycle data is shown in Figure 2-12.

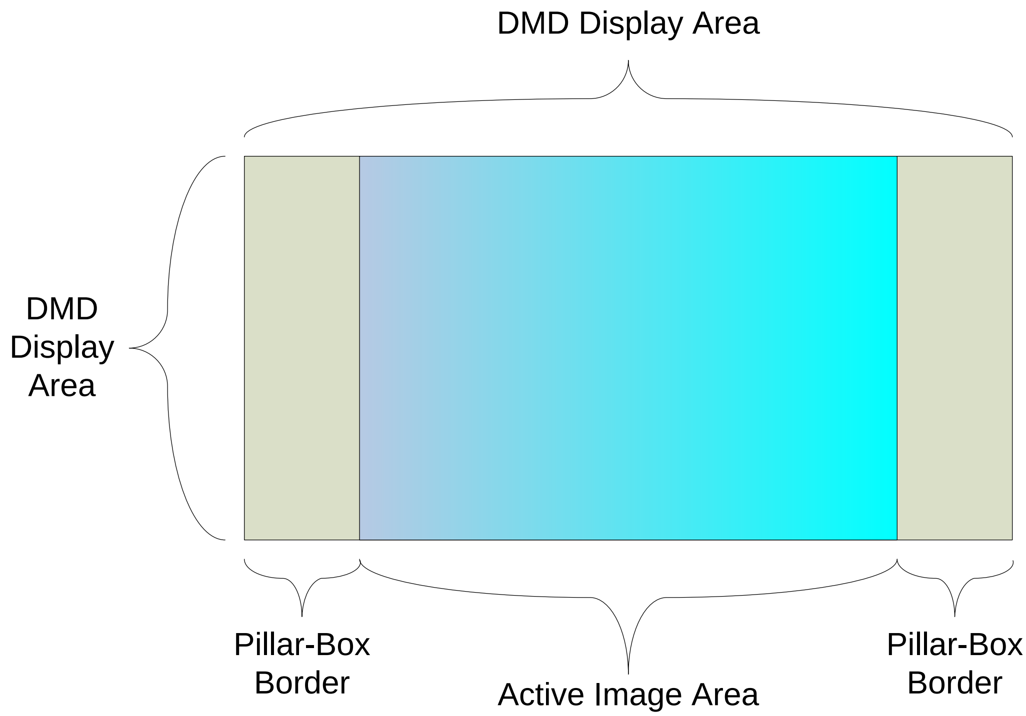

Figure 2-12. Bit Weight and Bit Order for Duty Cycle Data

Note 3: The duty cycle data is specified as each colors percent of the frame time. The sum of the three duty cycles must add up to 100.
(ex. R = 30.5 = 1E80h , G = 50 = 3200h, B = 19.5 = 1380h)

Note 4: The sequence maximum and minimum frame counts are specified in units of 66.67ns (based on the internal 15MHz clock used to time between input frame syncs). These are specified in this way to enable fast and simple compares to the frame count by software.

Note 5: The Max # of Sequence Vectors byte is defined below.

msb	***Byte 15 and 30***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4) – *Reserved*

b(3:0) – Max # of Sequence Vectors

2.3.3.28 Write Gamma/CMT Select (27h)

2.3.3.28.1 Write

This command is used to select a specific Degamma/CMT LUT for the display module.

2.3.3.28.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – Degamma/CMT LUT Index Number

Default: Composer specified

Note 1: Degamma/CMT LUTs are stored in Flash until needed.

Note 2: The Degamma/CMT LUT Number specified by the user determines the degamma applied by the system.

Note 3: For TI software purposes, this Degamma/CMT LUT number is the CMT Index # in the Flash structure. Thus, if there is a degamma of 1.5 (for example) at CMT Index #0, then every sequence will have a CMT Index #0 that references a degamma of 1.5 that is appropriate for each respective sequence.

2.3.3.29 Read Degamma/CMT Select (28h)

2.3.3.29.1 Read

This command is used to read the status of the degamma/CMT select command for the display module.

2.3.3.29.2 Read Parameters

This command has no command parameters.

2.3.3.29.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – Degamma/CMT LUT Number

2.3.3.30 Write CCA Select (29h)

2.3.3.30.1 Write

This command is used to select a specific set of CCA parameters (to specify the color points) for the display module.

2.3.3.30.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – CCA Parameter Set

Default: Composer specified

Note 1: CCA parameter sets are used to set a target color points for the system. The sets are stored in Flash until needed.

Note 2: CCA parameter sets are specified in this byte by an enumerated value (that is, 0, 1, 2, 3, and so forth.). This number specifies the actual CCA number reference in the flash structure.

2.3.3.31 Read CCA Select (2Ah)

2.3.3.31.1 Read

This command is used to read the status of the CCA select command for the display module.

2.3.3.31.2 Read Parameters

This command has no command parameters.

2.3.3.31.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – CCA Parameter Set

Note 1: CCA parameter sets are specified in this byte by an enumerated value (that is, 0, 1, 2, 3, and so forth.)

2.3.3.32 Write Execute Flash Batch File (2Dh)

2.3.3.32.1 Write

This command is used to command the execution of a Flash batch file for the display module.

2.3.3.32.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	Batch File Number

Note 1: This command is used to command the execution of a batch file stored in the Flash of the display module. Any system *write* command that can be sent by itself can be grouped together with other system commands or command parameters into a Flash batch file, with the exception of those listed in . Flash batch files are created using the DLP Composer tool, and then stored in the Flash build. One example for a Flash batch file might be the commands and command parameters required for initialization of the system after power-up.

Note 2: The Flash batch file numbers to be specified in this byte are enumerated values (that is, 0, 1, 2, 3, and so forth.).

Note 3: Flash batch file 0 is a special Auto-Init batch file that is run automatically by the DLPC3439 software immediately after system initialization has been completed. As such, Flash batch file 0 will not typically be called using the Write Execute Batch File command (although the system will allow it). This special Flash batch file would typically be used to specify the source to be used (for example, splash screen or data port) once the system is initialized.

Note 4: Embedding Flash batch file calls within a Flash batch file is not allowed (that is, calling another batch file from within a batch file is not allowed). If it is desired to have two batch files executed back to back, they should be called by back to back execute batch file commands.

Note 5: The system provides the ability to add an execution delay between commands within a Flash batch file. This is done using the *Write Flash Batch File Delay* command (Section 2.3.3.76).

Note 6: The order of command execution for commands within a Flash batch file will be the same as if the commands had been received over the I^2C port.

Table 2-14. List of Commands Excluded from Batch File Use

Command	Op-Code	Paragraph
Write Command Synchronization	N/A	-
Write Execute Flash Batch File	2D	Section 2.3.3.32
Flash Data Type Select	DE	Section 2.3.3.78
Flash Data Length	DF	0
Erase Flash Data	E0	Section 2.3.3.81
Write Flash Start	E1	Section 2.3.3.82
Write Flash Continue	E2	Section 2.3.3.83
Write Internal Mailbox Address	E8	Section 2.3.3.89
Write Internal Mailbox	E9	Section 2.3.3.90
Write External PAD Address	EB	Section 2.3.3.92
Write External PAD Data	EC	Section 2.3.3.93
All *Read* commands	Various	Various

2.3.3.33 Write External Input Image Size (2Eh)

2.3.3.33.1 Write

This command is used to specify the active data size of the external input image to the display module.

2.3.3.33.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description	
Byte 1	Pixels per Line	(LSByte)
Byte 2	Pixels per Line	(MSByte)
Byte 3	Lines per Frame	(LSByte)
Byte 4	Lines per Frame	(MSByte)

Default: DMD Resolution

Note 1: This command is used in conjunction with the *Write Input Source Select* command (Section 2.3.3.2). This command specifies the active data size of the input image to the system for all external video interfaces when the *Write Input Source Select* command selects External Video Port as the image source. The settings for this command are to be retained until changed using this command. These settings will be automatically applied each time the External Video Port is selected.

Note 2: When the source data for the parallel interface does not provide an active data framing signal, the user must specify where the active data is located within the frame using the *Write Parallel I/F Manual Image Framing* command (Section 2.3.3.42) in addition to this command.

Note 3: The parameter values are to be *1* based. (That is, a value of 1280 pixels will specify 1280 pixels per line.)

Note 4: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note that this command is a source associated command.*

Note 5: The maximum and minimum input values are shown in Table 2-15. Values outside of these ranges will be flagged as an error (invalid command parameter), and the command will not be executed.

Table 2-15. Input Source Limits for Active Data

Parameter		Minimum Value	Maximum Value
Input Source Active Pixels per Line	(Single ASIC)	320	1280
Input Source Active Lines per Frame	(Single ASIC)	200	800
Input Source Active Pixels per Line	(Dual ASIC)	1280 [1]	1920 [1]
Input Source Active Lines per Frame	(Dual ASIC)	720 [1]	1080 [1]

(1) Limited Scaling is supported for dual ASIC configurations.

2.3.3.34 Read External Input Image Size (2Fh)

2.3.3.34.1 Read

This command is used to read the specified data size of the external input image to the display module

2.3.3.34.2 Read Parameters

This command has no command parameters.

2.3.3.34.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description	
Byte 1	Pixels per Line	(LSByte)
Byte 2	Pixels per Line	(MSByte)
Byte 3	Lines per Frame	(LSByte)
Byte 4	Lines per Frame	(MSByte)

Note 1: The parameter values are to be *1* based. (That is, a value of 1280 pixels will specify 1280 pixels per line.)

Note 2: This command returns the value specified by the *Write External Input Image Size* command (Section 2.3.3.33).

2.3.3.35 Write 3-D Reference (30h)

2.3.3.35.1 Write

This command is used to provide a 3-D reference for the display module.

2.3.3.35.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – 3-D Reference
0h: Next Frame Left
1h: Next Frame Right

Default: 00h

Note 1: The 3-D Reference is used to specify whether a frame of data contains left eye data or right eye data. The 3-D reference can be provide to the display as a GPIO hardware signal or by using this command (selection is made using the *Write 3-D Control* command (Section 2.3.3.23)). When using this command as the reference, it is recommend that the command be sent every frame, or at least at the start of each eye pair (for example, sent before each left eye frame). At a minimum, it must be sent once at the start of 3-D operation. If the 3-D Reference is misaligned with the data, it can be corrected using this command or by using the *Polarity of 3-D Reference* parameter in the *Write 3-D Control* command.

Note 2: When the *Write 3-D Reference* command is received, its parameter value will be applied at the next VSYNC (In other words, the parameter value will be applied to the image data following the next VSYNC or Start of Frame command.)

Note 3: When this command is received, software must set up the internal ASIC 3-D reference generator. If the command is sent every frame, software can just monitor to insure that the output of the internal ASIC 3-D reference generator is still correct.

2.3.3.36 Write GPIO[19:00] Control (31h)

2.3.3.36.1 Write

This command is used to specify how GPIO(19:09) and GPIO(07:04) of the display module are to be used.

2.3.3.36.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description
Byte 1	See Below
Byte 2	See Below
Byte 3	See Below
Byte 4	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – GPIO(12)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(11)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(09)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(10)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

msb	***Byte 2***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – GPIO(16)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(15)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(14)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(13)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

msb	***Byte 3***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(19)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(18)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(17)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

msb	***Byte 4***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – GPIO(07)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(06)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(05)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(04)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

Default: Composer specified

Note 1: GPIO(19:09) and GPIO(07:04) can individually function as defined by the OEM in Composer, or their function can be redefined on-the-fly using this command. When their functions are redefined by this command, the *Write GPIO[19:00] Outputs* (Section 2.3.3.38) and *Read GPIO[19:00] Inputs* (Section 2.3.3.41) commands are used to write or read the redefined GPIO. GPIO(08) has a fixed function, and the functions of GPIO(03:00) can only be specified via Composer.

Note 2: The OEM must insure that signal conflicts do not arise when switching GPIO signal directions (for example, external signal driving a GPIO that is configured as an output).

2.3.3.37 Read GPIO[19:00] Control (32h)

2.3.3.37.1 Read

This command is used to read back the control state for GPIO(19:09) and FPIO(07:04) of the display module.

2.3.3.37.2 Read Parameters

This command has no command parameters.

2.3.3.37.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description
Byte 1	See Below
Byte 2	See Below
Byte 3	See Below
Byte 4	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – GPIO(12)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(11)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(10)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(09)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

msb	*Byte 2*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – GPIO(16)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(15)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(14)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(13)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

msb	*Byte 3*						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(19)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(18)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(17)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

msb	*Byte 4*						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – GPIO(07)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(5:4) – GPIO(06)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(3:2) – GPIO(05)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

b(1:0) – GPIO(04)
0h: Composer Defined
1h: Input
2h: Output (Standard)
3h: Output (Open Drain)

2.3.3.38 Write GPIO[19:00] Outputs (33h)

2.3.3.38.1 Write

This command is used to write the output values for GPIO(19:09) and GPIO(07:00) of the display module.

2.3.3.38.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description	
Byte 1	GPIO Mask	(See Below)
Byte 2	GPIO Mask	(See Below)
Byte 3	GPIO Mask	(See Below)
Byte 4	GPIO Value	(See Below)
Byte 5	GPIO Value	(See Below)
Byte 6	GPIO Value	(See Below)

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(7)
0h: Not Selected
1h: Selected

b(6) – GPIO(6)
0h: Not Selected
1h: Selected

b(5) – GPIO(5)
0h: Not Selected
1h: Selected

b(4) – GPIO(4)
0h: Not Selected
1h: Selected

b(3) – GPIO(3)
0h: Not Selected
1h: Selected

b(2) – GPIO(2)
0h: Not Selected
1h: Selected

b(1) – GPIO(1)
0h: Not Selected
1h: Selected

b(0) – GPIO(0)
0h: Not Selected
1h: Selected

msb	*Byte 2*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(7)
0h: Not Selected
1h: Selected

b(6) – GPIO(6)
0h: Not Selected
1h: Selected

b(5) – GPIO(5)
0h: Not Selected
1h: Selected

b(4) – GPIO(4)

b(3) – GPIO(3)
0h: Not Selected
1h: Selected

b(2) – GPIO(2)
0h: Not Selected
1h: Selected

b(1) – GPIO(1)
0h: Not Selected
1h: Selected

b(0) – GPIO(0)

0h: Not Selected
1h: Selected

0h: Not Selected
1h: Selected

msb	Byte 3						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*
0h: Not Selected
1h: Selected

b(2) – GPIO(19)
0h: Not Selected
1h: Selected

b(1) – GPIO(18)
0h: Not Selected
1h: Selected

b(0) – GPIO(17)
0h: Not Selected
1h: Selected

msb	Byte 4						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(7)
b(6) – GPIO(6)
b(5) – GPIO(5)
b(4) – GPIO(4)

b(3) – GPIO(3)
b(2) – GPIO(2)
b(1) – GPIO(1)
b(0) – GPIO(0)

msb	Byte 5						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(16)
b(6) – GPIO(15)
b(5) – GPIO(14)
b(4) – GPIO(13)

b(3) – GPIO(312)
b(2) – GPIO(11)
b(1) – GPIO(10)
b(0) – GPIO(9)

msb	Byte 6						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*
b(2) – GPIO(19)

b(1) – GPIO(18)
b(0) – GPIO(9)

Default: Composer specified

Note 1: GPIO(19:09) and GPIO(07:04) can function as defined by the OEM in Composer, or their function can be redefined on-the-fly using the *Write GPIO[19:00] Control* command (Section 2.3.3.36) .

GPIO(08) is not re-definable on-the-fly, and is not available for use as an OEM GPIO.

GPIO(03:00) can only function as defined by the OEM in Composer. One of the choices in Composer is to define one or more of these GPIO to be OEM GPIO Outputs.

Note 2: When one or more of GPIO(19:09) and GPIO(07:04) are defined as output signals (using the *Write GPIO[19:00] Control* command), and/or one or more of GPIO(03:00) are defined as output signals by the OEM in Composer, this command is used to specify the values of those output signals. All of these values will be retained for later application if required, and for read back using the *Read GPIO(19:00) Outputs* command (Section 2.3.3.40).

This command will have no effect on GPIO(19:09) and GPIO(07:04) that are not defined as output signals by the *Write GPIO[19:00] Control* command, although any values entered for these GPIO will be retained (and used if/when these GPIO are later specified to be outputs).

This command will have no effect on GPIO(03:00) that are not defined as output signals by Composer. Even so, any values entered for these GPIO will be retained for read back (although they will not be applied to the GPIO).

Note 3: In order to set the value of a GPIO, the GPIO must be selected using bytes 1 to 3 of this command, with the appropriate value then being specified using bytes 3 to 6.

2.3.3.39 Read GPIO[19:00] Outputs (34h)

2.3.3.39.1 Read

This command is used to read the output values for GPIO(19:09) and GPIO(07:00) of the display module.

2.3.3.39.2 Read Parameters

This command has no command parameters.

2.3.3.39.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description	
Byte 1	GPIO Value	(See Below)
Byte 2	GPIO Value	(See Below)
Byte 3	GPIO Value	(See Below)

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(7)
b(6) – GPIO(6)
b(5) – GPIO(5)
b(4) – GPIO(4)
b(3) – GPIO(3)
b(2) – GPIO(2)
b(1) – GPIO(1)
b(0) – GPIO(0)

msb	***Byte 2***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(16)
b(6) – GPIO(15)
b(5) – GPIO(14)
b(4) – GPIO(13)
b(3) – GPIO(12)
b(2) – GPIO(11)
b(1) – GPIO(10)
b(0) – GPIO(9)

msb	***Byte 3***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*
b(2) – GPIO(19)
b(1) – GPIO(18)
b(0) – GPIO(17)

Note 1: This command will return the values specified by the *Write GPIO[19:00] Outputs* command (Section 2.3.3.38). Any GPIO not having a value specified by the *Write GPIO[19:00] Outputs* command will return a value of zero. The value returned may or may not be the value at the GPIO. See the *Write GPIO[19:09] Control* (Section 2.3.3.36) and *Read GPIO[19:00] Inputs* (Section 2.3.3.41) command for further information.

Note 2: When one or more of GPIO(19:09) are defined as input signals (using the *Write GPIO[19:09] Control* command), and/or one or more of GPIO(07:00) are defined as input signals by the OEM in Composer, this command is used to read back the current value of those specific GPIO. Each time a read request is made the ARM software will de-bounce, sample, and return the current value on these GPIO. This command will return zero for any GPIO(19:09) and GPIO(07:00) not defined as an input signal.

2.3.3.40 Write Splash Screen Execute (35h)

2.3.3.40.1 Write

This command is used to start the process of retrieving a splash screen from Flash for display on the display module.

2.3.3.40.2 Write Parameters

This command has no command parameters.

Note 1: This command is used in conjunction with the *Write Input Source Select* (Section 2.3.3.2) and the *Write Splash Screen Select* (Section 2.3.3.10) commands. It is used to start the process of retrieving a splash screen from Flash for display.

Note 2: The Splash Screen is a unique source since it is read from Flash and sent down the processing path of the ASIC one time, to be stored in memory for display at the end of the processing path. As such, *all image processing settings* (for example, image crop, image orientation, display size, splash screen select, splash screen as input source, and so forth.) *should be set appropriately by the user before executing this command.* Any data path processing changed after the splash screen has been executed will require this command to be re-executed before the result will be seen on the display. Thus, the splash screen retrieval process will be repeated each time this command is received. See also the *Write Image Freeze* command (Section 2.3.3.21) for information on hiding on-screen artifacts when selecting and retrieving a splash image.

Note 3: *It is important that the user review the notes for the Write Input Source Select command in Section 2.3.3.2 to understand the concept of source associated commands. This concept will determine when source associated commands are executed by the system. Note that this command is a source associated command; however, this command is special in that there is no maintained state or history. Thus, this command has no settings to be stored or reused by the system.*

Note 4: When this command is processed, the system will automatically set up the system color processing based on the splash header information *prior* to sending the splash image down the data path.

2.3.3.41 Read GPIO[19:00] Inputs (36h)

2.3.3.41.1 Read

This command is used to read the input values for GPIO(19:09) and GPIO(07:00) of the display module.

2.3.3.41.2 Read Parameters

This command has no command parameters.

2.3.3.41.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description	
Byte 1	GPIO Value	(See Below)
Byte 2	GPIO Value	(See Below)
Byte 3	GPIO Value	(See Below)

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(7)
b(6) – GPIO(6)
b(5) – GPIO(5)
b(4) – GPIO(4)
b(3) – GPIO(3)
b(2) – GPIO(2)
b(1) – GPIO(1)
b(0) – GPIO(0)

msb	***Byte 2***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – GPIO(16)
b(6) – GPIO(15)
b(5) – GPIO(14)
b(4) – GPIO(13)
b(3) – GPIO(12)
b(2) – GPIO(11)
b(1) – GPIO(10)
b(0) – GPIO(9)

msb	***Byte 3***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*
b(2) – GPIO(19)
b(1) – GPIO(18)
b(0) – GPIO(17)

Note 1: GPIO(19:09) can function as defined by the OEM in Composer, or their function can be redefined on-the-fly using the *Write GPIO[19:09] Control* command (Section 2.3.3.36).

GPIO(08) is not re-definable on-the-fly, and is not available for use as an OEM GPIO.

GPIO(07:00) can only function as defined by the OEM in Composer. One of the choices in Composer is to define one or more of these GPIO to be OEM GPIO Inputs

Note 2: When one or more of GPIO(19:09) are defined as input signals (using the *Write GPIO[19:09] Control* command), and/or one or more of GPIO(07:00) are defined as input signals by the OEM in Composer, this command is used to read back the current value of those specific GPIO. Each time a read request is made the ARM software will de-bounce, sample, and return the current value on these GPIO. This command will return zero for any GPIO(19:09) and GPIO(07:00) not defined as an input signal.

2.3.3.42 Write External Parallel I/F Data Mask Control (37h)

2.3.3.42.1 Write

This command is used to control the masking function for the external parallel port I/F of the display module.

2.3.3.42.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1) – Polarity Select for Data Mask Control
0h: Unmasked = 0, Masked = 1
1h: Unmasked = 1, Masked = 0

b(0) – Data Mask Enable
0h: Mask Disable
1h: Mask Enabled

Default: 00h

Note 1: When this function is enabled, the DLPC3439 input pin PDM_CVS_TE functions as a Data Mask control for the image data on the Parallel Port interface. When this function is enabled and the mask control is active, input image frames will be ignored and the source image will not be propagated to the display. During image frames that are masked, the last unmasked image frame received will continue to be displayed. The mask control signal (PDM_CVS_TE) should only be updated during vertical blanking.

Note 2: The Polarity Select specifies the active state for the mask control signal. The polarity should only be updated when the mask function is disabled (via this command).

2.3.3.43 Read External Parallel I/F Data Mask Control (38h)

2.3.3.43.1 Read

This command is used to used to read the state of the masking function for the external parallel port I/F of the display module.

2.3.3.43.2 Read Parameters

This command has no command parameters.

2.3.3.43.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1) – Polarity Select for Data Mask Control
0h: Unmasked = 0, Masked = 1
1h: Unmasked = 1, Masked = 0

b(0) – Data Mask Enable
0h: Mask Disabled
1h: Mask Enabled

2.3.3.44 Write LED Output Control Method (50h)

2.3.3.44.1 Write

This command is used to specify the method for controlling the LED outputs for the display module.

2.3.3.44.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1:0) – LED Control Method

00h: Manual RGB LED Currents (disables CAIC algorithm)

01h: CAIC (automatic) RGB LED Power (enables CAIC algorithm)

10h: *Reserved*

11h: *Reserved*

Default: Composer specified

Note 1: This command selects the method to be used to control the output of the red, green, and blue LEDs. Based on the method chosen, a specific set of commands are available for controlling the LED outputs. These are shown in Table 2-16.

Note 2: The Manual RGB LED Currents method provides for manual control of the LED currents, and as such, the CAIC algorithm (Section 2.3.3.57) will be disabled.

Note 3: The CAIC (Automatic) RGB LED Current Control method provides automatic control of the LED currents using the CAIC algorithm.

Table 2-16. Available Commands Based on LED Control Method

Available Commands	Reference Section
Manual RGB LED Current Control (CAIC Disabled)	
Write RGB LED Enable (52h)	Section 2.3.3.46
Read RGB LED Enable (53h)	Section 2.3.3.47
Write RGB LED Current (54h)	Section 2.3.3.46
Read RGB LED Current (55h)	Section 2.3.3.49
Write RGB LED Max Current (5Ch)	Section 2.3.3.51
Read RGB LED Max Current (5Dh)	Section 2.3.3.52
CAIC Automatic RGB LED Current Control (CAIC Enabled)	
Write RGB LED Enable (52h)	Section 2.3.3.46
Read RGB LED Enable (53h)	Section 2.3.3.47
Write RGB LED Current (54h)	Section 2.3.3.48
Read RGB LED Current (55h)	Section 2.3.3.49
Read CAIC LED Max Available Power (57h)	Section 2.3.3.50
Read CAIC RGB LED Current (5Fh)	Section 2.3.3.54

2.3.3.45 Read LED Output Control Method (51h)

2.3.3.45.1 Read

This command is used to read the state of the LED output control method for the display module.

2.3.3.45.2 Read Parameters

This command has no command parameters.

2.3.3.45.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1:0) – LED Control Method
00h: Manual RGB LED Currents (CAIC algorithm disabled)
01h: CAIC (automatic) RGB LED PowerCurrent Control (CAIC algorithm enabled)
10h: *Reserved*
11h: *Reserved*

2.3.3.46 Write RGB LED Enable (52h)

2.3.3.46.1 Write

This command is used to enable the LEDs for the display module.

2.3.3.46.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – Blue LED Enable
0h: Blue LED Disabled
1h: Blue LED Enabled

b(1) – Green LED Enable
0h: Green LED Disabled
1h: Green LED Enabled

b(0) – Red LED Enable
0h: Red LED Disabled
1h: Red LED Enabled

Default: 07h

2.3.3.47 Read RGB LED Enable (53h)

2.3.3.47.1 Read

This command is used to read the state of the LED enables for the display module.

2.3.3.47.2 Read Parameters

This command has no command parameters.

2.3.3.47.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – Blue LED Enable
0h: Blue LED Disabled
1h: Blue LED Enabled

b(1) – Green LED Enable
0h: Green LED Disabled
1h: Green LED Enabled

b(0) – Red LED Enable
0h: Red LED Disabled
1h: Red LED Enabled

2.3.3.48 Write RGB LED Current (54h)

2.3.3.48.1 Write

This command is used to set the current for the red, green, and blue LEDs of the display module.

2.3.3.48.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description	
Byte 1	Red LED current parameter	(LSByte)
Byte 2	Red LED current parameter	(MSByte)
Byte 3	Green LED current parameter	(LSByte)
Byte 4	Green LED current parameter	(MSByte)
Byte 5	Blue LED current parameter	(LSByte)
Byte 6	Blue LED current parameter	(MSByte)

Default: Composer specified

Note 1: When an all white image is being displayed, this command allows the system white point to be adjusted while also establishing the total LED power. This is true whether the CAIC algorithm is enabled or disabled.

Note 2: The parameters specified by this command will have a resolution of 10 bits, and are to be as defined by the appropriate PAD specification.

Note 3: When the CAIC algorithm is disabled, this command will directly set the LED currents (that is, the R, G, and B values provided will be sent directly to the PAD device) regardless of the image being displayed.

Note 4: When CAIC algorithm is enabled:

- This command will directly set the LED currents when an all-white image is displayed. If the image is changed from an all-white image, depending on the image, the CAIC algorithm may alter one or more of the LED currents from those specified by this command and the total LED power may also drop. Command *Read CAIC RGB LED Current (5Fh)* can be used to read the actual LED currents for the image currently being displayed.
- In the case of an all-white image, the values read by command *Read CAIC RGB LED Current (5Fh)* will closely match but may not exactly match those requested using command *Write RGB LED Current (54h)* . For an all-white image command *Read CAIC RGB LED Current (5Fh)* will give currents within ±4 PAD device current steps for each LED color relative to those requested by command *Write RGB LED Current (54h).*
- When command *Write RGB LED Current (54h)* is used to change the LED currents, the LED current for any color should not be changed by more than ±25% from the nominal current used for that color when the CAIC LUTs were created. Furthermore, no LED current should be set to a current value beyond the maximum value supported in the CAIC Intensity-to-Current LUT for the corresponding color.
- The maximum total LED power for any displayed image occurs for an all-white image since in this case the CAIC algorithm will request the CAIC LED max available power. The max available LED power for CAIC is controlled by command *Write RGB LED Current* since this command controls currents for an all-white image. After the currents are adjusted, command *Read CAIC LED Max Available Power (57h)* can be used to see the max power in Watts that CAIC derived.

2.3.3.49 Read RGB LED Current (55h)

2.3.3.49.1 Read

This command is used to read the state of the current for the red, green, and blue LEDs of the display module.

2.3.3.49.2 Read Parameters

This command has no command parameters.

2.3.3.49.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***	
Byte 1	Red LED current parameter	(LSByte)
Byte 2	Red LED current parameter	(MSByte)
Byte 3	Green LED current parameter	(LSByte)
Byte 4	Green LED current parameter	(MSByte)
Byte 5	Blue LED current parameter	(LSByte)
Byte 6	Blue LED current parameter	(MSByte)

Note 1: See the Write RGB LED Current Control command 54h for a detailed description of the return parameters.

Note 2: Unused, most significant bits will be set to *0*.

2.3.3.50 Read CAIC LED Max Available Power (57h)

2.3.3.50.1 Read

This command is used to read the maximum LED power allowed for the display module at the LED current settings set by the *Write RGB LED Current (54h)* command.

2.3.3.50.2 Read Parameters

This command has no command parameters.

2.3.3.50.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	Maximum LED Power (LSByte)
Byte 2	Maximum LED Power (MSByte)

Note 1: The value will be specified in Watts × 100
Example: 25.75 W = A0Fh

Note 2: This command is only applicable when CAIC is enabled.

Note 3: The maximum available LED power associated with the CAIC algorithm is specific to an all white displayed image where the LED currents are set by the *Write RGB LED Current* command (Section 2.3.3.48).

The calculation is:

Max Avail Pwr = (Rdc × Rledc × Rledv) + (Gdc × Gledc × Gledv) + (Bdc × Bledc × Bledv)

1. Rdc = Red Duty Cycle; Rledc = Red LED Current; Rledv = Red LED Voltage
2. Gdc = Green Duty Cycle; Gledc = Green LED Current; Gledv = Green LED Voltage
3. Bdc = Blue Duty Cycle; Bledc = Blue LED Current; Bledv = Blue LED Voltage

Example: (.30 × .49 A × 2.0 V) + (.50 × .39 A × 3.1 V) + (.20 × .39 A × 3.1 V) = 1.140 W

2.3.3.51 Write RGB LED Max Current (5Ch)

2.3.3.51.1 Write

This command is used to specify the maximum LED current allowed for each LED in the display module when CAIC is disabled.

2.3.3.51.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Maximum Red LED Current (LSByte)
Byte 2	Maximum Red LED Current (MSByte)
Byte 3	Maximum Green LED Current (LSByte)
Byte 4	Maximum Green LED Current (MSByte)
Byte 5	Maximum Blue LED Current (LSByte)
Byte 6	Maximum Blue LED Current (MSByte)

Default: Composer specified

Note 1: The parameters specified by this command will have a resolution of 10 bits, and are to be as defined by the appropriate PAD specification.

Note 2: This command sets the maximum LED currents that can be used when the CAIC algorithm is disabled. When the CAIC algorithm is enabled, the maximum LED currents are determined by the CAIC algorithm LUTs stored in Flash.

Note 3: For further information about LED current and the CAIC algorithm, see the notes for the *Write RGB LED Current* (54h) command.

Note 4: Unused, most significant bits should be set to *0*.

2.3.3.52 Read RGB LED Max Current (5Dh)

2.3.3.52.1 Read

This command is used to read the specified maximum LED current allowed for each LED in the display module.

2.3.3.52.2 Read Parameters

This command has no command parameters.

2.3.3.52.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*	
Byte 1	Maximum Red LED Current	(LSByte)
Byte 2	Maximum Red LED Current	(MSByte)
Byte 3	Maximum Green LED Current	(LSByte)
Byte 4	Maximum Green LED Current	(MSByte)
Byte 5	Maximum Blue LED Current	(LSByte)
Byte 6	Maximum Blue LED Current	(MSByte)

Note 1: See the Write RGB LED Current Control command for a detailed description of the return parameters.

Note 2: Unused, most significant bits will be set to *0*.

2.3.3.53 Read Measured LED Parameters (5Eh)

2.3.3.53.1 Read

This command is used to read the measured values for a number of LED-based parameters in the display module.

2.3.3.53.2 Read Parameters

This command has no command parameters.

2.3.3.53.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***	
Byte 1	Measured Red LED Current	(LSByte)
Byte 2	Measured Red LED Current	(MSByte)
Byte 3	Measured Green LED Current	(LSByte)
Byte 4	Measured Green LED Current	(MSByte)
Byte 5	Measured Blue LED Current	(LSByte)
Byte 6	Measured Blue LED Current	(MSByte)
Byte 7	Measured Red LED Voltage	(LSByte)
Byte 8	Measured Red LED Voltage	(MSByte)
Byte 9	Measured Green LED Voltage	(LSByte)
Byte 10	Measured Green LED Voltage	(MSByte)
Byte 11	Measured Blue LED Voltage	(LSByte)
Byte 12	Measured Blue LED Voltage	(MSByte)
Byte 13	Measured Red LED Power	(LSByte)
Byte 14	Measured Red LED Power	(MSByte)
Byte 15	Measured Green LED Power	(LSByte)
Byte 16	Measured Green LED Power	(MSByte)
Byte 17	Measured Blue LED Power	(LSByte)
Byte 18	Measured Blue LED Power	(MSByte)
Byte 19	Total LED Power	(LSByte)
Byte 20	Total LED Power	(MSByte)

Note 1: Current will be specified as Milliamps × 2, with the maximum value = 32,767.5 mA
Example: 1287.5 mA = 0A0Fh

Note 2: Voltage will be specified as Voltage × 1700, with the maximum value = 38.550 V
Example: 1.548 mA = 0A48h

Note 3: Power will be specified as Watts × 325, with the maximum value = 201.64 W
Example: 7.923 W = A0Fh

2.3.3.54 Read CAIC RGB LED Current (5Fh)

2.3.3.54.1 Read

This command is used to read the state of the current for the red, green, and blue LEDs of the display module.

2.3.3.54.2 Read Parameters

This command has no command parameters.

2.3.3.54.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***	
Byte 1	Red LED current parameter	(LSByte)
Byte 2	Red LED current parameter	(MSByte)
Byte 3	Green LED current parameter	(LSByte)
Byte 4	Green LED current parameter	(MSByte)
Byte 5	Blue LED current parameter	(LSByte)
Byte 6	Blue LED current parameter	(MSByte)

Note 1: The parameters returned by this command will have a resolution of 10 bits, and will be as defined by the appropriate PAD specification.

Note 2: When the CAIC algorithm is enabled using the *LED Output Control Method* command:

- The *Write RGB LED Current* command will directly set the LED currents when an all white image is being displayed. If the image is changed from an all white image, depending on the image, the CAIC algorithm may alter one or more of the LED currents from those specified the *Write RGB LED current* command and the total LED power may also drop. The actual LED currents for the image currently being displayed can be read using this command (the *Read CAIC RGB LED Current* (5Fh) command).
- In the case of an all white image, the values returned by this command will closely match, but may not exactly match, those specified using the *Write RGB LED Current* command. For an all white image, this command will provide values within +/- 4 PAD device current steps for each LED color relative to those specified with the *Write RGB LED Current* command.

Note 3: Use of this command is only appropriate when the *LED Output Control Method* is set to *CAIC (Automatic) RGB LED Current Control*.

Note 4: Unused, most significant bits will be set to *0*.

2.3.3.55 Write Local Area Brightness Boost Control (80h)

2.3.3.55.1 Read

This command is used to control the local area brightness boost image processing functionality for the display module.

2.3.3.55.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	See Below
Byte 2	LABB Strength setting

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4) – Sharpness Strength

b(3:2) – *Reserved*

b(1:0) – LABB Control

0h: Disabled

1h: Enabled: Manual Strength Control (no light sensor)

2h: Enabled: Automatic Strength Control (uses light sensor)

3h: *Reserved*

Default: 0001h

Note 1: The key function of the LABB is to adaptively gain up darker parts of the image to achieve an overall brighter image.

Note 2: For automatic strength control, a light sensor will be used to automatically adjust the applied image strength based on the measured black level of the screen, or the ambient lighting level of the room.

Note 3: For LABB Strength, 0 indicates no boost applied, and 255 indicates the maximum boost that is considered viable in a product. The strength is not a direct indication of the gain since the gain will vary depending on image content.

Note 4: Sharpness strength can range from 0 to 15, with 0 indicating sharpness disabled, and 15 indicating the maximum sharpness. The LABB function must be enabled (either Manual or Automatic) to make use of Sharpness.

Note 5: LABB is supported in TPG, Splash, External Input mode, but auto-disabled in curtain mode.

2.3.3.56 Read Local Area Brightness Boost Control (81h)

2.3.3.56.1 Read

This command is used to read the state of the local area brightness boost image processing functionality for the display module.

2.3.3.56.2 Read Parameters

This command has no command parameters.

2.3.3.56.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below
Byte 2	LABB Strength setting
Byte 3	LABB Gain Value

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4) – Sharpness Strength

b(3:2) – *Reserved*

b(1:0) – LABB Control

0h: Disabled

1h: Enabled: Manual Strength Control (no light sensor)

2h: Enabled: Automatic Strength Control (uses light sensor)

3h: *Reserved*

Note 1: Shows the bit order and weighting for the LABB Gain value, which can range from 1 to almost 8 (ASIC software should limit the lower value to 1).

Table 2-17. Bit Weight Definition for LABB Gain Value

b7	b6	b5	b4	b3	b2	b1	b0
2^{2}	2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}

Note 2: The software equation to calculate LABB Gain as a fixed point value is shown below:

LABB_gain = add_8lsb(APL) / pre_LABB_APL (//add 8 LSBs (u8.0 / u8.0 = u8.8 / u8.0 = u8.8)

2.3.3.57 Write CAIC Image Processing Control (84h)

2.3.3.57.1 Write

This command is used to control the CAIC functionality for the display module.

2.3.3.57.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below
Byte 2	CAIC Maximum Lumens Gain
Byte 3	CAIC Clipping Threshold

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – CAIC Gain Display Enable
0h: Disabled
1h: Enabled

b(6) – CAIC Gain Display Scale
0h: 100% = 1024 Pixels
1h: 100% = 512 Pixels

b(5:3) – *Reserved*

b(2:0) – CAIC WPC Control
0h: White Point Correction Disabled
1h: White Point Correction Enabled
else: *Reserved*

Default: Composer specified

Note 1: The CAIC algorithm (Content Adaptive Illumination Control) provides adaptive control of the LED currents and the digital gain applied to the image. In addition, when an external sensor is provided by the OEM (and when WPC is enabled by this command), the algorithm will provide automatic white point correction.

Note 2: The CAIC algorithm is enabled or disabled based on the method of LED current control selected by the OEM using the Write LED Output Control Method (Section 2.3.3.44) command. When enabled, the CAIC algorithm provides automatic control of the LED currents as specified by this command and the Write LED Output Control Method command.

Note 3: The CAIC Gain Display provides a visual presentation of the instantaneous gain provided by the CAIC algorithm. This is typically used as a debug tool and to show the performance of the algorithm. It should never be used for normal operation. The display is made up of 5 bars, where the bottom three bars (green, red, and blue) show the respective CAIC gain for each color. The top two bars are for TI debug use only. For SW, the CAIC Gain Display Enable is controlled by CAIC_DEBUG_MODE (2:0), where Disabled = 0h, and Enabled = 3h. The Display Scale is set using CAIC_DEBUG_MODE(3).

Note 4: Table 2-18 shows the bit order and weighting for the CAIC Maximum Lumens Gain value, which has a valid range from 1.0 to 4.0. Values outside of this range will be considered an error (invalid command parameter value – communication status) and the command will not be executed.

Table 2-18. Bit Weight Definition for the CAIC Maximum Gain Value

b7	b6	b5	b4	b3	b2	b1	b0
2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}

Note 5: The CAIC Maximum Lumens Gain parameter sets the maximum lumens gain that a pixel can have as a result of both digital gain and increasing LED currents. It also serves to bias the CAIC algorithm towards either Constant Power (variable brightness) or Constant Lumens (variable power). Some examples are listed below:

- Maximum Gain value = 1.0: This biases performance to Constant Lumens. In this case, LED power is reduced for those images where this is possible, but lumens do not increase or decrease.
- Maximum Lumens Gain value = 4.0: This biases performance to Constant Power. In this case, power is held constant for most images, while the lumens are gained up. It should be noted that for the small percent of images where the gain would exceed 4.0, lumens will stop increasing and the power is reduced instead.

Note 6: Table 2-19 shows the bit order and weighting for the CAIC Clipping Threshold value, which has a valid range from 0.0% to 2.0%. Values outside of this range will be considered an error (invalid command parameter value – communication status) and the command will not be executed.

Table 2-19. Bit Weight Definition for the CAIC Clipping Threshold Value

b7	b6	b5	b4	b3	b2	b1	b0
2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}

Note 7: The CAIC Clipping Threshold parameter sets the percentage of pixels that can be clipped by the CAIC algorithm over the full frame of active data due to the digital gain being applied by the CAIC algorithm.

Note 8: Table 2-20 shows the bit order and weighting for the CAIC RGB Intensity Gain values, which have a valid range from 0.0 to almost 1.0. Values outside of this range will be considered an error (invalid command parameter value – communication status) and the command will not be executed.

Table 2-20. Bit Weight Definition for the CAIC RGB Intensity Gain Values

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	Res	Res	Res	Res	Res	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}

Note 9: CAIC can be enabled in TPG and External Input mode, but auto-disabled in Splash and Curtain mode.

Feature	TPG	Splash	Curtain	External Input
LABB	Supported	Supported	Auto-Disabled	Supported
CAIC	Supported	Auto-Disabled	Auto-Disabled	Supported

2.3.3.58 Read CAIC Image Processing Control (85h)

2.3.3.58.1 Read

This command is used to read the state of the CAIC functionality within the display module.

2.3.3.58.2 Read Parameters

This command has no command parameters.

2.3.3.58.3 Return Parameters

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – CAIC Gain Display Enable
0h: Disabled
1h: Enabled

b(6) – CAIC Gain Display Scale
0h: 100% = 1024 Pixels
1h: 100% = 512 Pixels

b(5:3) – *Reserved*

b(2:0) – CAIC WPC Control
0h: White Point Correction Disabled
1h: White Point Correction Enabled
else: *Reserved*

Note 1: Information on these parameters can be found in the notes for the Write CAIC Image Processing Control command (Section 2.3.3.57).

2.3.3.59 Write Color Coordinate Adjustment Control (86h)

2.3.3.59.1 Write

This command is used to control the CCA image processing functionality for the display module.

2.3.3.59.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – CCA Enable
0h: Disabled
1h: Enabled

Default: 01h

Note 1: This command is for TI debug purposes only. This function should remain enabled during normal operation.

Note 2: When CCA is disabled, and identity matrix should be used.

2.3.3.60 Read Color Coordinate Adjustment Control (87h)

2.3.3.60.1 Read

This command is used to read the state of the CCA image processing within the display module.

2.3.3.60.2 Read Parameters

This command has no command parameters.

2.3.3.60.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – CCA Enable
0h: Disabled
1h: Enabled

2.3.3.61 Write Border Color (B2h)

2.3.3.61.1 Write

This command is used to specify the on screen border color for the display module.

2.3.3.61.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2:0) – Display Border Color
- 0h: Black
- 1h: Red
- 2h: Green
- 3h: Blue
- 4h: Cyan
- 5h: Magenta
- 6h: Yellow
- 7h: White

Default: 00h

Note 1: Whenever the display image size is smaller than the active area of the DMD, this border color will be used for all non image pixels. Some examples where a border might come into play would be for a Window Box, Pillar Box, or Letterbox image.

Note 2: For the special case of displaying a pillar box image (Figure 2-13), the OEM can make use of the border color defined by this command, or make use of a dithered 24-bit border color. The definition of this dithered 24-bit border color, as well as the decision whether to use it, or use the color selected by this command is made with the DLP Composer software tool and stored in flash.

Note 3: The border color specified by this command is separate from the curtain color defined in the Display Image Curtain command (Section 2.3.3.19), even though they are both displayed using the curtain capability.

Note 4: The dithered 24-bit border color is specified in the VGP/CCP.

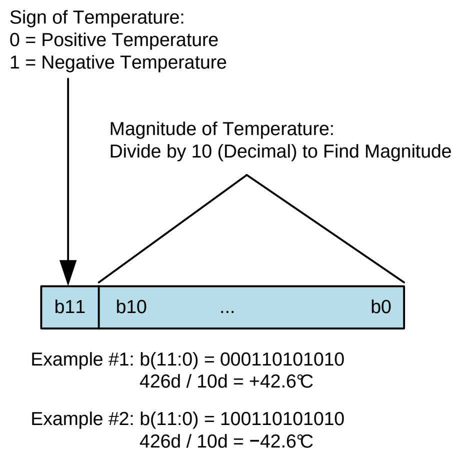

Figure 2-13. Pillar-Box Border Example

2.3.3.62 Read Border Color (B3h)

2.3.3.62.1 Read

This command is used to read the state of the on screen border color for the display module.

2.3.3.62.2 Read Parameters

This command has no command parameters.

2.3.3.62.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – Pillar-Box Border Color Source
0h: Defined by this command
1h: Flash defined 24-bit color

b(6:3) – *Reserved*

Display Border Color
0h: Black
1h: Red
2h: Green
3h: Blue
4h: Cyan
5h: Magenta
6h: Yellow
7h: White

Note 1: For the special case of a pillar box image (Figure 2-13), the OEM can make use of the border color defined by the Write Border Color command (Section 2.3.3.61), or make use of a dithered 24-bit border color. The definition of this dithered 24-bit border color, as well as the decision whether to use it, or use the color selected by this command is made with the DLP Composer™ software tool and stored in flash. The use decision stored in flash is shown by bit-7 of this command.

2.3.3.63 Write External Parallel I/F SYNC Polarity (B6h)

2.3.3.63.1 Write

This command is used to specify the SYNC polarity for the external parallel interface of the display module.

2.3.3.63.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – Manual Mode – Parallel Port HSYNC Polarity
0h: Falling Edge Active (Negative Pulse)
1h: Rising Edge Active (Positive Pulse)

b(1) – Manual Mode – Parallel Port VSYNC Polarity
0h: Falling Edge Active (Negative Pulse)
1h: Rising Edge Active (Positive Pulse)

b(0) – Parallel Port Sync Polarity Mode
0h: Automatic Mode
1h: Manual Mode

Default: 00h

Note 1: This command is required whenever the source makes use of the Parallel port input, except for BT656 sources. This command is not applicable for BT656 sources. In Automatic mode, the system can typically determine the appropriate polarity of the syncs. In Manual mode, the OEM is allowed to specify these polarities should the need arise.

2.3.3.64 Read External Parallel I/F SYNC Polarity (B7h)

2.3.3.64.1 Read

This command is used to read the state of the SYNC polarity for the external parallel interface of the display module.

2.3.3.64.2 Read Parameters

This command has no command parameters.

2.3.3.64.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1) – Parallel Port HSYNC Polarity
0h: Falling Edge Active (Negative Pulse)
1h: Rising Edge Active (Positive Pulse)

b(0) – Parallel Port VSYNC Polarity
0h: Falling Edge Active (Negative Pulse)
1h: Rising Edge Active (Positive Pulse)

2.3.3.65 Write External Parallel I/F Video Manual Image Framing (B8h)

2.3.3.65.1 Write

This command is used to specify the external parallel interface video manual image framing parameters for the display module.

2.3.3.65.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*	
Byte 1	See Below	
Byte 2	Start Pixel	(LSByte)
Byte 3	Start Pixel	(MSByte)
Byte 4	Start Line	(MSByte)
Byte 5	Start Line	(MSByte)

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – External Video Manual Image Framing Enable

0h: Disabled

1h: Enable

Default: All bytes: 00h

Note 1: This command is only required when the source data for the parallel interface does not provide an active data valid framing signal (vertical and horizontal syncs are still required). These are referenced to the appropriate sync signal (for example, start pixel referenced to each horizontal sync). This command is used in conjunction with the Write Video Input Image Size command (Section 2.3.3.33).

Note 2: The user must enable or disable manual framing as appropriate. If manual framing is specified, it will be used even if an active data valid framing signal is provided with the input. (In other words, manual framing will override the active data valid signal.)

Note 3: The parameter values are to be *1* based. (That is, a value of 1 will specify the first pixel of a line, of the first line of the frame.)

Note 4: This function is NOT applicable to BT656 sources. Framing for these sources will be handled automatically by the system.

2.3.3.66 Read External Parallel I/F Video Manual Image Framing (B9h)

2.3.3.66.1 Read

This command is used to read the state of the external parallel interface video manual image framing parameters for the display module

2.3.3.66.2 Read Parameters

This command has no command parameters.

2.3.3.66.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*	
Byte 1	See Below	
Byte 2	Start Pixel	(LSByte)
Byte 3	Start Pixel	(MSByte)
Byte 4	Start Line	(MSByte)
Byte 5	Start Line	(MSByte)

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – External Video Manual Image Framing Enable
0h: Disabled
1h: Enable

Note 1: The parameter values are to be *1* based. (That is, a value of 1 will specify the first pixel of a line, of the first line of the frame.)

2.3.3.67 Read Auto Framing Information (BAh)

2.3.3.67.1 Read

This command is used to read the external input framing information that is automatically determined by the display module

2.3.3.67.2 Read Parameters

This command has no command parameters.

2.3.3.67.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***	
Byte 1	External Input VSYNC Rate	(LSByte)
Byte 2	External Input VSYNC Rate	
Byte 3	External Input VSYNC Rate	
Byte 4	External Input VSYNC Rate	(MSByte)
Byte 5	External Input Total Pixels per Line	(LSByte)
Byte 6	External Input Total Pixels per Line	(MSByte)
Byte 7	External Input Total Lines per Frame	(LSByte)
Byte 8	External Input Total Lines per Frame	(MSByte)
Byte 9	External Input Active Pixels per Line	(LSByte)
Byte 10	External Input Active Pixels per Line	(MSByte)
Byte 11	External Input Active Lines per Frame	(LSByte)
Byte 12	External Input Active Lines per Frame	(MSByte)
Byte 13	Pixel/Line Reference Clock Rate	(LSByte)
Byte 14	Pixel/Line Reference Clock Rate	(MSByte)

Note 1: In most cases, the above data can be measured by the system (even when manual data framing is used). *This data is provided for debug purposes only.*

Note 2: The external input frame rate is returned as a count that is specified in units of 66.67ns (based on the internal 15MHz clock used to time between input frame syncs).

Note 3: The pixels per line and lines per frame parameters are to be 1 based. (That is, a value of 1280 active pixels indicates that there are 1280 active pixels per line.)

Note 4: The pixels per line and lines per frame parameters are based on measurement of the actual input pixel clock for the Parallel Bus. This clock rate is returned as the *Pixel/Line Reference Clock Rate*. This parameter value is the clock rate times 100 in MHz (for example, 60.00MHz = 1770h).

2.3.3.68 Read Short Status (D0h)

2.3.3.68.1 Read

This command is used to provide a short system status for the display module.

2.3.3.68.2 Read Parameters

This command has no command parameters.

2.3.3.68.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below

msb	***Byte 1*** **- General Status**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – Boot/Main Application
0h: Boot
1h: Main

b(6) – *Reserved*

b(5) – Flash Error
0h: No Error
1h: Error

b(4) – Flash Erase Complete
0h: Complete
1h: Not Complete

b(3) – System Error
0h: No Error
1h: Error

b(2) – *Reserved*

b(1) – Communication Error
0h: No Error
1h: Error

b(0) – System Initialization
0h: Not Complete
1h: Complete

Note 1: The *Flash Erase Complete* status bit will be set at the start of the Flash erase process, and will be cleared when the erase process is complete. The flash status can be obtained during or after the erase process. To obtain this status during the erase process, only this command can be sent after the start of the flash erase. If any other command is sent during the erase process, it will be held without processing until the flash erase has completed (thus blocking any following status requests until the previously sent command is processed).

Note 2: The *Flash Error* bit is used to indicate an error during any Flash operation. For Flash writes, this bit will be updated at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until cleared. This will allow the OEM the option of checking the status between each write transaction, or at the end of the update. Once a write transaction has started, the flash status (and this error bit) will not be accessible until the write transaction has completed.

Note 3: The *Communication Error* bit is used to indicate any error on the I^2C command interfaces. Specific details about communication errors are available using the *Read Communication Status* command (Section 2.3.3.71).

Note 4: Any errors other than *Flash Error* and *Communication Error* are indicated by the *System Error* bit. Specific details about system errors are available using the *Read System Status* command (Section 2.3.3.69).

Note 5: The *Flash Error*, *Communication Error*, and *System Error* bits will be cleared when the *Read Short Status* is read.

Note 6: The *Read Short Status* command should only be checked *periodically*, not *continuously*. It is likely that continuous access will severely impact system performance.

2.3.3.69 Read System Status (D1h)

2.3.3.69.1 Read

This command is used to read system status information for the display module.

2.3.3.69.2 Read Parameters

This command has no command parameters.

2.3.3.69.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	DMD Interface Status
Byte 2	LED Status
Byte 3	Internal Interrupt Status
Byte 4	Misc. Status

Note 1: All system status error bits will be cleared when the *Read System Status* is read.

msb	***Byte 1* – DMD Interface Status**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – DMD Training Error
0h: No Error
1h: Error

b(1) – DMD Interface Error
0h: No Error
1h: Error

b(0) – DMD Device Error
0h: No Error
1h: Error

Note 1: The system will set the DMD Device Error for the following conditions:
- The system cannot read the DMD Device ID from the DMD
- The Composer specified DMD Device ID does not match the actual DMD Device ID

Note 2: The system will set the DMD Interface Error when there are power management setup conflicts on this interface.

Note 3: The system will set the DMD Training Error when the training algorithm can't find a data eye that will meet the specified requirements.

msb	***Byte 2* – LED Status**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*

b(5) – Blue LED Error
0h: No Error
1h: Error

b(4) – Green LED Error
0h: No Error
1h: Error

b(3) – Red LED Error
0h: No Error
1h: Error

b(2) – Blue LED State
0h: Off
1h: On

b(1) – Green LED State
0h: Off
1h: On

b(0) – Red LED State
0h: Off
1h: On

msb	***Byte 3* – Internal Interrupt Status**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *TBD*

b(1) – Sequence Error
0h: No Error
1h: Error

b(0) – Sequence Abort Error
0h: No Error
1h: Error

msb	***Byte 4* – Misc. Status**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*

b(5) – Watchdog Timer Timeout
0h: No Timeout
1h: Timeout

b(4) – Product Configuration Error
0h: No Error
1h: Error

b(3) – Master vs. Slave Operation

0h: Master
1h: Slave

b(2) – Single vs. Dual ASIC Configuration
0h: Single
1h: Dual

b(1) – SPI Flashless Communication Error
0h: No Error
1h: Error

b(0) – SPI Flashless Data Request Error
0h: No Error
1h: Error

Note 1: The system will set the SPI Flashless Data Request Error bit if the display does not start sending the requested data before the SPI flashless data request timeout is exceeded. Once the timeout is exceeded, the display will abort the current request, and then try again.

Note 2: The system will set the SPI Flashless Communication Error bit if the display has three consecutive SPI Flashless Data Request Errors. If this happens, it is assume that the SPI communication link is not operational, and system operations will halt. A reset will be required to restart operations.

Note 3: The system will set the Master vs. Slave bit as appropriate in both single and dual ASIC configurations.

Note 4: The system will set the Product Configuration Error bit if it determines that some piece of the product configuration is not correct. Some examples are:

- Invalid ASIC/DMD Combination
- Invalid ASIC/DLPA300x Combination
- Invalid Flash build for current ASIC, DMD, and/or DLPA300x configuration

Note 5: The system will set the Watchdog Timer Timeout bit if the system has been reset due to a watchdog timer timeout.

2.3.3.70 Read System Software Version (D2h)

2.3.3.70.1 Read

This command is used to read the main application software version information for the display module.

2.3.3.70.2 Read Parameters

This command has no command parameters.

2.3.3.70.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*	
Byte 1	ASIC Main Application Software Version	Patch LSByte
Byte 2	ASIC Main Application Software Version	Patch MSByte
Byte 3	ASIC Main Application Software Version	Minor
Byte 4	ASIC Main Application Software Version	Major

2.3.3.71 Read Communication Status (D3h)

2.3.3.71.1 Read

This command is used to read system status information for the display module.

2.3.3.71.2 Read Parameters

Parameter Bytes	***Description***
Byte 1	Command Bus Status Selection

msb	***Byte 1* – Command Bus Status Selection**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2) – *Reserved*

b(1:0) – Command Bus Status Selection
0h: *Reserved*
1h: *Reserved*
10h: I^2C only
11h: *Reserved*

Note 1: This command will return the communication status for the command bus specified.

- Reserved: This selection will return status bytes 1 through 6
- Reserved: This selection will return status bytes 1 though 4
- I^2C only: This selection will return status bytes 5 though 6

Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	*Reserved*
Byte 2	*Reserved*
Byte 3	*Reserved*
Byte 4	*Reserved*
Byte 5	I^2C Communication Status
Byte 6	I^2C Aborted Op-Code

Note 1: All communication status error bits will be cleared when the *Read Communication Status* is read.

msb	***Byte 5* – I^2C Communication Status**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7) – *Reserved*

b(6) – Bus Timeout by Display Error
0h: No Error

1h: Error

b(5) – Invalid Number of Command Parameters
0h: No Error
1h: Error

b(4) – Read Command Error
0h: No Error
1h: Error

b(3) – Flash Batch File Error
0h: No Error
1h: Error

b(2) – Command Processing Error
0h: No Error
1h: Error

b(1) – Invalid Command Parameter Value
0h: No Error
1h: Error

b(0) – Invalid Command Error
0h: No Error
1h: Error

Note 1: The system will set the Invalid Command Error bit when it does not recognize the command op-code. The invalid command op-code will be reported in the I²C CMD Error Op-Code byte of this status.

Note 2: The system will set the Invalid Command Parameter Error bit when the it detects that the value of a command parameter is not valid (for example, out of allowed range).

Note 3: The system will set the Command Processing Error bit when a fault is detected when processing a command. In this case, the command will be aborted with the system moving on to the next command. The op-code for the aborted command will be reported in the I²C CMD Error Op-Code byte of this status.

Note 4: The system will set the Flash Batch File Error bit when an error occurs during the processing of a flash batch file. When this bit is set, typically another bit will be set to indicate what kind of error was detected (for example, Invalid Command Error).

Note 5: The system will set the Read Command Error bit when the host terminates the read operation before all of the requested data has been provided, or if the host continues to request read data after all of the requested data has been provided.

Note 6: The system will set the Invalid Number of Command Parameters Error bit when too many or too few command parameters are received. In this case, the command will be aborted with the system moving on to the next command. The op-code for the aborted command will be reported in the I²C CMD Error Op-Code byte of this status.

Note 7: The system will set the Bus Timeout by Display Error bit when the display releases control of the bus because the bus timeout value was exceeded.

msb	***Byte 6* – I²C CMD Error Op-Code**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – I²C CMD Error Op-Code

Note 1: The CMD Error Op-Code is associated with various I²C communication status bits, and reports the op-code for an I²C command as noted.

2.3.3.72 Read ASIC Device ID (D4h)

2.3.3.72.1 Read

This command is used to read the ASIC Device ID for the display module.

2.3.3.72.2 Read Parameters

This command has no command parameters.

2.3.3.72.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	Device ID

msb	*Byte 1*						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:4) – *Reserved*

b(3:0) – ASIC Device ID

Note 1: The ASIC Device ID can be decoded using Table 2-21.

Table 2-21. ASIC Device ID Decode

ASIC Device ID	Device Number	DMD Resolution	# of ASICs	Package	LED Driver
00h	DLPC3430	≤ 1280 × 720	1	7 mm × 7 mm (0.4-mm pitch)	DLPA200x
01h	DLPC3433	≤ 1280 × 720	1	7 mm × 7mm (0.4-mm pitch)	DLPA200x/ DLPA3000
04h	DLPC3435	≤ 1280 × 720	1	13 mm × 13 mm (0.8-mm pitch)	DLPA200x
05h	DLPC3438	≤ 1280 × 720	1	13 mm × 13 mm (0.8-mm pitch)	DLPA200x/ DLPA3000
09h	DLPC3439	1920 × 1080	2	13 mm × 13 mm (0.8-mm pitch)	DLPA3000/ 3005

2.3.3.73 Read DMD Device ID (D5h)

2.3.3.73.1 Read

This command is used to read the DMD Device ID for the display module.

2.3.3.73.2 Read Parameters

Parameter Bytes	*Description*
Byte 1	DMD Register Selection

msb	***Byte 1* – DMD Register Selection**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(0:0) – DMD Data Selection
0h: DMD Device ID
1h to 7h: *Reserved*

2.3.3.73.3 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See DMD Device ID Reference Table.
Byte 2	See DMD Device ID Reference Table.
Byte 3	See DMD Device ID Reference Table.
Byte 4	See DMD Device ID Reference Table.

DMD Device ID Reference Table

DMD Device ID				**Device Description**
Byte 1 (Identifier)	**Byte 2 (Byte Count)**	**Byte 3 (ID-MSByte)**	**Byte 4 (ID-LSByte)**	**(Resolution and Type)**
60h	0Dh	00h	64h	0.2 WVGA (854 × 480, Sub-LVDS)
60h	0Dh	00h	68h	0.3 720p (1280 × 720, Sub-LVDS)
60h	0Dh	00h	6Bh	0.47 1080p (1920 × 1080, Sub-LVDS)

2.3.3.74 Read System Temperature (D6h)

2.3.3.74.1 Read

This command is used to read the system temperature for the display module.

2.3.3.74.2 Read Parameters

This command has no command parameters.

2.3.3.74.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	See Below (LSByte)
Byte 2	See Below (MSByte)

Note 1: Figure 2-14 shows the bit order and definition for the signed magnitude system temperature data, which will be returned in °C. The unspecified msbits (bits 15:12) will be set to *0*.

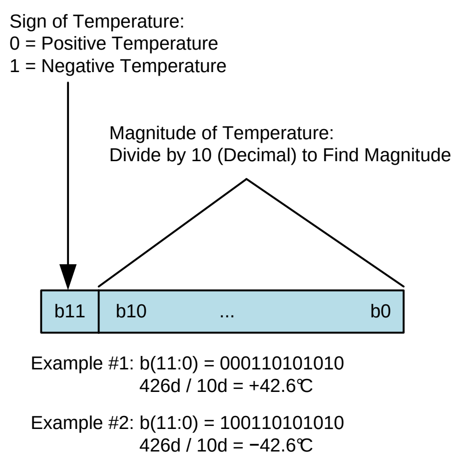

Figure 2-14. Bit Order and Definition for System Temperature

2.3.3.75 Read Flash Build Version (D9h)

2.3.3.75.1 Read

This command is used to read the ASIC flash version for the display module.

2.3.3.75.2 Read Parameters

The command has no command parameters.

2.3.3.75.3 Return Parameters

The return parameters are described below.

Parameter Bytes	Description	
Byte 1	Flash Build Version	Patch LSByte
Byte 2	Flash Build Version	Patch MSByte
Byte 3	Flash Build Version	Minor
Byte 4	Flash Build Version	Major

Note 1: The OEM is allowed to specify (via Composer) a version number for the ASIC flash build in the format specified by this command. This command allows the OEM to read back this version information.

2.3.3.76 Write Flash Batch File Delay (DBh)

2.3.3.76.1 Write

This command is used to specify an execution time delay within a Flash batch file for the display module.

2.3.3.76.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	Description	
Byte 1	Flash Batch File Delay	LSB
Byte 2	Flash Batch File Delay	MSB

Default: Composer specified

Note 1: This command is used to specify an execution delay time within a Flash batch file. It can only be used within a Flash batch file, and is not a valid command on the I^2C interfaces. (It will be flagged as a TBD command.)

Note 2: The Flash batch file delay is to be specified in units of 1 ms.
Example: 500ms = 1F4h

Note 3: Typical use of this command will be in the Auto-Init Flash batch file (batch file 0), but is valid for use in any batch file. (See *Write Execute Flash Batch File* in Section 2.3.3.32)

Note 4: Software should make use of the available hardware timers.

2.3.3.77 Read DMD I/F Training Data (DCh)

2.3.3.77.1 Read

This command is used to read back the DMD interface training data for the display module.

2.3.3.77.2 Read Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	DMD I/F Training Data Selection (See Below)

msb	*Byte 1* – DMD I/F Data Selection						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:5) – *Reserved*

b(4) – Training Data Selection
0h: High/Low/Selected
1h: Full Profile

b(3:0) – ASIC Pin Pair Selection
0h: A
1h: B
2h: C
3h: D
4h: E
5h: F
6h: G
7h: H
8h-Fh: *Reserved*

Note 1: This command will return the DMD I/F training data specified for the ASIC pin pair specified

- High/Low/Selected: This selection will return bytes 1 through 4.
- Full Profile: This selection will return bytes 5 though 11.

2.3.3.77.2.1 Return Parameters

The return parameters are described below.

Parameter Bytes	*Description*		
Byte 1	High/Low/Selected	(See Below)	(LSByte)
Byte 2	High/Low/Selected	(See Below)	
Byte 3	High/Low/Selected	(See Below)	
Byte 4	High/Low/Selected	(See Below)	(MSByte)
Byte 5	Full Profile	(Bits 7-0)	(LSByte)
Byte 6	Full Profile	(Bits 15-8)	
Byte 7	Full Profile	(Bits 23-16)	
Byte 8	Full Profile	(Bits 31-24)	
Byte 9	Full Profile	(Bits 39-32)	

Parameter Bytes	Description		
Byte 10	Full Profile	(Bits 47-40)	
Byte 11	Full Profile	(Bits 50-48)	(MSByte)

msb	Byte 1 – High/Low/Selected						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*

b(5) – Training Error
0h: No Error
1h: Error

b(4) – Pin Pair Selected for Training
0h: No
1h: Yes

b(3:0) – ASIC Pin Pair Selected
0h: A
1h: B
2h: C
3h: D
4h: E
5h: F
6h: G
7h: H
8h-Fh: *Reserved*

msb	Byte 2 – High/Low/Selected						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*

b(5:0) – Selected DLL Value

msb	Byte 3 – High/Low/Selected						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*

b(5:0) – Low Pass DLL Value

msb	Byte 4 – High/Low/Selected						lsb
b7	b6	b5	b4	b3	b2	b1	b0

b(7:6) – *Reserved*

b(5:0) – High Pass DLL Value

Note 1: This command is typically used for debug or characterization of the ASIC to DMD interface.

Note 2: The return data is specified by the read parameter data.

Note 3: DMD I/F training tests/calibrates the DLL that is associated with each ASIC pin pair, trying each of the DLL parameter values (0 to 50), looking for a pass (*0*) or fail (*1*) response for each value. Thus, the full training profile for each pin pair is made up of a 51-bit pass/fail result. This result is provided on Full Profile bits 50:0.

Note 4: The full profile response should have a region of contiguous passing DLL values. The highest DLL value for this contiguous region is returned as the High, the smallest DLL value is returned as the Low, and the algorithm selected value as the Selected.

Note 5: This command does not run the DMD I/F training algorithm. This is done automatically by the system. This command returns the result from the most resent training event.

2.3.3.78 Flash Update PreCheck (DDh)

2.3.3.78.1 Read

This command is used to verify that a pending flash update (write) is appropriate for the specified block of the display module flash.

2.3.3.78.2 Read Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Flash Build Data Size (LSByte)
Byte 2	Flash Build Data Size
Byte 3	Flash Build Data Size
Byte 4	Flash Build Data Size (MSByte)

2.3.3.78.3 Return Parameters

The return parameters are described below.

msb	***Byte 1* – Flash PreCheck Results**						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2) – Package Configuration (Identifier)
0h: No Error
1h: Error

b(1) – Package Configuration (Collapsed)
0h: No Error
1h: Error

b(0) – Package Size
0h: No Error
1h: Error

Note 1: This command is used in conjunction with the *Flash Data Type Select* (Section 2.3.3.79) command. This command would be sent after the flash data type has been selected, but before any other flash operation. The purpose is to verify that the desired flash update is compatible, and will fit within the existing flash space, for the current flash configuration.

Note 2: The Flash Build Data Size specifies the size of the flash update package in bytes.

Note 3: When the ASIC software receives the flash build data size, it will verify that the package is appropriate for the specified location. This to include size, identifier, sequence build type, and so forth.

Note 3: A Package Size error indicates that the flash package is too large to fit into the specified location. A few examples are listed below.

- If replacing the entire flash, the size of the flash build exceeds the size of the flash device in the system.
- If replacing the entire flash except for the OEM blocks, the size of the flash build will either overwrite some portion of the existing OEM blocks, or exceeds the size of the flash device in the system.

- If replacing the LOOK block, the size of the flash build exceeds the size of the existing LOOK block in the flash.
- If replacing a single sequence (that is, partial update), the size of the flash build exceeds the size of the existing splash screen.

Note 5: A Package Configuration error indicates that the flash package is not appropriate for the flash update requested. An example is listed below.

- If replacing a single splash screen (that is, partial update), and the specified splash screen index value (Identifier) is not being used in the flash build. Partial updates can only replace an existing flash entity.

Note 6: *If an error is returned by this command, the OEM is responsible for correcting the error before updating the flash. If the OEM chooses to ignore the error and update the flash anyway, the system will allow this. In this case, the OEM is responsible for any problems or system behaviors that arise from this. It should also be noted that this PreCheck does NOT cover all possible mismatches that might arise when replacing blocks or partial blocks in the flash.*

2.3.3.79 Flash Data Type Select (DEh)

2.3.3.79.1 Write

This command is used to specify the type of data that will be written to or read from the Flash of the display module.

2.3.3.79.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	Flash Data Type (See Below)
Byte 2	Optional: Partial Data Identifier (See Byte 1 and Notes 1, 4 and 5)
Byte 3	Optional: Partial Data Identifier (See Byte 1 and Notes 1, 4 and 5)
Byte 4	Optional: Partial Data Identifier (See Byte 1 and Notes 1, 4 and 5)

Default: 00h

msb	*Byte 1*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:0) – Flash Data Type

——— Entire Flash ———

00h – Entire Flash

02h – Entire Flash except OEM Calibration Data and OEM Scratchpad Data

01h, 03h-0Fh – *Reserved*

—— TI Software ——

10h – Main Software Application

11h-1Fh – *Reserved*

—— TI Application Data ——

20h – TI Application Data Set (AOM)

21h-2Fh – *Reserved*

—— OEM Batch Files ——

30h – OEM Batch Files

31h-3Fh – *Reserved*

—— Look Data ——

40h – Look Data Set

41h-4Fh – *Reserved*

—— Sequence Data ——

50h – Entire Sequence Data Set

51h – Partial Sequence Data Set (Reads Only)

52h-5Fh – *Reserved*

—— Degamma/CMT Data ——

60h –	Entire Degamma/CMT Data Set
61h –	Partial Degamma/CMT Data Set (Reads Only)
62h-6Fh –	*Reserved*

—— CCA Data ——

70h –	CCA Data Set
71h-7Fh –	*Reserved*

—— General LUT Data ——

80h –	General LUT Data Set
81h-8Fh –	*Reserved*

—— OEM Splash Screen Data ——

90h –	Entire OEM Splash Screen Data Set
91h –	Partial OEM Splash Screen Data Set
92h-9Fh –	*Reserved*

—— OEM Calibration Data ——

A0h –	OEM Calibration Data Set
A1h-AFh –	*Reserved*

—— OEM Scratchpad Data ——

B0h –	Entire OEM Scratchpad Data Set 0
B1h –	Partial OEM Scratchpad Data Set 0
B2h –	Entire OEM Scratchpad Data Set 1
B3h –	Partial OEM Scratchpad Data Set 1
B4h –	Entire OEM Scratchpad Data Set 2
B5h –	Partial OEM Scratchpad Data Set 2
B6h –	Entire OEM Scratchpad Data Set 3
B7h –	Partial OEM Scratchpad Data Set 3
B8h-BFh –	*Reserved*

Note 1: The flash data type command must be provided each time a new flash write or read operation is desired to ensure that the appropriate data type parameters are provided. The system expects four parameter bytes regardless of whether all four bytes are needed. Any unused bytes should be set to zero.

Note 2: The Flash Data Length (Section 2.3.3.80) must be provided to indicate the amount of flash data that will be provided for each write or read transaction.

Note 3: The specified flash data will be written to or read from Flash using the *Write Flash Start*, *Write Flash Continue*, *Read Flash Start* , and *Read Flash Continue* commands discussed in Section 2.3.3.82, Section 2.3.3.83, Section 2.3.3.84, and Section 2.3.3.85, and respectively.

Note 4: While all of the flash data sets indicated can be written/replaced in their entirety, a few will also support partial writes/updates. Partial update command parameters will use an *odd* command number (such as 91h or B1h) which will indicate that one to three additional command parameter bytes of information must be provided to specify which subset of data is to be updated. The additional command parameter data required is described in Table 2-22.

Table 2-22. 2nd Command Parameter for Partial Flash Data Set Updates (Writes)

Data Type (Writes Only)	2nd CMD parameter (Byte 2)	3rd CMD parameter (Byte 2)	4th CMD parameter (Byte 2)	Comments
Partial OEM Splash Screen Set	Splash Number	N/A	N/A	A Splash screen will be specified by its Splash screen number (see Section 2.3.3.10)
Partial OEM Scratchpad Data Set	Sector Number	N/A	N/A	If this data set is allocated more than one sector, each sector can be specified (0 = 1 st sector, 1 = 2 nd sector, and so forth.)

Note 5: While all of the flash data sets indicated can be read starting at the beginning of the data set, a few will also support read starts at the beginning of a data subset. The partial update command parameters which use an *odd* command number (such as 41h, 43h, and 75h) will indicate that one to three additional command parameter bytes must be provided to specify the start location for these reads. The additional command parameter data required is described in Table 2-23.

Note 6: It is expected that all TI formatted factory calibration data, including the Golden Ratio, the Power-up RGB Currents, and the OEM Thermister LUT Trim data, will be stored in the OEM Calibration block of the flash. It will be the responsibility of the OEM to manage updates to this block, which may require the OEM to read the entire block, modify, and then rewrite the entire block when making an update within the block.

Table 2-23. Additional Command Parameters for Partial Flash Data Set Reads

Data Type (Writes Only)	2nd CMD parameter (Byte 2)	3rd CMD parameter (Byte 2)	4th CMD parameter (Byte 2)	Comments
Partial Sequence Data Set	Look Number	Sequence Index	N/A	A sequence data set will be specified by its sequence index number
Partial CMT Data Set	Look Number	Sequence Index	Degamma/CMT Index Number	A CMT data set will be specified by its CMT index number (see Section 2.3.3.28)
Partial OEM Splash Screen Set	Splash Number	N/A	N/A	A Splash screen will be specified by its Splash screen number (see Section 2.3.3.10)
Partial OEM Scratchpad Data Set	Sector Number	Sub-Sector Address (LSB)	Sub-Sector Address (MSB)	If this data set is allocated more than one sector, each sector can be specified (0 = 1st sector, 1 = 2nd sector, and so forth) The host is also allowed to specify the start address within the sector specified in byte 2. This address to be a relative address within the specified sector (that is, the value can range from 0 to 4096), and must be a 32-bit aligned byte address.

Note 7: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

Note 8: It is recommended that the OEM make use of the Flash Update PreCheck command (Section 2.3.3.78) before updating an existing flash build.

Note 9: The system allows the OEM to allocate up to four separable blocks of flash space for their own use (OEM Scratchpad Data). The OEM can also specify the size of each of these blocks, where each block can be one or more sectors in (one sector = 4KB). This is all defined via Composer. It is the responsibility of the OEM to manage these data sets, including updates, which may require the OEM to read an entire sector, modify, and then rewrite the entire sector when making an update within a sector. References to an unavailable data set will result in an *Invalid Command Parameter Value* Error in the Communication Status (Section 2.3.3.71).

2.3.3.80 Flash Data Length(DFh)

2.3.3.80.1 Write

This command is used to specify the length of the data that will be written to or read from the Flash of the display module.

2.3.3.80.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Flash Data Length (lsb)
Byte 2	Flash Data Length (msb)

Default: 0000h

Note 1: Flash data length must be a multiple of four bytes.

Note 2: The flash data length applies to each write or read transaction, not to the length of the data type selected.

Note 3: The maximum data length allowed for each write transaction is 1024 bytes. The maximum data length allowed for each read transaction is 256 bytes.

Note 4: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

2.3.3.81 Erase Flash Data (E0h)

2.3.3.81.1 Write

This command directs the display module to erase the specified Flash data.

2.3.3.81.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Signature: Value = AAh
Byte 2	Signature: Value = BBh
Byte 3	Signature: Value = CCh
Byte 4	Signature: Value = DDh

Default: All bytes: 00h

Note 1: When this command is executed, the system will erase all sectors associated with the data type specified by the *Flash Data Type Select* (Section 2.3.3.76) command. As such, this command does not make use of the Flash Data Length parameter.

Note 2: Since the process of erasing Flash sectors can take a significant amount of time, the *Flash Erase Complete* status bit in the *Read Short Status* command (Section 2.3.3.68) should be checked periodically (*not continuously*) to determine when this task has been completed. This bit will be set at the start of the erase process, and will be cleared when the erase process is complete. Flash writes should *not* be started before the erase process has been completed.

Note 3: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

Note 4: The signature bytes are used to minimize unintended flash erases. The command OpCode and four signature bytes must be received correctly before this command will be recognized and executed.

2.3.3.82 Write Flash Start (E1h)

2.3.3.82.1 Write

This command is used to write data to the Flash for the display module.

2.3.3.82.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

Note 1: The *Flash Data Length* command (Section 2.3.3.80) must be used to specify how much data will be sent by the *Write Flash Start* command.

Note 2: The *Write Flash Start* command is used to write up to 1024 bytes of data starting at the first address of the data type selected. If more than 1024 bytes are to be written, the *Write Flash Continue* (Section 2.3.3.83) command must be used. Up to 1024 bytes of data can be written with each *Write Flash Continue* command, which starts at the end of the last data written.

Note 3: The *Flash Error* bit of the *Write Short Status* command (Section 2.3.3.67) will indicate if the Flash update was successful. This bit will be set for an error at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until a new data type is selected (selecting a new data type will clear this bit). This will allow the OEM the option of checking the status between each write transaction, or at the end of the update of a specific data type. Once a write transaction has started, the flash status (and this error bit) will not be accessible until the write transaction has completed.

Note 4: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

2.3.3.83 Write Flash Continue (E2h)

2.3.3.83.1 Write

This command is used to write data to the Flash for the display module.

2.3.3.83.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

Note 1: The *Flash Data Length* command must be used to specify how much data will be sent by the *Write Flash Continue* command.

Note 2: The *Write Flash Start* command (Section 2.3.3.82) is used to write up to 1024 bytes of data starting at the first address of the data type selected. If more than 1024 bytes are to be written, the *Write Flash Continue* command must be used. Up to 1024 bytes of data can be written with each *Write Flash Continue* command, which starts at the end of the last data written.

Note 3: The *Flash Error* bit of the *Write Short Status* command (Section 2.3.3.67) will indicate if the Flash update was successful. This bit will be set for an error at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until a new data type is selected (selecting a new data type will clear this bit). This will allow the OEM the option of checking the status between each write transaction, or at the end of the update of a specific data type. Once a write transaction has started, the flash status (and this error bit) will not be accessible until the write transaction has completed.

Note 4: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

2.3.3.84 Read Flash Start (E3h)

2.3.3.84.1 Read

This command is used to read data from the Flash for the display module.

2.3.3.84.2 Read Parameters

This command has no command parameters.

2.3.3.84.3 Return Parameters

This command has a variable number of return parameters as described below.

Parameter Bytes	*Description*
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

Note 1: The *Flash Data Length* command (Section 2.3.3.80) must be used to specify how much data is to be read by the *Read Flash Start* command.

Note 2: The *Read Flash Start* command is used to read up to 256 bytes of data starting at the specified address, or at the first address, of the data type selected. If more than 256 bytes are to be read, the *Read Flash Continue* command (Section 2.3.3.85) must be used. Up to 256 bytes of data can be read with each *Read Flash Continue* command, which starts at the end of the last data read.

Note 3: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

2.3.3.85 Read Flash Continue (E4h)

2.3.3.85.1 Read

This command is used to read data from the Flash for the display module.

2.3.3.85.2 Read Parameters

This command has no command parameters.

2.3.3.85.3 Return Parameters

This command has a variable number of return parameters as described below.

Parameter Bytes	***Description***
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

Note 1: The *Flash Data Length* command (Section 2.3.3.80) must be used to specify how much data is to be read by the *Read Flash Continue* command.

Note 2: The *Read Flash Start* command (Section 2.3.3.84) is used to read up to 256 bytes of data starting at the specified address, or at the first address of the data type selected. If more than 256 bytes are to be read, the *Read Flash Continue* command must be used. Up to 256 bytes of data can be read with each *Read Flash Continue* command, which starts at the end of the last data read.

Note 3: While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

2.3.3.86 Write Internal Register Address (E5h)

2.3.3.86.1 Write

This command is used to specify the address of an internal register in the display module. This command is applicable to both register writes and reads. This command is typically used for debug purposes.

2.3.3.86.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	Register Address (lsb)
Byte 2	Register Address
Byte 3	Register Address
Byte 4	Register Address (msb)

Default: All bytes: 00h

Note 1: If a register address requires less than 4 bytes, the unused MSBytes and/or bits should be set to 0.

Note 2: Register data will be written to or read from the system at the address specified using the commands discussed in Section 2.3.3.87 and Section 2.3.3.88.

2.3.3.87 Write Internal Register (E6h)

2.3.3.87.1 Write

This command is used to write data to an internal register of the display module. This command is typically used for debug purposes.

2.3.3.87.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	*Description*
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4

Note 1: If a register holds less than 4 bytes of data, the unused MSBytes and/or bits should be set to 0.

Note 2: This command is to be used in conjunction with the *Write Internal Register Address* command discussed in Section 2.3.3.86.

2.3.3.88 Read Internal Register (E7h)

2.3.3.88.1 Read

This command is used to read data from an internal register of the display module. This command is typically used for debug purposes.

2.3.3.88.2 Read Parameters

This command has no command parameters.

2.3.3.88.3 Return Parameters

The return parameters are described below.

Parameter Bytes	***Description***
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4

Note 1: If a register holds less than 4 bytes of data, the unused MSBytes and/or bits will be set to 0.

Note 2: This command is to be used in conjunction with the *Write Internal Register Address* command discussed in Section 2.3.3.86.

Note 3: For an internal register read, no pre-fFh will be done.

2.3.3.89 Write Internal Mailbox Address (E8h)

2.3.3.89.1 Write

This command is used to specify the parameters for an internal mailbox transaction in the display module. This command is applicable to both mailbox writes and reads. This command is typically used for debug purposes.

2.3.3.89.2 Write Parameters

The command parameter descriptions follow:

Parameter Bytes	***Description***
Byte 1	Mailbox Select (LSByte)
Byte 2	Mailbox Select
Byte 3	Mailbox Select
Byte 4	Mailbox Select (MSByte)
Byte 5	Mailbox LUT Start Address (LSByte)
Byte 6	Mailbox LUT Start Address
Byte 7	Mailbox LUT Start Address
Byte 8	Mailbox LUT Start Address (MSByte)
Byte 9	Mailbox LUT Select (LSByte)
Byte 10	Mailbox LUT Select
Byte 11	Mailbox LUT Select
Byte 12	Mailbox LUT Select (MSByte)
Byte 13	Mailbox Data Length (LSByte)
Byte 14	Mailbox Data Length
Byte 15	Mailbox Data Length
Byte 16	Mailbox Data Length (MSByte)
Byte 17	(See Below)

Default: All bytes: 00h

msb	***Byte 17***						**lsb**
b7	b6	b5	b4	b3	b2	b1	b0

b(7:3) – *Reserved*

b(2:1) – LUT Packed
0h: No: 1 to 1
1h: Yes: 1 to 4
0h: Yes: 1 to 2
1h: Yes: 3 to 4

b(0) – Write/Read
0h: Write
1h: Read

Note 1: Data length must be a multiple of four bytes.

Note 2: Mailbox data will be written/read using the *Write Internal Mailbox* and *Read Internal Mailbox* commands discussed in Section 2.3.3.90 and Section 2.3.3.91, respectively.

Note 3: Software will automatically open the specified mailbox upon receipt of this command, and will automatically close the opened mailbox once the data length has been reached (using the commands discussed in Section 2.3.3.90 and Section 2.3.3.91).

Note 4: The Mailbox Select parameter specifies the register address of the mailbox to be accessed. Examples of mailboxes to be selected are: FMT Mailbox 0 [40001800], RSC Mailbox 1 [40002820], VGP Mailbox 2 [40005840]. The parameter will be the actual address of the specified mailbox register.

Note 5: The Mailbox LUT Start Address parameter is the start address for the LUT selected, and is typically zero. This is the data value passed to the Mailbox Select register.

Note 6: The Mailbox LUT Select parameter specifies the specific LUT within the selected mailbox that is to be accessed. This is the data value passed to the Mailbox Access Control Register.

Note 7: The address for the Mailbox Access Control Register to be determined by adding four to the Mailbox Select value.

Note 8: The address for the Mailbox Data Register is to be determined by adding sixteen to the Mailbox Select value.

Note 9: The Mailbox Data Length specifies how much data is to be written to or read from the specified mailbox, on a transactional basis.

Note 10: The maximum data length allowed for a write transaction is 1024 bytes. The maximum data length allowed for a read transaction is 256 bytes.

Note 11: The write/read information in this command is only used to indicate a read pre-fetch should be done. This enables the system to more efficiently process read commands. The actual mailbox operation is based on the mailbox write or read command used. Since this is a TI debug command (and because error handling is complex), no error checking will be done to verify that these match (address-specified write/read operation versus write/read command). The outcome of a mismatch: For writes, the mailbox address will be off (since we already did a read pre-fetch) – this is acceptable. For reads, no pre-fetch will be done – this is also acceptable.

Note 12: It is expected that all mailbox data will be just the raw LUT data, with no *header* information included. Any ancillary information that might be required (for example, color space for a splash image) must be set by another means (for example, using register accesses).

Note 13: All data sent using the mailbox must be uncompressed. The FDMA is not available from the I^2C ports, thus decompression would have to be done using software. It was decided that this was not practical.

Note 14: Data unpacking is supported by the mailbox command. Thus, the LUT Packed parameter indicates whether the LUT to be written will require unpacking prior to writing to the specified mailbox. Details are shown in Table 2-24.

Table 2-24. LUT Mailbox Packing Information

LUT Packing	Flash	PBC
NO: 1 to 1	One 32-bit word	One 32-bit word
YES: 1 to 4	One 32-bit word	Four 8-bit words
YES: 1 to 2	One 32-bit word	Two 16-bit words
YES: 3 to 4	Three 32-bit words	Four 24-bit words

2.3.3.90 Write Internal Mailbox (E9h)

2.3.3.90.1 Write

This command is used to write data to an internal mailbox of the display module. This command is typically used for debug purposes.

2.3.3.90.2 Write Parameters

This command has a variable number of parameters as described below.

Parameter Bytes	*Description*
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

Note 1: This command is to be used in conjunction with the *Write Internal Mailbox Address* command Section 2.3.3.89.

Note 2: The address for the mailbox will be auto-incremented. Auto-decrement is not supported.

Note 3: The maximum data length allowed for a write transaction is 1024 bytes.

2.3.3.91 Read Internal Mailbox (EAh)

2.3.3.91.1 Read

This command is used to read data from an internal mailbox of the display module. This command is typically used for debug purposes.

2.3.3.91.2 Read Parameters

This command has no command parameters.

2.3.3.91.3 Return Parameters

The variable number of return parameters are described below.

Parameter Bytes	*Description*
Byte 1	See Below
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

Note 1: This command is to be used in conjunction with the *Write Internal Mailbox Address* command Section 2.3.3.89.

Note 2: The address for the mailbox will be auto-incremented. Auto-decrement is not supported.

Note 3: The maximum data length allowed for a read transaction is 256 bytes.

2.3.3.92 Write External PAD Address (EBh)

2.3.3.92.1 Write

This command is used to specify the parameters for an external PAD device transaction. This command is applicable to both writes and reads.

2.3.3.92.2 Write Parameters

Parameter Bytes	*Description*
Byte 1	PAD Register Start Address (lsb)
Byte 2	PAD Register Start Address
Byte 3	PAD Register Start Address (msb)
Byte 4	Data Length
Byte 5	See Below

msb	*Byte 6*						*lsb*
b7	b6	b5	b4	b3	b2	b1	b0

b(7:1) – *Reserved*

b(0) – Write/Read
0h: Write
1h: Read

Default: All bytes: 00h

Note 1: This command is to be used in conjunction with the *Write External PAD Data (Section 2.3.3.93) and the Read External PAD Data (Section 2.3.3.94)* commands.

Note 2: The maximum data length is 32 bytes, and the minimum data length is 1 byte. This is true for write or read transactions.

Note 3: If a read operation is specified, software will immediately proceed to obtain the requested read data so that it will be available when the *Read External PAD Data* command is received.

Note 4: The PAD registers feature an address auto-increment. As such, single or multiple register accesses only need the indicated start address.

Note 5: No error checking will be done to verify that the address command specified write/read operation matches the actual write/read command. The outcome of a mismatch: For writes, software will write the provided data to the previously provided address – thus no error. For reads, since no read data was obtained, software can return any values it wants (can be all F's - if memory has been allocated, it can be whatever was previously in the allocated memory – whatever is easiest for software to provide with no error checking) – it is accepted that the read data in this case will not be valid.

2.3.3.93 Write External PAD Data (ECh)

2.3.3.93.1 Write

This command is used to write data to an external PAD device.

2.3.3.93.2 Write Parameters

This command has a variable number of parameters as described below.

Parameter Bytes	*Description*
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 ... n	Data Byte 5 ... n

Note 1: This command is to be used in conjunction with the *Write External PAD Address (Section 2.3.3.92) and the Read External PAD Data (Section 2.3.3.94)* commands.

Note 2: The maximum data length allowed for a write transaction is 32 bytes.

2.3.3.94 Read External PAD Data (EDh)

2.3.3.94.1 Read

This command is used to read data from an external PAD device.

2.3.3.94.2 Read Parameters

This command has no command parameters.

2.3.3.94.3 Return Parameters

The variable numbers of return parameters are described below.

Parameter Bytes	*Description*
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 ... n	Data Byte 5 ... n

Note 1: This command is to be used in conjunction with the *Write External PAD Address (Section 2.3.3.94) and the Write External PAD Data (Section 2.3.3.92)* commands.

Note 2: The maximum data length allowed for a read transaction is 32 bytes.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265