DLPC120-Q1 I²C User Interface

Programmer's Guide

Literature Number: DLPU055
March 2018

Contents

List of Figures

List of Tables

Introduction

Trademarks

DLP is a registered trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

Purpose & Scope

This document defines the I^2C user interface to the DLPC120-Q1. It provides descriptions of each register and explains command list operations.

The document is organized by I^2C function and gives examples of how to use the registers while executing system operations. Refer to System Overview for a description of the different features of the DLPC120.

The DLPC120-Q1 is compatible with two DMD components:

- DLP3000-Q1 0.3 WVGA Type A100 DMD
- DLP3030-Q1 0.3 WVGA S450 DMD

However, the details of this document apply specifically to the DLP3030-Q1 chipset.

Applicable Documents

The following documents and drawings may aid in the understanding of this plan.

DLPS096	DLPC120-Q1 Datasheet
2511810	DLP3000-Q1 0.3 WVGA Type A DMD Datasheet
DLPS076	DLP3030-Q1 0.3 WVGA S450 DMD Datasheet
DLPA084	LED Driver for DLP3030-Q1 Displays Application Note
DLPU057	Piccolo SPI Command Interface
DLPU061	Piccolo Software Programmer's Guide

Reference Mnemonics

ASIC – Application Specific Integrated Circuit

AST – ADC Sampling Timer

BIST – Built-in Self Test

CMT – Contour Mitigation Table

DLPC – TI DLP® Controller

DMD – Digital Micro-mirror Device

HUD – Head-Up Display

IC – Integrated Circuit

LDC – LED Drive Control

LED – Light Emitting Diode

LUT – Look Up Table

PLL – Phase Locked Loop

RGB – Red Green Blue

SEQ – Sequence

UCA – User-supplied Command Access

UMC – Unified Memory Controller

System Overview

The DLPC120-Q1 is responsible for accepting video input and formatting the data to display on the DMD while simultaneously controlling RGB LEDs in order to create a real-time image. It is also responsible for controlling the power-up and power-down events of the DMD, based on external system control or temperature input from the DMD. Combined with an external dimming circuit and micro-controller, the DLP120 supports a wide dimming range > 5000:1 for HUD applications. Additionally, the DLPC120-Q1 can be configured to operate in different display modes, supports several diagnostic tests, and offers system event triggers for external ADC sampling. For initialization and configuration, the DLPC120-Q1 accesses an external flash memory chip. The contents of the flash are provided by TI, and the settings can be customized for each customer's design requirements.

Figure 1 below is a block diagram of the DLPC120-Q1 and DLP3030-Q1 chipset.

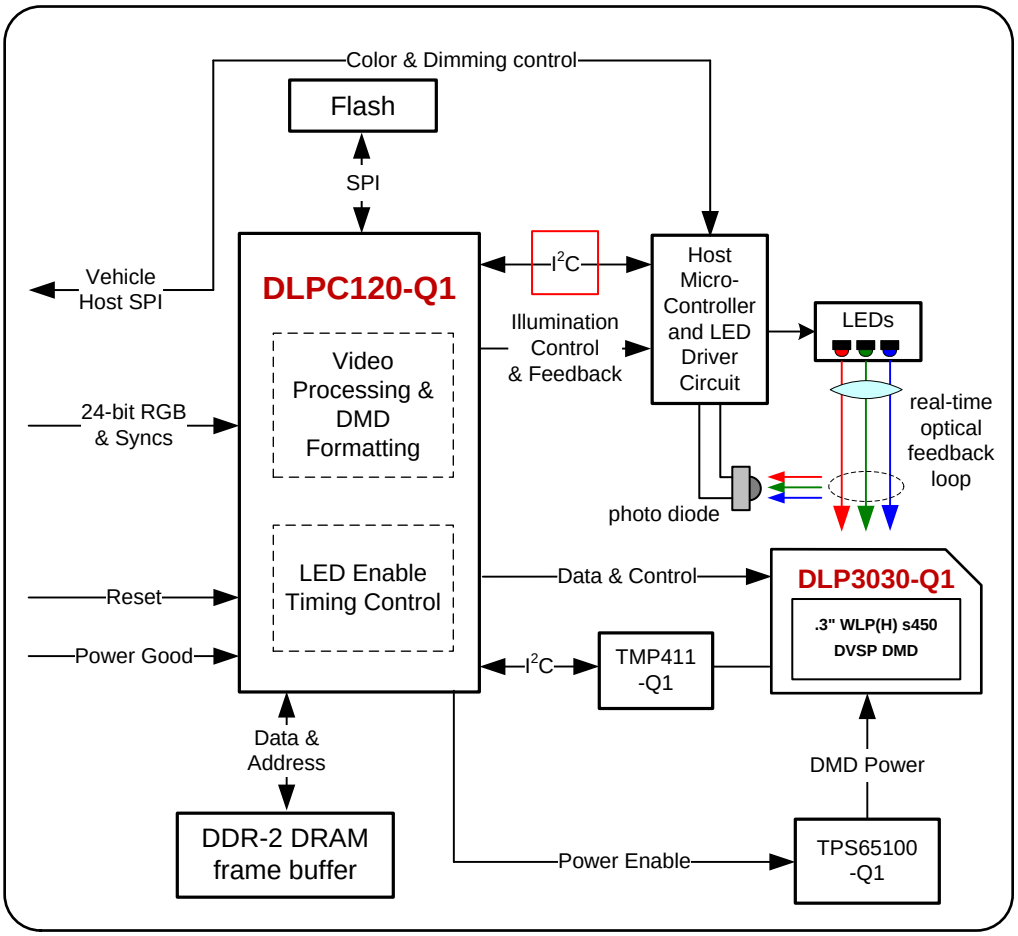


Figure 1. DLPC120-Q1 Chipset Diagram

1.1 Read and Write Formats

The I²C protocol used in communicating information to the DLPC120-Q1 consists of a serial data bus that is rated at speeds between 100 kHz and 400 kHz. Refer to the DLPC120-Q1 datasheet for detailed specifications.

1.1.1 Unused and Reserved Bits

Within the register descriptions there are two special cases for bits:

- **Unused bits** – These bits are not instantiated in the DLPC120-Q1. Any value written to these bits will be ignored by the DLPC120-Q1.
- **Reserved bits** – These bits are reserved for debugging and development purposes. These bits should be set to the value that is specified for them. Care must be taken to avoid setting incorrect values in these fields.

1.1.2 Slave Receive Mode (Write)

When the DLPC120-Q1 is operating in the slave-receiver configuration, the first byte following the start condition is the DLPC120-Q1 device write address (0x36). The next byte specifies the register sub-address. Each register sub-address accepts 4 bytes of data.

Table 1-1. Slave Write Command

Write Command		
Address	Sub-Address	Data
(8-bit) 0x36	(8-bit) 0xAA	(32-bit) 0xddddddddd
(0xAA = Register Address, 0xddddddddd = write data)		

Example: Write 0x00000001 to image curtain register (Register 0xE9)

Send data 36 e9 00 00 00 01. See Figure 1-1 below:

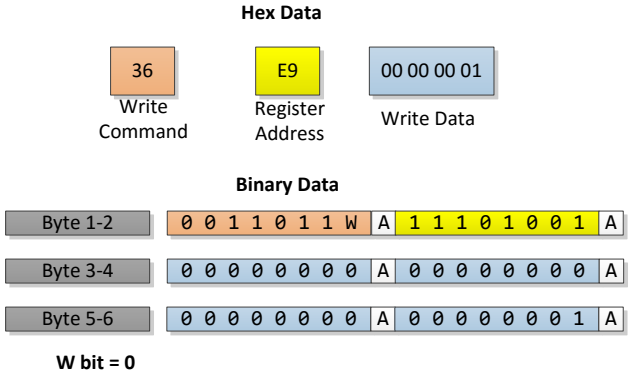


Figure 1-1. Write Register Example

1.1.3 Slave Transmit Mode (Read)

When the DLPC120-Q1 is operating in the slave-transmitter mode, the first byte following the start condition is the DLPC120-Q1 device read address (0x37). The selected register to read must be specified with a write previously.

Table 1-2. Slave Read Command

Read Command (Part 1) – Write requested address		
Address	Sub-Address	Data
(8-bit) 0x36	(8-bit) 0x15	(8-bit) 0xAA
(0xAA=Address of requested register)		
Read Command (Part 2) – Read requested address data		
(8-bit) 0x37		(32-bit) 0xddddddddd
(0xddddddddd=Read data of requested register)		

Example: Read the HUD Version (Register 0x04)

1. Send data 36 15 04 to write the requested register
2. Send data 37 to start the read

See Figure 1-2 below:

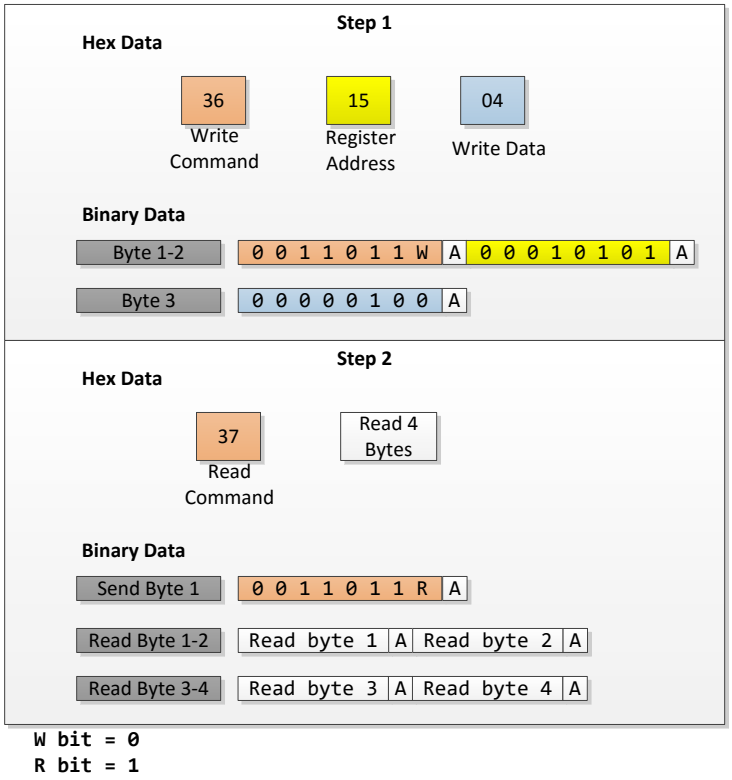


Figure 1-2. Read Register Example

1.2 Command List Execution (Registers 0x40 and 0x41)

Registers used for this operation:

Table 1-3. Command List Base Address (Register 0x40)

Bit	Description	Reset	Type	Notes
24:0	**Base Address**	0x0	Write	Base addresses are listed in a flash file's Configuration File (Chapter 2)
31:25	Unused	0x0	Write	

Table 1-4. Command List Execution (Register 0x41)

Bit	Description	Reset	Type	Notes
0	**Command List Execute** 1: Execute	0x0	Write	Self clears after command list execution
31:1	Unused	0x0	Write	

DLPC120-Q1 uses command list execution to configure different modes of operation such as input resolutions, internal test patterns, and splash screens.

These commands are stored in flash memory. In order to execute a command list:

1. Write the base address of the command list to register 0x40.
2. Write 0x00000001 to register 0x41 to execute.

Configuration File (.cfg)

2.1 Configuration File Overview

A Configuration File is provided when a new flash file is requested. The flash data can be extracted from the Configuration File using the Automotive Control Program. This flash data can then be programmed to the SPI Flash. An example Configuration File can be found on the TI extranet.

A configuration file has two purposes:

1. Contains the full hex or binary data that is programmed to the SPI Flash that is accessed by the DLPC120-Q1.
2. Contains header information about the flash data that is contained within it.

This file is in XML format so that custom software tools can easily parse its contents.

2.2 Header Information

The config file contains high-level descriptions about the flash contents including:

Temperature Data	DMD Park Temperatures
Command Lists	Name
	Command List Base Address
	Name
	...
Dimming/Sequence LUTs	RGB Duty Cycle
	On/Off Duty Cycle
	Attenuation/Pulsing
	LUT Base Address
	RGB Duty Cycle
	...

For a detailed breakdown of the flash contents, please refer to Section A.2.1. The Command List Base Address information can be used to call command lists as explained in Section 1.2.

Power-On Initialization

3.1 Power-On Overview

There are several tasks that should be performed at power-on.

1. PLL configuration and reset
2. UMC initialization
3. DMD drive strength control selection
4. Built-in Self Tests
5. Enable video

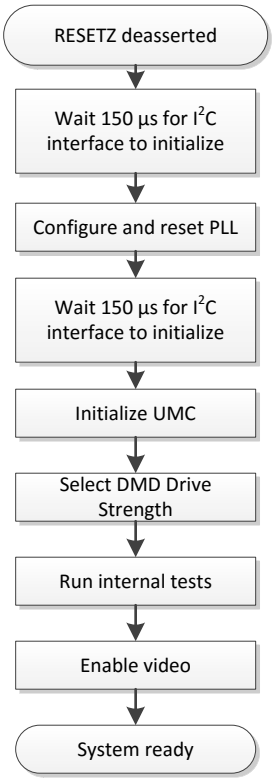


Figure 3-1. Power-On Overview

3.2 PLL Configuration

The following registers are used for PLL configuration:

Table 3-1. Support Functions Control (Register 0xA0)

Bit	Description	Reset	Default Config	Type	Notes
2:0	Reserved	0x0	0x1	Write	This value will change during operation and can also vary based on default flash configuration.
3	**DMD Park** 0: Unpark DMD 1: Park DMD	0x0	0x0	Write	Only supported with DLP3000-Q1 . For DLP3030-Q1 park and unpark methods, please refer to Chapter 11.
4	**PLL Reset** 1: Force PLL to reset	0x0	0x0	Write	
31:5	Unused	0x0	0x0	Write	

Table 3-2. PLL Control (Register 0xA4)

Bit	Description	Reset	Type	Notes
7:0	Reserved	0x3A	Write	
8	**PLL Spread Spectrum Enable**	0x0	Write	
31:9	Unused	0x0	Write	

Table 3-3. PLL Clock Configuration (Register 0xA5)

Bit	Description	Reset	Type	Notes
11:0	**PLL Spreading Slope**	0x35F	Write	
15:12	Unused	0x0	Write	
23:16	**PLL Spreading Frequency**	0x2C	Write	
31:24	Unused	0x0	Write	

The Phase Locked Loop is used to create the base clock for the HUD ASIC. It is configured to receive an input clock of 16 MHz and provide a stable internal source clock of 312 MHz. The PLL spread spectrum function is capable of down-spreading the output 312-MHz clock to a programmed percentage and spreading frequency.

Spreading is disabled by default. If spreading is enabled, the default setting is a down-spread by 2% at a spreading frequency of 60 kHz.

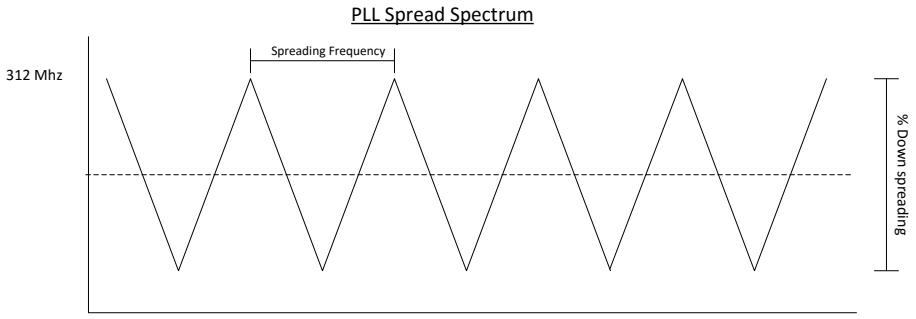


Figure 3-2. PLL with Spread Spectrum

3.2.1 PLL Spread Spectrum Enable Procedure

The procedure for enabling the PLL is shown below:

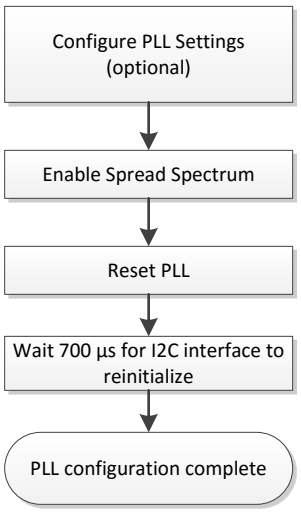


Figure 3-3. PLL Spread Spectrum Enable Procedure

The PLL must be reset in order for configuration changes to be applied. The PLL reset should be treated similarly to a full ASIC reset because all register values will return to their default startup state. This means that the PLL configuration register may not contain the values that were set before the PLL reset occurred even though the settings have been applied.

3.2.2 PLL Configuration Options

In order to adjust the down-spread percentage or the spreading frequency, two values must be modified. Both values are located in register 0xA5. The details of this register can be found at the beginning of Section 3.2.

Table 3-4 contains valid settings for PLL spreading.

Note: The modes listed below have been validated. All other modes are unsupported and may cause system issues.

Table 3-4. Supported Spread Spectrum Settings

	60 kHz		30 kHz		100 kHz	
	2%	1%	2%	1%	2%	1%
Spreading Slope	0x35F	0x1AF	0x1AF	0xD8	0x59E	0x2CF
Spreading Frequency	0x2C	0x2C	0x59	0x59	0x1B	0x1B

3.2.3 Example: Set PLL down-spreading to 2% at a frequency of 60 kHz

From Section 3.2.2, find the Spreading Slope and Spreading Frequency	Spreading Slope = 0x35F Spreading Frequency = 0x2C
After system startup, write these 2 values to register 0xA5	Write 0x002C035F to register 0xA5
Enable PLL Spread Spectrum	Write 0x0000013A to register 0xA4
Reset PLL	Write 0x00000010 to register 0xA0

3.3 UMC Initialization

The UMC (Unified Memory Controller) manages and controls memory accesses to external DDR2. The UMC must be initialized in order to allow the DLPC120-Q1 to have access to frame data.

Table 3-5. UMC Control (Register 0xAC)

Bit	Description	Reset	Type	Notes
31:0	**Reserved** Sets address pointer location in the UMC block of the DLPC120-Q1.	0x0	Write	Refer to Section 3.3.1 for exact usage.

Table 3-6. UMC Read Data 1 (Register 0xAE)

Bit	Description	Reset	Type	Notes
31:0	**Reserved** Provides status bits to verify during DDR Initialization procedure.	0x0	Read	Refer to Section 3.3.1 for exact usage.

Table 3-7. UMC Read Data 2 (Register 0xAF)

Bit	Description	Reset	Type	Notes
31:0	**Reserved** Provides status bits to verify during DDR Initialization procedure.	0x0	Read	Refer to Section 3.3.1 for exact usage.

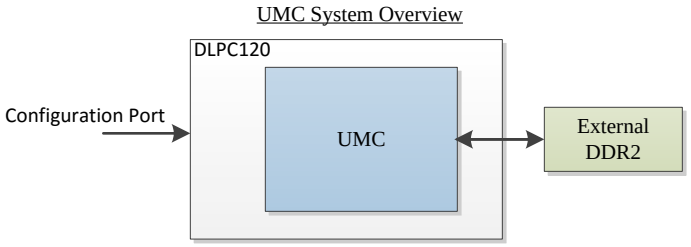


Figure 3-4. UMC System Overview

3.3.1 DDR External Termination Initialization Procedure

In order to initialize UMC for the DLPC120-Q1 ASIC, the app flash file that is programmed to the ASIC's flash memory must contain four command lists that configure the memory. Refer to a TI Applications engineer to verify the UMC configuration of an app flash file or have one created.

The following steps are used to initialize the UMC after the proper flash file has been programmed into the ASIC's flash memory. Figure 3-4 shows a flow chart of these steps.

1. Write 0x00000001 to register 0x41(command list execution) to start part 1 of the UMC initialization.
2. Write 0x0000080C to register 0xAC to set the Address pointer to General Status Register (PGSR).
3. Write 0xAF to register 0x15 in order to read back register 0xAF.
4. Read back 4 bytes. The value should read 0x0000000B. If the value is not 0x0000000B, repeat Step 2-4 until the correct value is received or a timeout condition is reached*.
5. Write 0x01 to address 0x41(command list execution) to start part 2 of the UMC initialization.
6. Write 0x00000048 to address 0xAC to set Address pointer to Powerup Status (POWSTAT).
7. Write 0xAE to address 0x15 in order to read back register 0xAE.
8. Read back 4 bytes. The value should read 0x00000001. If the value is not 0x00000001, repeat Step 6-8 until the correct value is received or a timeout condition is reached*.
9. Write 0x01 to address 0x41 to start part 3 of the UMC initialization.
10. Write 0x00000008 to address 0xAC to set Address Pointer to State Status Register (STAT).
11. Write 0xAE to address 0x15 in order to read back register 0xAE.
12. Read back 4 bytes. The value should read 0x00000003. If the value is not 0x00000003, repeat Step 10-12 until the correct value is received or a timeout condition is reached*. This verifies that the DDR memory is in the access state and is ready for normal transactions. The DDR memory cannot be trained unless it is in the access state.

13. Write 0x0000080C to address 0xAC to set the Address pointer to General Status Register (PGSR).
14. Write 0xAF to address 0x15 in order to read back register 0xAF.
15. Read back 4 bytes. The value should read 0x0000001B. If the value is not 0x0000001B, repeat Step 13-15 until the correct value is received or a timeout condition is reached*.
16. Write 0x01 to address 0x41(command list execution) to start part 4 of the UMC initialization. This will set the UMC_done signal to signify the Initialization is complete.

***Timeout condition:** The TI reference Piccolo code uses 1ms as the duration of the timeout for each status check during DDR initialization. This timeout is reset before the first read of each status value. A timeout length of less than 1ms has not been verified and could result in the detection of false failures due to insufficient DDR initialization time.

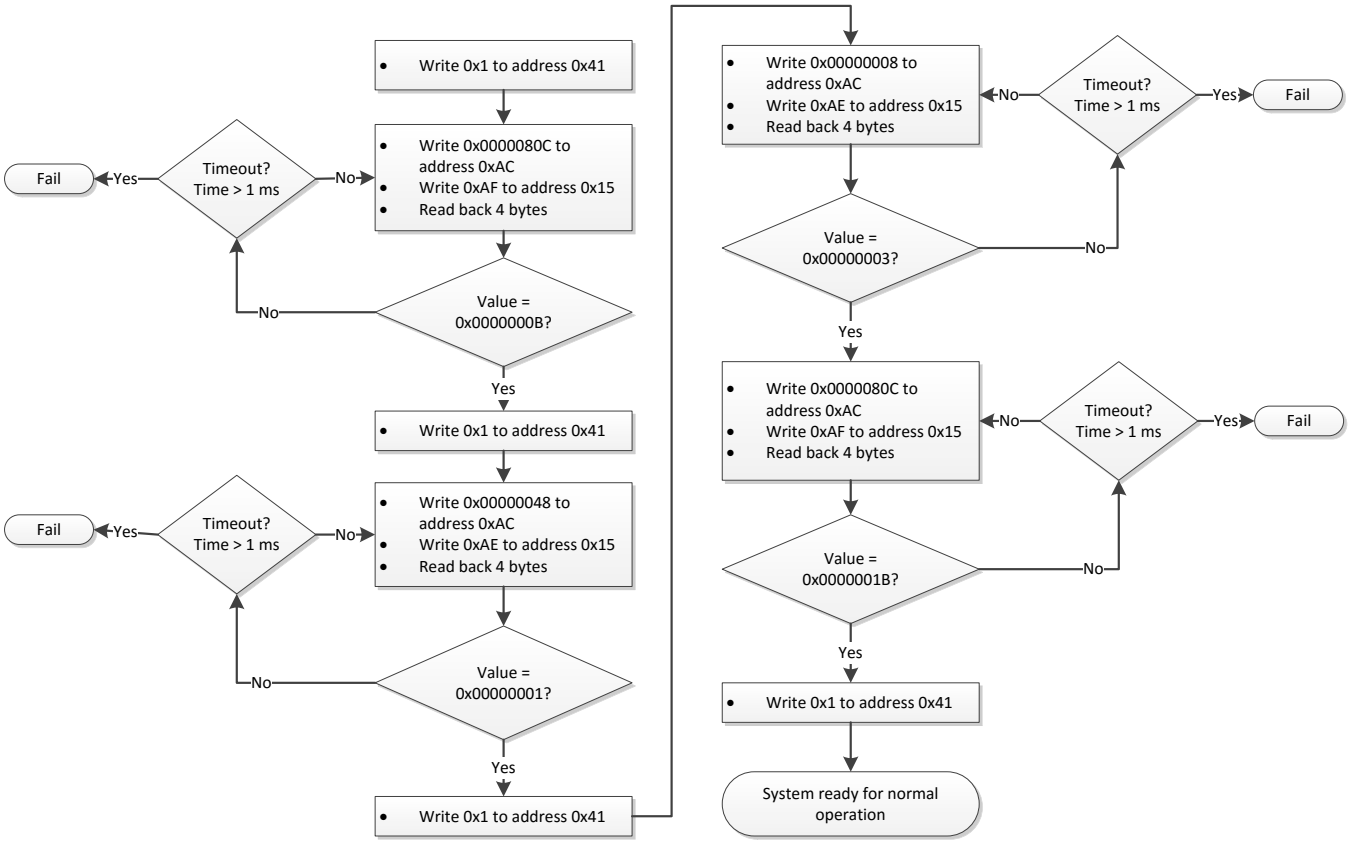


Figure 3-5. DDR External Termination Initialization

3.4 DMD Drive Strength Control

The following register is used to set the DMD drive strength.

Table 3-8. DMD Drive Control (Register 0x65)

Bits	Description	Reset	Type	Notes
1:0	**DMD Drive Control** 00_2: 6 mA 01_2: 10 mA 10_2 or 11_2: 12 mA	0x2	Read	
31:2	Unused	0x0	Read	

The appropriate DMD drive strength varies based on PCB design. The ASIC and DMD IBIS models should be used in addition to a specific PCB layout to verify the necessary drive strength for that PCB layout. Additionally, the drive strength must operate the DMD signals within the operating specifications listed in the DLP3030-Q1 datasheet.

It is recommended that 12 mA be used for the TI HUD reference design.

The following DLPC120-Q1 signals are affected by modifying DMD Drive control:

Table 3-9. Variable Drive Strength Signals

DMD_D0	DMD_D12
DMD_D1	DMD_D13
DMD_D2	DMD_D14
DMD_D3	DMD_DCLK
DMD_D4	DMD_LOADB
DMD_D5	DMD_SCTRL
DMD_D6	DMD_TRC
DMD_D7	DMD_DAD_OEZ
DMD_D8	DMD_DAD_BUS
DMD_D9	DMD_DAD_STRB
DMD_D10	DMD_SAC_BUS
DMD_D11	DMD_SAC_CLK

3.5 Start-Up Tests

The following self tests may be performed before normal operation:

1. DDR2 Built-in Self Test (Section 7.2)
2. Flash Built-In Self Test (Section 7.3)
3. DMD JTAG BIST (Section 7.4)
4. System Built-In Self Test (Section 7.5)
5. Front End Video Built-In Self Test (Section 7.6)

3.6 HUD Status

Table 3-10. HUD Status (Register 0x03)

Bits	Description	Reset	Type	Notes
1:0	**Reserved**	0x0	Read	Value may change during operation.
3	**Flash Controller Status** 0: Flash is idle 1: Flash is busy, wait until idle	0x0	Read	
31:4	**Reserved**	0x0	Read	Value may change during operation.

The HUD Status register must be read to verify whether the ASIC is accessing the flash controller. This is used during the startup procedure in order to verify whether the ASIC has completed an action that requires flash access.

In the TI sample Piccolo code this status is read during the startup procedure two times:

1. When ASIC comes out of reset
    1. The ASIC must read default configuration when reset is de-asserted. This allows the Piccolo to wait until this action is completed.
2. When PLL is initialized

1. The ASIC will read default configuration again since PLL initialization will reset the ASIC. This allows the Piccolo to wait until this action is completed.

Version Info

4.1 HUD Version (Register 0x04)

Table 4-1. HUD Version (Register 0x04)

Bits	Description	Reset	Type	Notes
7:0	**Major Release Version Number**	0x00	Read	
15:8	**Minor Release Version Number**	0x01	Read	
31:16	**Build Number**	0x55	Read	

The reset values shown above are specific to the ASIC.

4.2 Flash Configuration and Version (Register 0x05)

Table 4-2. Flash Version (Register 0x05)

Bits	Description	Reset	Type	Notes
31:0	**CRC Flash Data Checksum**	0x0	Read	

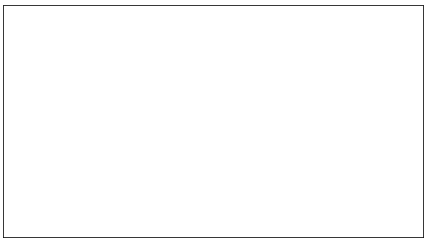
The Flash Version is set by the default configuration that is stored in the DLPC120-Q1 SPI flash. Therefore the Flash Version value stored in register 0x05 will read back a unique checksum value for each unique flash file. This register value is created by a checksum when a TI applications engineer creates the flash data file. This checksum value is not the same as the checksum returned by the Flash BIST in Section 7.3. See a TI Applications engineer for confirmation of the expected Flash Version.

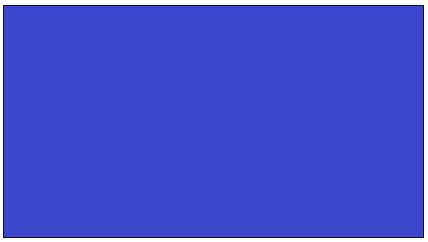
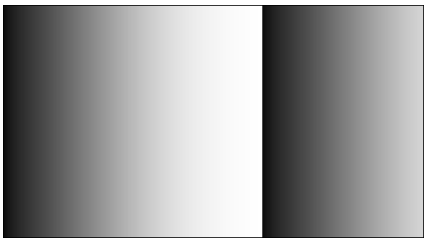
5.1 Internal Test Patterns

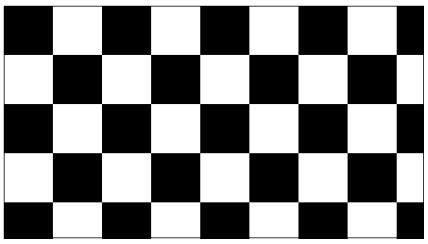
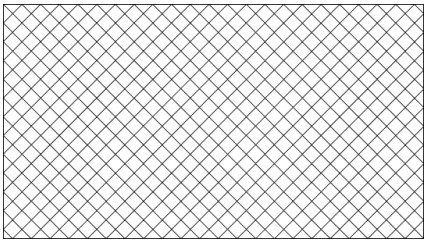
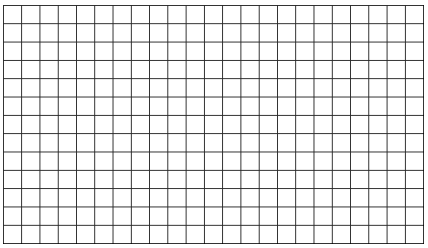
Internal test patterns are configured as command lists. See Section 1.2 for command list execution.

Built-in test patterns can be generated and will operate using an internal clock. The test patterns operate at a resolution of 864x480 and a frame rate of 60 Hz.

The following test patterns can be generated:

Description	Example
Solid White	
Solid Black	
Solid Red	
Solid Green	

Solid Blue	
Gray Horizontal Ramp	
Red Horizontal Ramp	
Green Horizontal Ramp	
Blue Horizontal Ramp	
Gray Vertical Ramp	

Checker	
Diagonal Grid	
Grid	

5.1.1 Smooth Transition to Test Pattern

Test patterns require a maximum of three 60-Hz video frames before the output image becomes stable. In order to prevent visible image artifacts when transitioning from any display mode to any test pattern, the following procedure is recommended.

1. Enable image curtain (Section 5.4)
2. Execute test pattern command list
3. Delay 3 video frames
4. Disable image curtain

The flowchart below demonstrates this process:

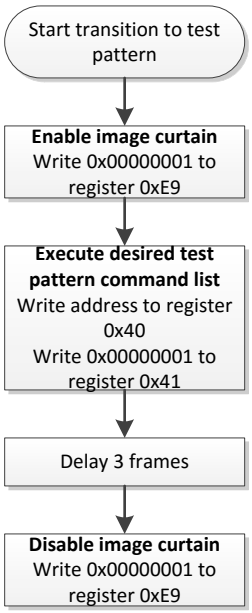


Figure 5-1. Smooth Test Pattern Transition Procedure

5.2 External Input Resolutions

External input resolutions are configured as command lists. See Section 1.2 for command list execution.

When an external video source is used, one of the following command lists must be executed in order to set the ASIC to external video mode and to set the output resolution to match the source video data.

The following is a list of supported resolutions:

Table 5-1. External Input Resolutions

Resolution
960x480
960x250
960x240
960x160
864x480
854x480
854x240
853x480
852x480
800x480
640x480
640x240
640x160
500x250
480x240
400x240
320x240
320x160
320x120

864x480 60 Hz is the native resolution of the 0.3” WVGA DMD.

5.2.1 Smooth Transition to External Video

External video configuration requires a maximum of three video frames before the output image becomes stable. In order to prevent visible image artifacts when transitioning from any display mode to any external video configuration, the following procedure is recommended.

1. Enable image curtain (Section 5.4)
2. Execute external input resolution command list
3. Delay 3 video frames
4. Disable image curtain

The flowchart below demonstrates this process:

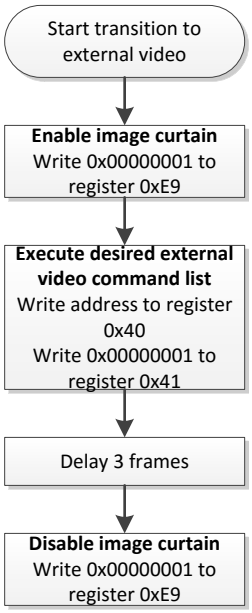


Figure 5-2. Smooth External Video Transition Procedure

5.3 Splash Screens

Splash screens are images stored in the ASIC app flash memory. These images can be selected using command list execution (see Section 1.2).

After calling a splash screen command list, the following register is used for further splash control:

Table 5-2. Splash Function Control (Register 0x98)

Bits	Description	Reset	Type	Notes
0	**Splash Screen Function Enable** 1: Enable 0: Disable	0x0	Write	Do not set. This bit is set as needed by command lists.
1	**Splash Curtain Enable** Black curtain during splash loading 1: Enable 0: Disable	0x0	Write	
2	**Splash Reload** Repeatedly load splash screen 1: Enable 0: Disable	0x0	Write	Required for splash bezel adjustment and splash CMT update.
31:3	Unused	0x0	Write	

608x684 (the native DMD resolution, in diamond pixel orientation) and 864x480 are the only allowable splash image sizes. Each splash image requires 1244160 bytes of flash memory.

A splash image requires 22 frames at 60 Hz to load from flash and display. It is possible to setup an interrupt event signal for Splash Load Done. Refer to Chapter 10 for details about setting up interrupts.

Contact a TI Applications engineer for support in adding desired image data.

To exit splash screen mode, an appropriate external input resolution command list or internal test pattern command list must be executed.

Because splash screens must be loaded and processed from flash, software must take actions to reload splash data after two events:

1. Updating bezel adjustment (see Section 5.3.2)
2. Updating CMT table (see Section 5.3.3)

5.3.1 Smooth Transition to Splash Screen

Splash screen configurations require a maximum of 22, 60-Hz video frames to load image data from flash and display before the output image becomes stable. In order to prevent visible image artifacts when transitioning from any display mode to any splash image configuration, the following procedure is recommended.

1. Enable image curtain (Section 5.4)
2. Execute splash screen command list
3. Delay for 22 video frames
4. Disable image curtain

If the current video setting is 608x684 resolution and the splash image is 864x480 resolution, then it is necessary to transition to an intermediate test pattern with 864x480 before transitioning to the splash image. This intermediate transition to test pattern allows the DLPC120-Q1 to adjust video scaling in order to prevent unexpected pixel artifacts in the splash image. See steps below:

1. Enable image curtain (Section 5.4)
2. Execute 864x480 test pattern command list (example: black test pattern)
3. Delay for 2 frames to allow video scalars to adjust to 864x480
4. Execute splash screen command list
5. Delay for 22 video frames to allow splash to configure
6. Disable image curtain

The flowchart below demonstrates this process:

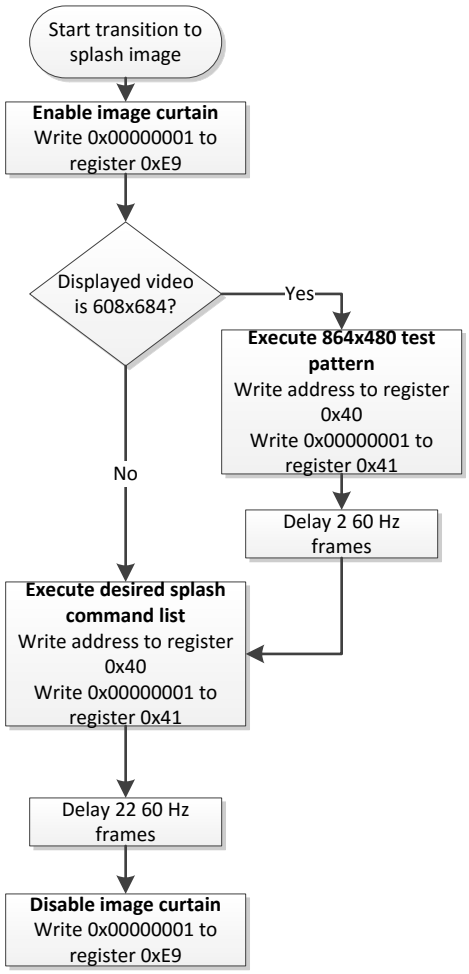


Figure 5-3. Smooth Splash Transition Procedure

5.3.2 Splash Bezel

While in splash image mode, bezel adjustments will not be applied until the next splash load. By setting bit 2 of register 0x98 to a value of 1, splash will repeatedly reload. This allows a new bezel adjustment to be applied at a maximum of once every 22 frames at 60 Hz.

5.3.3 Splash CMT Table Update

Refer to Chapter 9 for more information about CMT tables and refer to Chapter 10 for more information about updating CMT tables.

While in splash image mode, CMT table index changes will not be applied while splash is loading. In order to change the CMT table index, Splash Reload (bit 2 of register 0x98) should be disabled. If a splash load is already in progress when Splash Reload is disabled, the splash will finish the current loading. Therefore, it will take up to 22 frames at 60 Hz in order for the splash loading to complete once Splash Reload is disabled. When splash loading has completed, the new CMT table index can be written. Once the CMT table index is written through I^2C, Splash Reload can be enabled again for splash bezel adjustment (described in Section 5.3.2).

5.4 Image Curtain (Register 0xE9)

The image curtain will override any other display data with solid black when it is enabled.

Table 5-3. Image Curtain (Register 0xE9)

Bits	Description	Reset	Type	Notes
0	**Curtain Enable** 1: Enable curtain 0: Disable curtain	0x0	Write	
31:1	Unused	0x0	Write	

5.5 Image Flip

Image flip allows you to flip any displayed image. The four unique image flip settings are configured as command lists. See Section 1.2 for command list execution.

Flips can be applied in two directions as shown below.

Long flip:

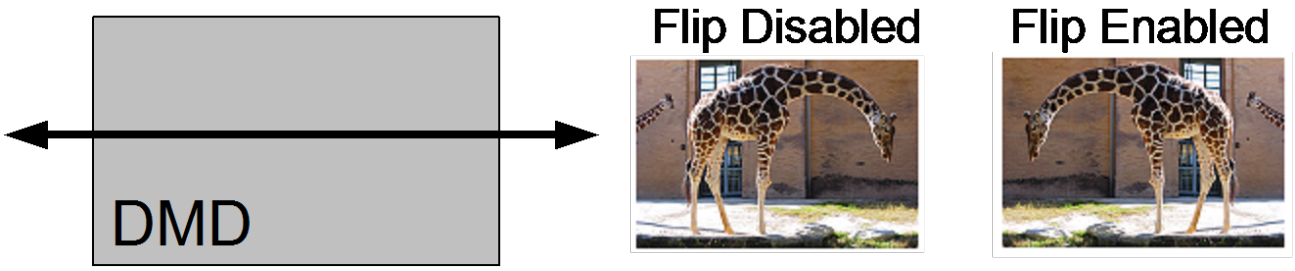


Figure 5-4. Long Flip

Short flip:

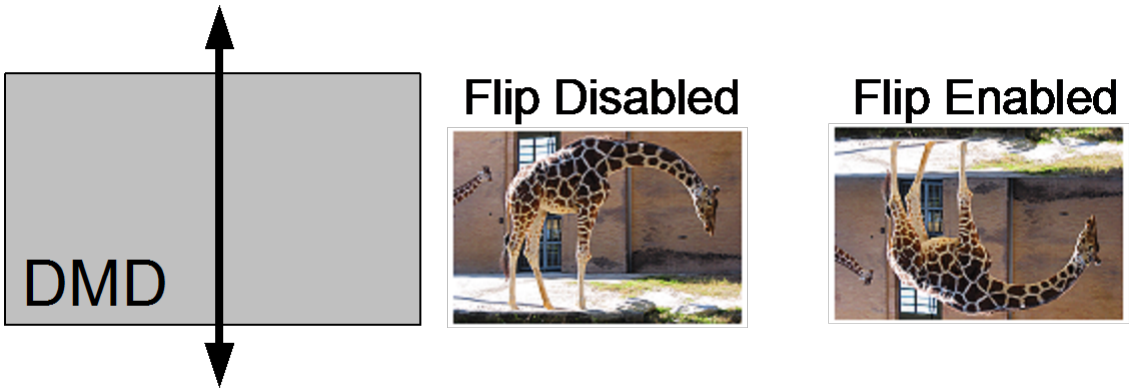


Figure 5-5. Short Flip

5.6 Bezel Adjustment (Registers 0x38 and 0x39)

Table 5-4. Bezel Adjustment Function Enable (Register 0x38)

Bits	Description	Reset	Type	Notes
0	**Bezel Adjustment Function Enable** 0: Disable 1: Enable	0x0	Write	
1	Reserved	0x0	Write	Set to 0

Table 5-4. Bezel Adjustment Function Enable (Register 0x38) (continued)

Bits	Description	Reset	Type	Notes
31:2	Unused	0x0	Write	

Table 5-5. Bezel Adjustment Offset (Register 0x39)

Bits	Description	Reset	Type	Notes
10:0	**Horizontal pixel offset**	0x0	Write	Must be a multiple of 4.
15:11	Unused	0x0	Write	
26:16	**Vertical pixel offset**	0x0	Write	Must be a multiple of 2.
31:27	Unused	0x0	Write	

This register sets the location of the first pixel of image data in any display mode. The direction of positive adjustment is determined by the direction that pixels are drawn in. The direction of positive adjustment will flip if the image is flipped.

Negative offset values are determined using two's complement binary.

The vertical offset can be shifted ±50% of the DMD's array's vertical dimension (684*0.5 = ±342 lines). The horizontal offset can be shifted by ±15% of the DMD array's horizontal dimension (608*0.15 = ±88 pixels).

This feature can be enabled or disabled via three methods: command list execution, direct register writes, or configured as part of the default configuration routine. Note that the Application Flash provided by TI will automatically disable the Bezel Adjustment feature, so it will need to be explicitly enabled prior to utilizing it. The benefits of disabling this feature are described in Section 5.6.4. Note that if the Bezel Adjustment feature is disabled while actively displaying content, there will be a one frame video loss appearing as a dark flash.

In splash screen mode, bezel adjustments will not update immediately because the splash screen data must be reloaded in order for the bezel adjustments to be applied. For bezel adjustment in splash screen mode, see the Piccolo software guide.

5.6.1 Example: Move image 12 pixels right and 10 pixels down

Figure 5-6. Bezel Adjust Example 1

Step 1: Horizontal Offset

Convert 12 to hex	12 = 0xC

Step 2: Vertical offset

Convert 10 to hex	10 = 0xA

Step 3: Combine values

Combine values into 4-byte hex	0x000A000C

Step 4: Write value to register 0x39

Write data	0x36 0x39 0x000A000C

5.6.2 Example: Move image 12 pixels left and 10 pixels up

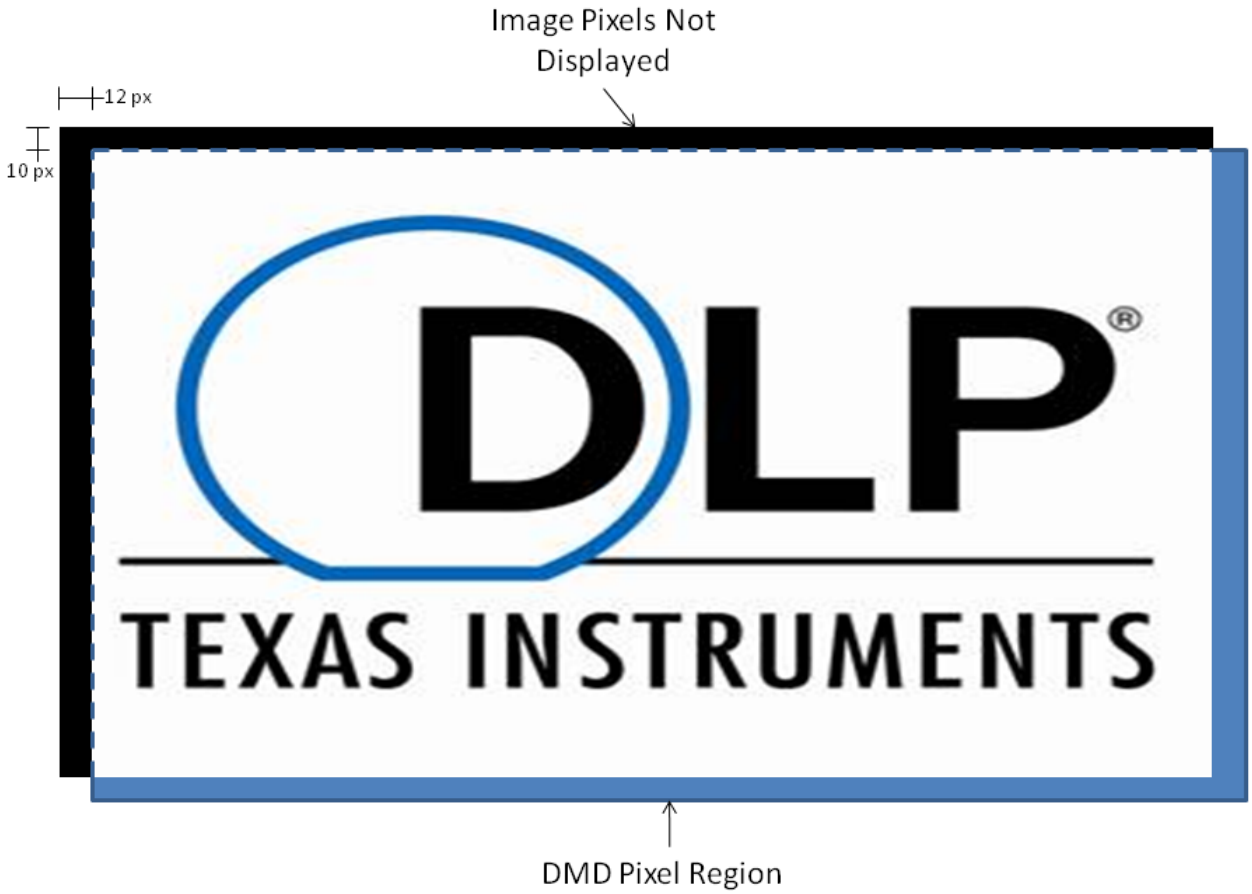


Figure 5-7. Bezel Adjust Example 2

Step 1: Horizontal offset

Convert 12 to 11-bit binary	12 = 00000001100
Find the two's complement	11111110100

Step 2: Vertical offset

Convert 10 to 11-bit binary	10 = 00000001010
Find the two's complement	11111110110

Step 3: Combine values

Combine values into 32-bit binary	00000**11111110110**00000**11111110100**
Convert to hex	0x07F607F4

Step 4: Write value to register 0x39

Write data	0x36 0x39 0x07F607F4

5.6.3 Example: Move image 12 pixels right and 10 pixels down while image has a long flip applied

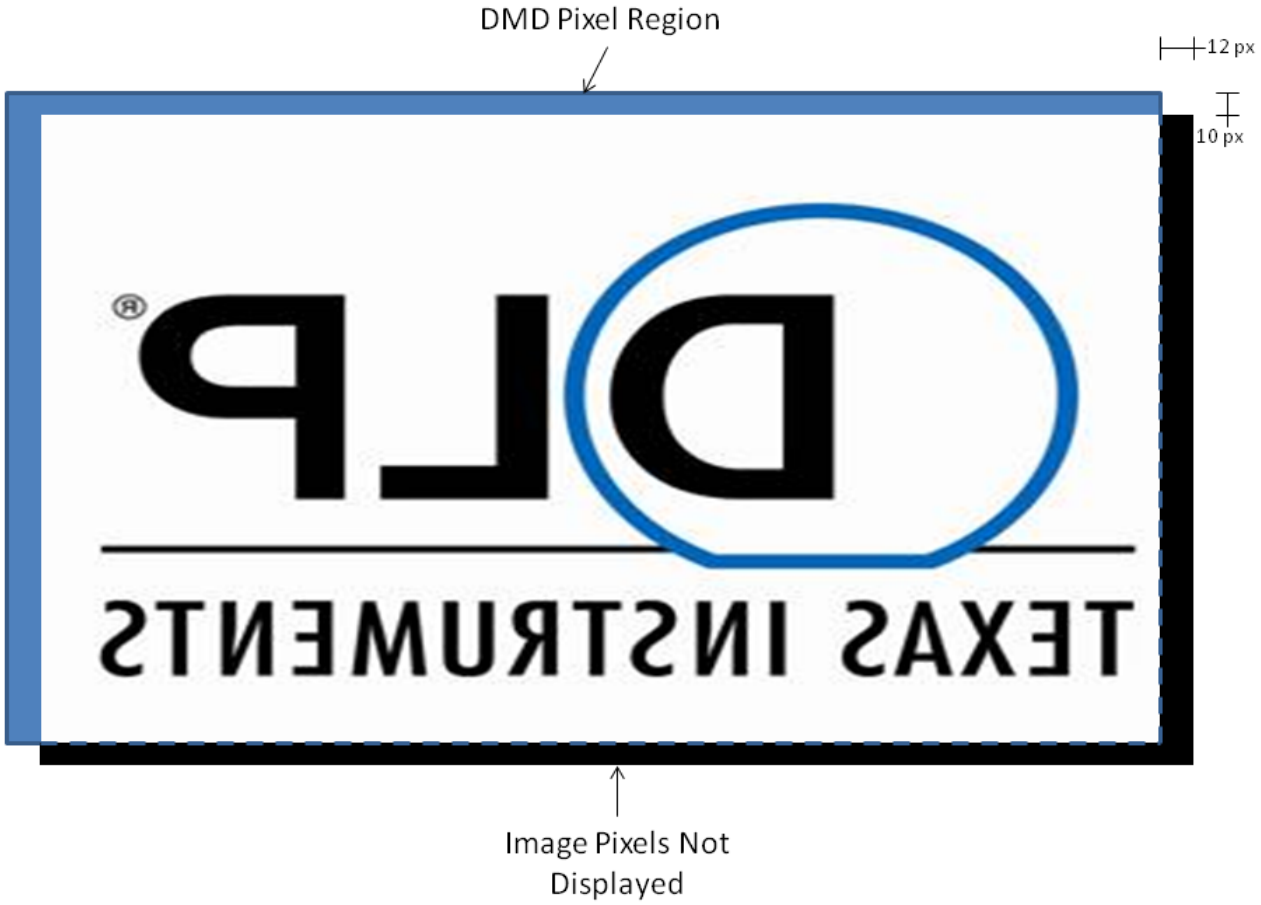


Figure 5-8. Bezel Adjust Example 3

Step 1: Horizontal offset

The horizontal pixels are reversed so movement right is considered negative.

Convert 12 to 11-bit binary	12 = 00000001100
Find the two's complement	11111110100

Step 2: Vertical offset

The vertical pixels are not reversed so movement down is still considered positive.

Convert 10 to 11-bit binary	10 = 00000001010

Step 3: Combine values

Combine values into 32-bit binary	00000**00000001010**00000**11111110100**
Convert to hex	0x000A07F4

Step 4: Write value to register 0x39

Write data	0x36 0x39 0x000A07F4

5.6.4 Video Frame Latency and Bezel Adjustment

The Bezel Adjustment feature requires one frame of processing to apply the new pixel positioning. By disabling this feature, the display latency can be reduced by one video frame. See below for frame timing diagrams of the latency with Bezel Adjustment both enabled and disabled.

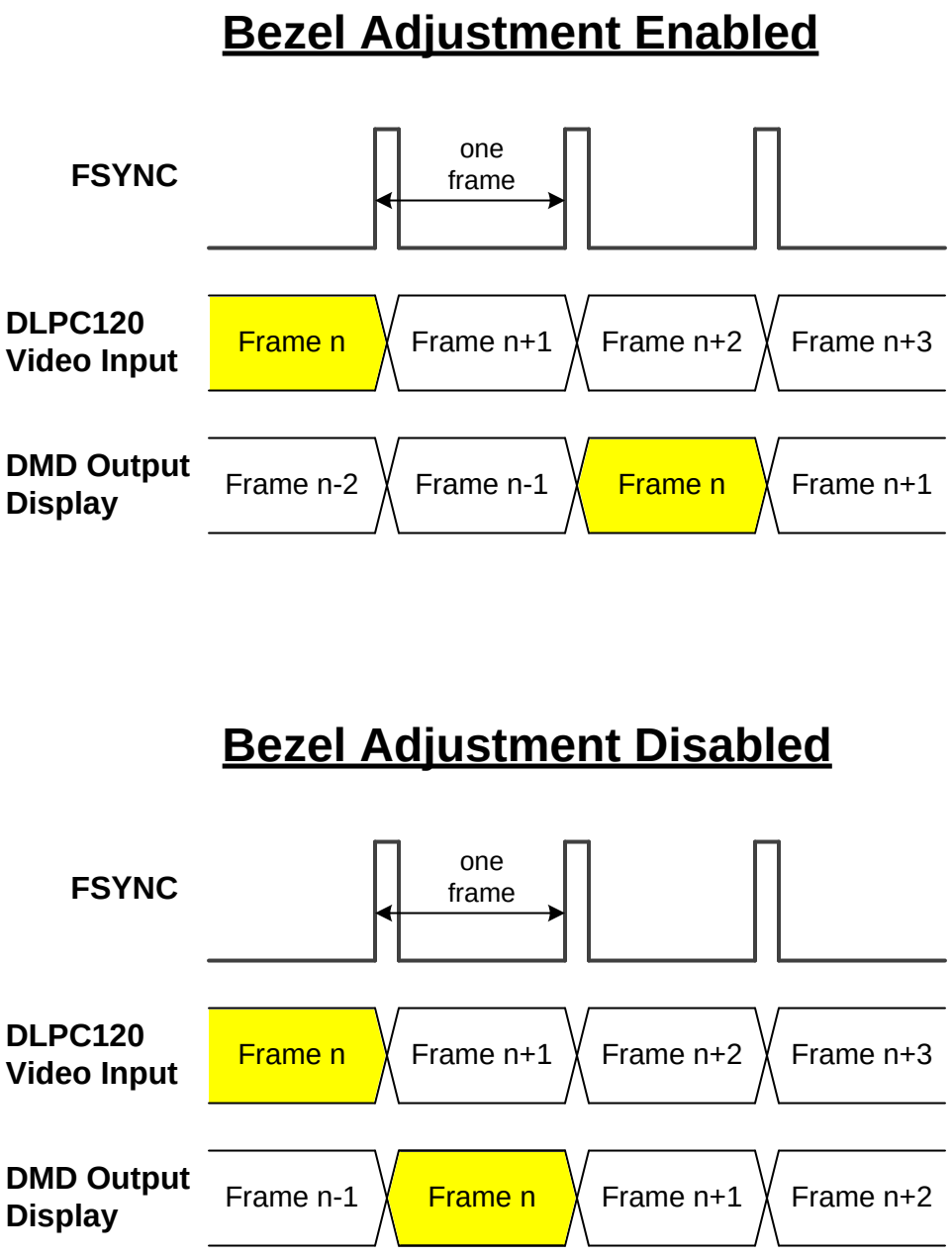


Figure 5-9. Frame Latency with Bezel Adjustment Enabled and Disabled

With the Bezel Adjustment feature enabled, there is a three frame delay for the offset parameters command to be reflected in the DMD image. See Figure 5-10 below for the timing diagram of the input command versus output display. This delay is relevant only if the offset command is applied dynamically after initialization. If the bezel offset commands are applied as part of the initialization routine, then the delay is not observed by the user.

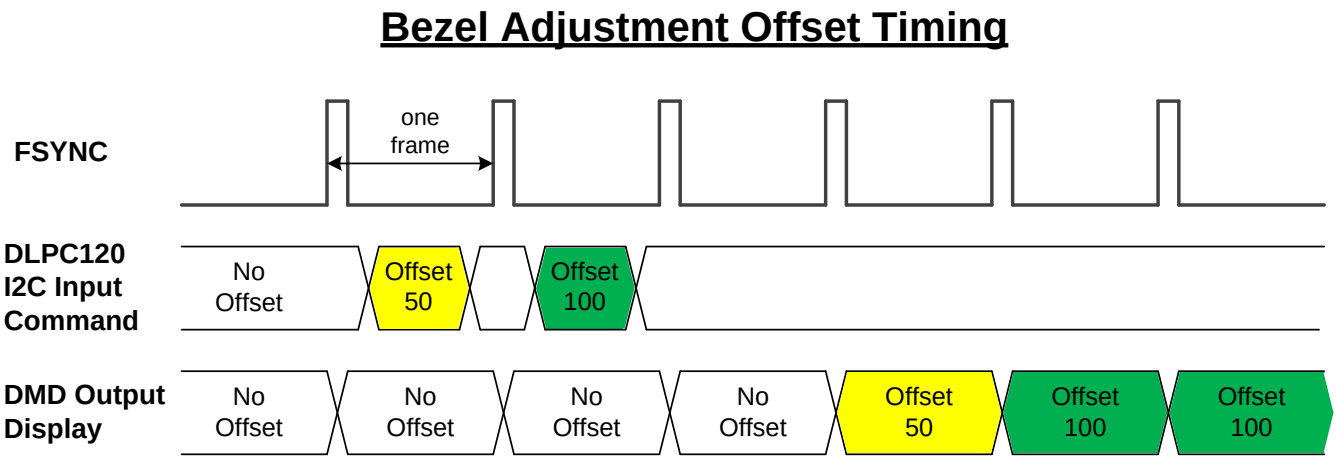


Figure 5-10. Bezel Adjustment Offset Command to DMD Display Frame Delay

5.7 RGB Data Bit Input Mappings

The DLPC120-Q1 supports 3 RGB data input mappings.

1. RGB888
2. RGB666
3. RGB565

By default the DLPC120-Q1 accepts RGB888 input data. RGB input modes are configured as command lists. See Section 1.2 for command list execution.

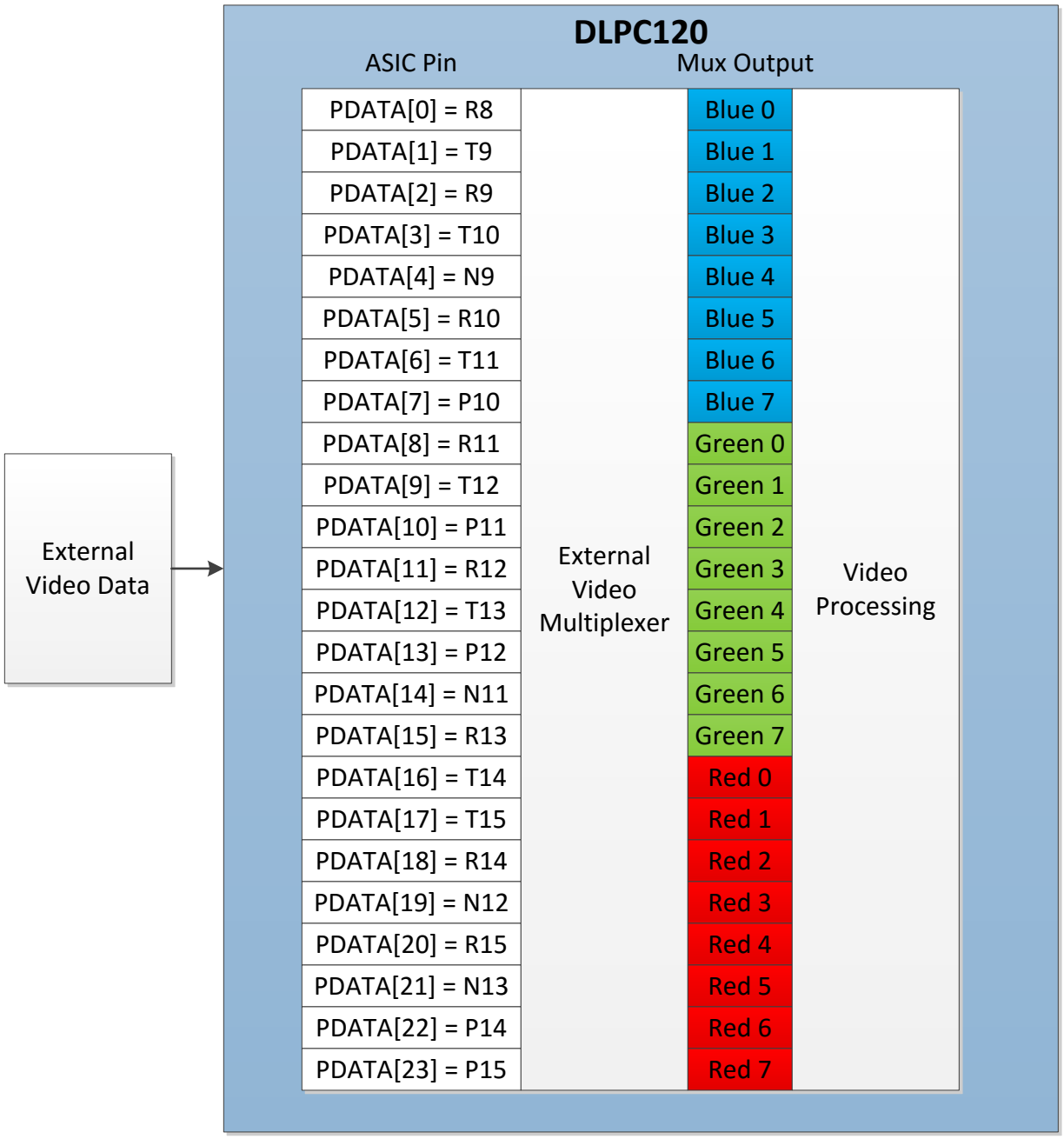


Figure 5-11. RGB888

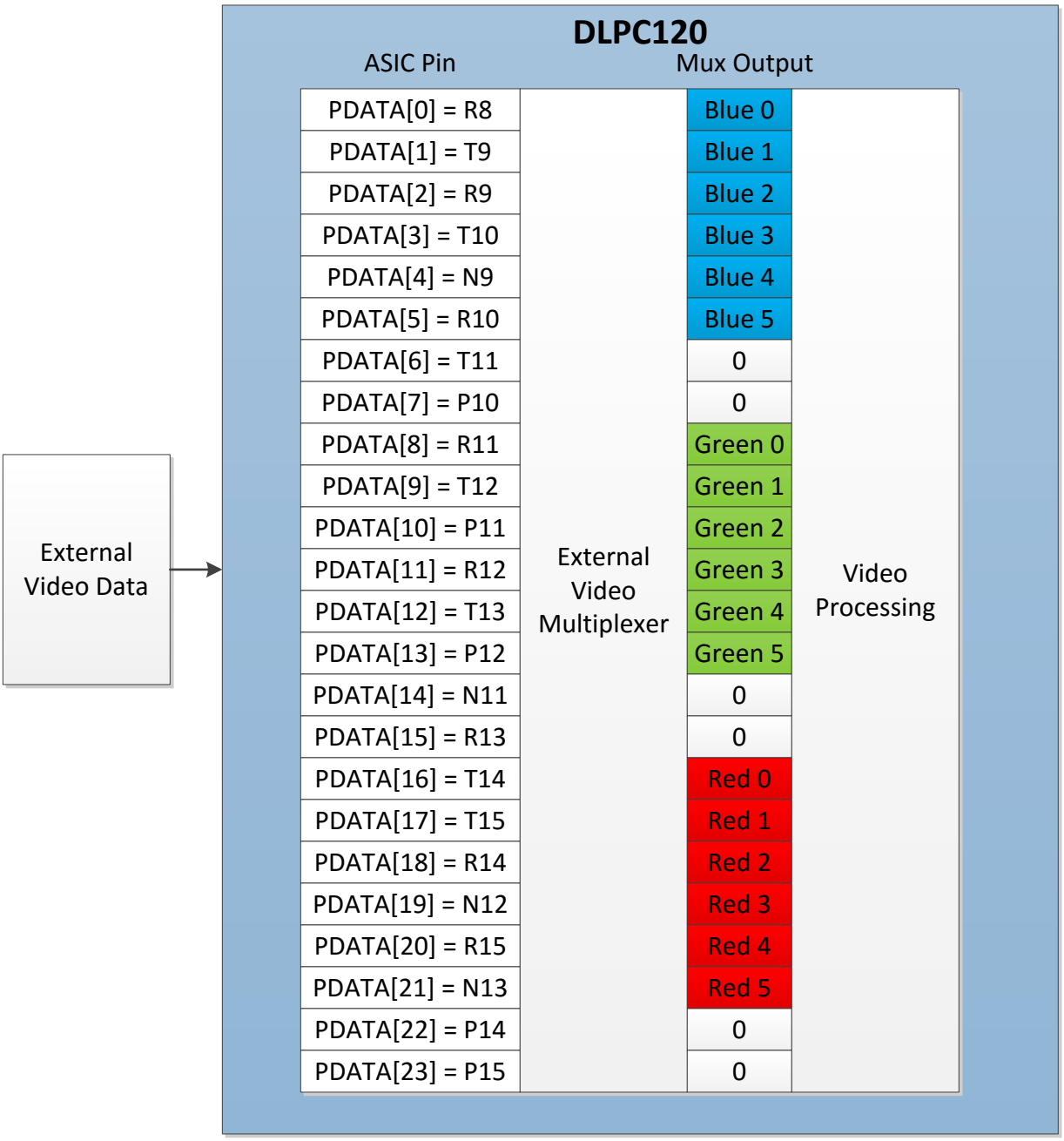


Figure 5-12. RGB666 MSB Aligned

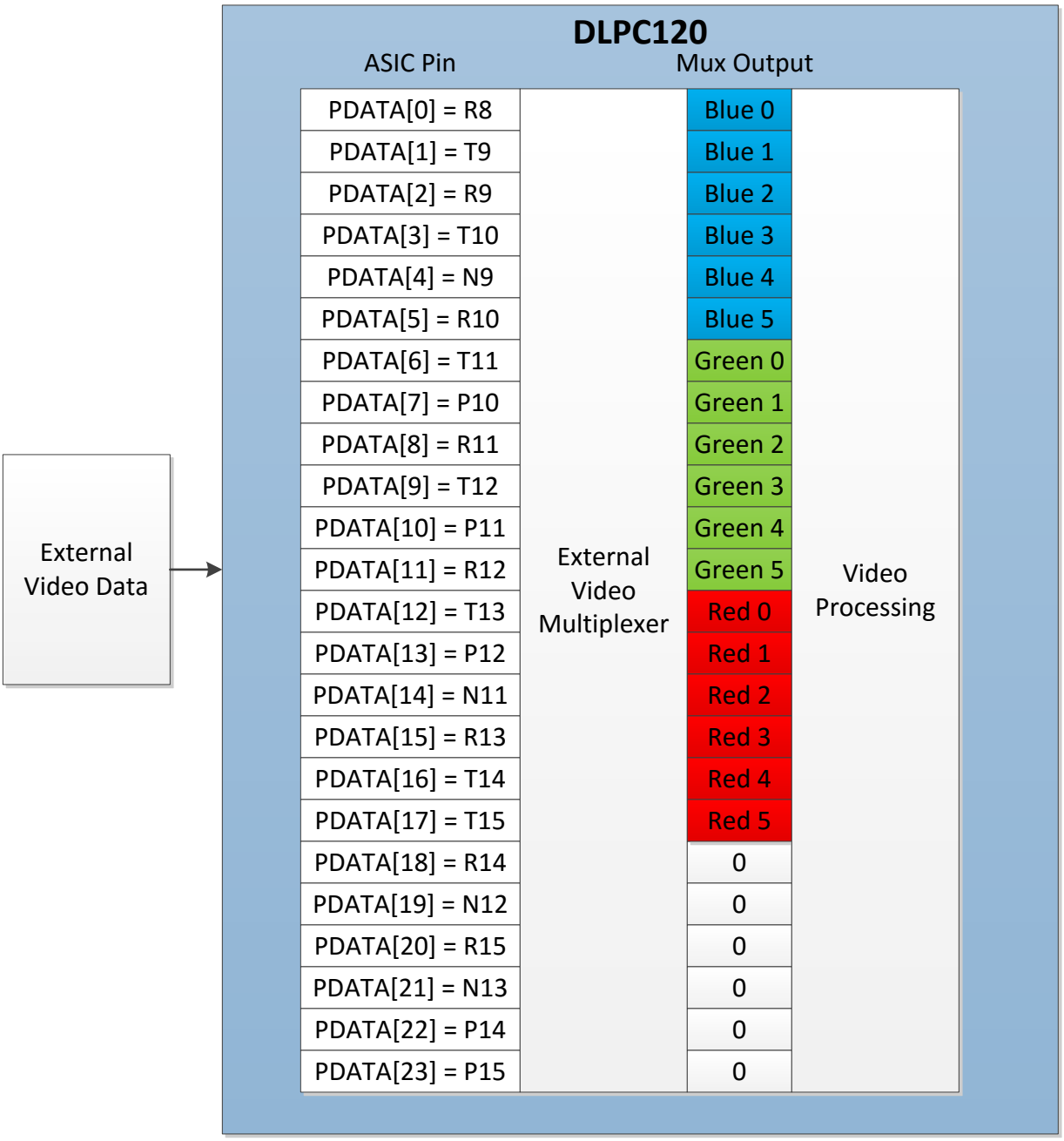


Figure 5-13. RGB666 Compressed

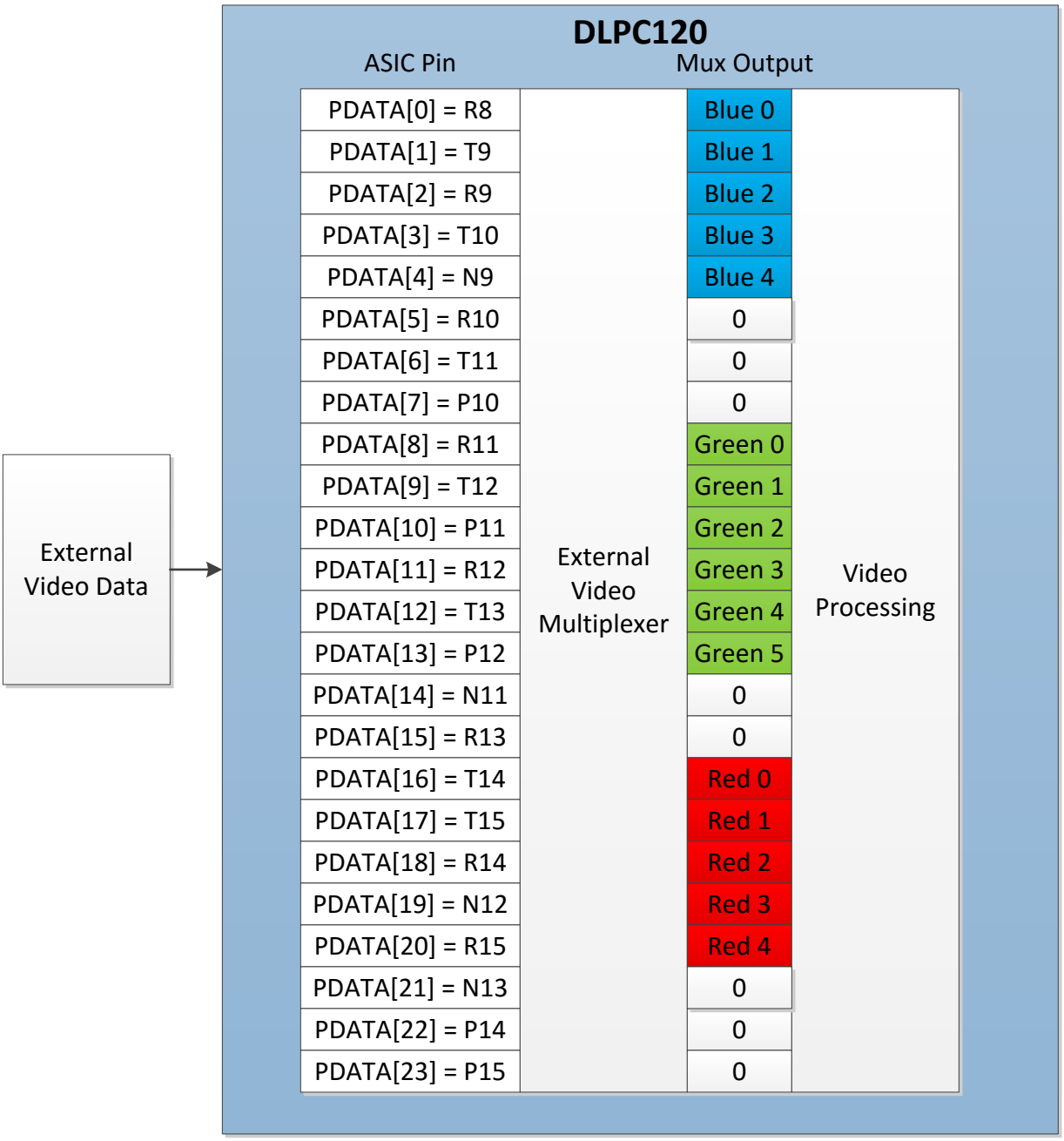


Figure 5-14. RGB565 MSB Aligned

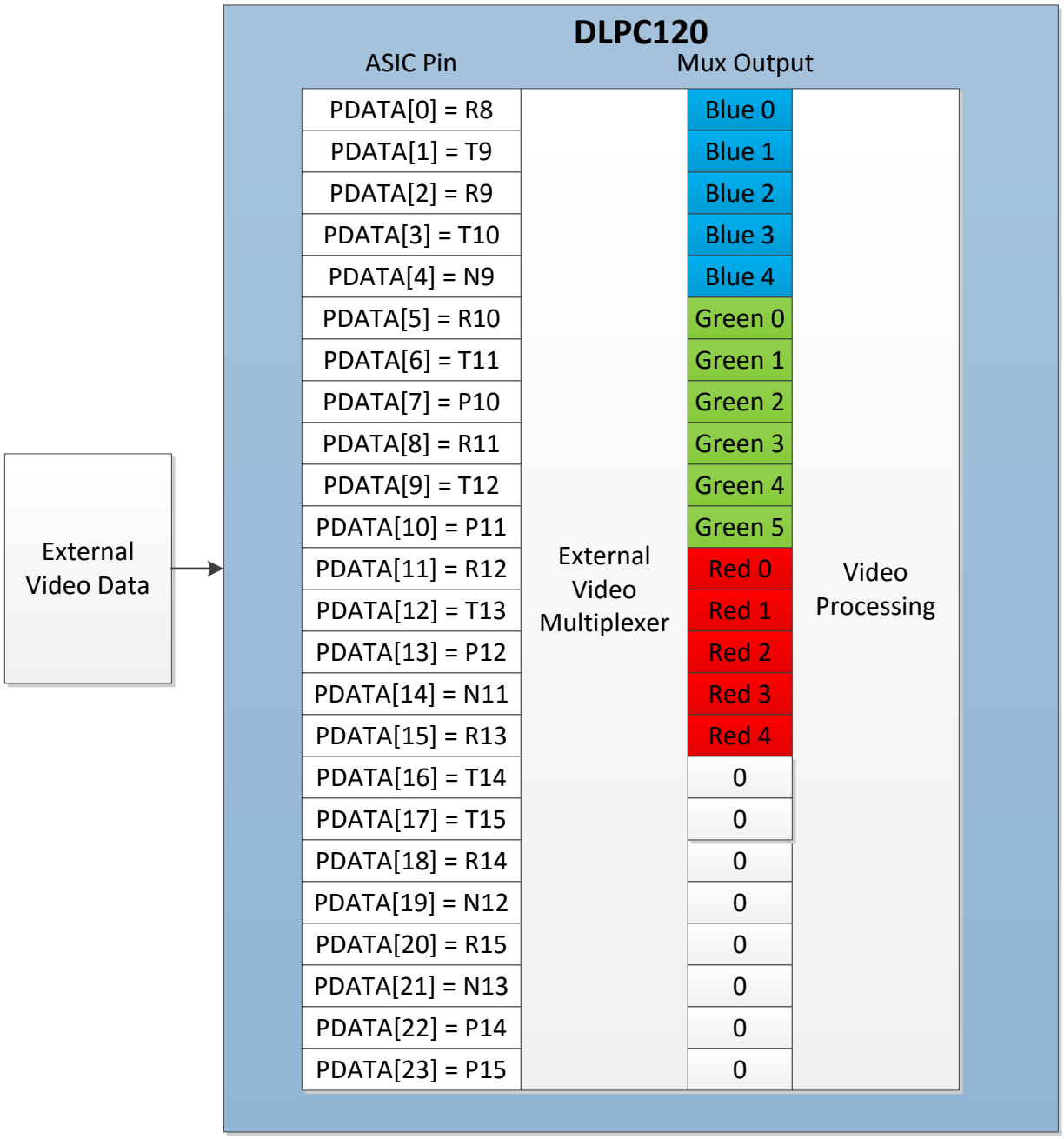


Figure 5-15. RGB565 Compressed

5.8 Video Adjustment Coefficient (VAC)

Table 5-6. VAC Control (Register 0x0D)

Bits	Description	Reset	Type	Notes
7:0	**VAC Coefficient**	0xFF	Write	Maximum digital input level from 0 to 255
8	**VAC Function Enable** 0: Disable 1: Enable	0x0	Write	
9	Reserved	0x0	Write	Set to 0

Table 5-6. VAC Control (Register 0x0D) (continued)

Bits	Description	Reset	Type	Notes
31:10	Unused	0x0	Write	

The DLPC120-Q1 supports digital attenuation of input data. This allows for additional dimming of the image with the tradeoff of image bit depth.

Red, Green, and Blue pixel data are input with a range from 0 to 255. When VAC is enabled the VAC coefficient is applied to each color channel of the pixel data using the following formula:

PixelDataOut = PixelDataIn(VAC/255)

5.8.1 Example: Set VAC Coefficient to 128

Step 1: Convert coefficient to hex

Convert to hex	128 = 0x80

Step 2: Write coefficient and function enable

Combine Enable and Coefficient	Bit 8 = 0x1 Bit 7:0 = 0x80 Value = 0x00000180
Write value to register	Write 0x00000180 to register 0x0D

5.9 Read Current Display Mode

Table 5-7. Display Mode (Register 0x8E)

Bits	Description	Reset	Type	Notes
1:0	**Display mode** States the video source type that the system is currently displaying. 0: External video 1: Test pattern 2: Splash screen	0x0	Read	
31:2	Unused	0x0	Read	

The current video source type that is being displayed can be determined by reading register 0x8E. Video source types are:

- External video mode
- Test pattern mode
- Splash screen mode

This function is used for two purposes in the TI reference Piccolo software:

- Check that external video is being displayed before executing Front End Video Test (refer to Section 7.6 for details on this test)
- Check that external video is being displayed to determine which results of the External Video Test are valid (refer to Section 7.7 for details on this test and the valid result conditions)

Temperature Functions

6.1 Temperature Function Overview

The DLPC120-Q1 connects via I^2C to a TMP411 temperature sensor to measure the DMD temperature and the TMP411 local temperature. The DLPC120-Q1 has built-in actions that will park and unpark the DMD at preset temperatures of the DMD array. See the DLP3030-Q1 datasheet for the operating temperature limits.

6.2 TMP411 Configuration

Several TMP411 configuration settings can be modified in default configuration. These settings must be configured immediately after a system reset so they cannot be performed after default configuration is executed by the ASIC. See a TI Applications engineer to request modifications to any of these settings listed below.

- TMP411 I^2C slave address
- TMP411 N-Factor
  - Set to 2 by default as recommended by TI

It is possible to use any version of the TMP411-Q1 sensor because the I^2C slave address is configurable in default configuration. The TI reference design uses version A which is set to the slave address 0x4C.

For more details on the TMP411, refer to the TMP411 datasheet.

6.3 DMD Temperature (Register 0x3C)

With a TMP411 properly connected to the DLPC120-Q1 and to the DMD temperature sensing diode, this register will report the DMD temperature. This value will update 8 times per second.

Table 6-1. DMD Temperature (Register 0x3C)

Bits	Description	Reset	Type	Notes
3:0	**DMD filtered temperature fraction**	0x0	Read	
11:4	**DMD filtered temperature integer**	0x0	Read	
15:12	Unused	0x0	Read	
27:16	Reserved	0x0	Read	Value may change during operation.
31:28	Unused	0x0	Read	

The values read from this register consist of two parts: an integer portion and a fractional portion.

1. The fractional portion is a single nibble ranging from 0_{dec} to 15_{dec}.
2. The integer portion is a value between -64_{dec} and 175_{dec}.

The formula for calculating the temperature:

Temperature = (Integer–64_{dec})+Fraction/16_{dec}

6.3.1 Example: Read temperature from a value of 0x018C018D

Step 1: Calculate the fraction from bits 3:0

Find value	0xC
Convert to decimal	0xC = 12
Divide by 16	12/16 = 0.75
Fraction =	0.75

Step 2: Calculate integer from bits 11:4

Find value	0x18
Convert to decimal	0x18 = 24
Subtract 64	24-64= -40
Integer =	-40

Step 3: Add values to find temperature in Celsius

Add fraction and integer	-40+0.8125 = -39.25
Temperature in Celsius =	-39.25°C

6.4 Temperature Status for TMP411 (Register 0x3D)

This register provides the local temperature of the TMP411 and also provides status bits for the TMP411. Temperature readings are taken 8 times per second.

Table 6-2. Temperature Status (Register 0x3D)

Bits	Description	Reset	Type	Notes
3:0	**TMP411 raw temperature fraction**	0x0	Read	
11:4	**TMP411 raw temperature integer**	0x0	Read	
12	**Temperature Flag** 1: Reading is valid 0: Reading is invalid	0x0	Read	
13	**I²C Temperature Reading Flag** 1: Reading is valid 0: Reading is invalid	0x0	Read	
15:14	Unused	0x0	Read	
27:16	Reserved	0x0	Read	Value may change during operation.
31:28	Unused	0x0	Read	

The temperature calculations follow the same procedure as shown in Section 6.3.

The Temperature Pass/Fail Flag (bit 12) indicates whether the TMP411 is configured properly. If this flag reads 0, the temperatures stored in this register are invalid.

The I²C Temperature Reading Flag (bit 13) indicates whether the DLPC is successfully communicating with the TMP411. If this flag reads 0, the temperatures stored in this register are invalid.

See TMP411 datasheet for additional specifications.

Built-In Self Tests

7.1 Built-In Self Test Overview

The ASIC has 6 Built-In Self Tests. These tests are optional and exist to verify certain functionality within the ASIC. If any Built-In Self Test is executed, it must be executed in a specific order due to operational requirements for each test. Figure 7-1 below shows the correct order for BIST execution. After UMC initialization is completed (see Section 3.3), no video output should be selected before preparation for the System BIST.

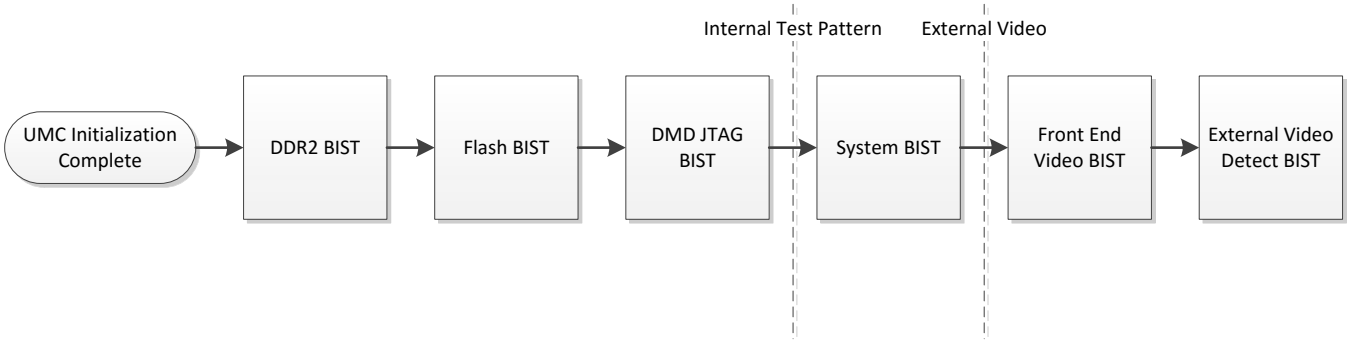


Figure 7-1. BIST Order

The following registers are used for self tests:

Table 7-1. Hardware Tests Enable (Register 0x50)

Bit	Description	Reset	Type	Notes
0	Unused	0x0	Write	
1	**System BIST Enable**	0x0	Write	
2	**Flash BIST Enable**	0x0	Write	
3	**DDR2 BIST Enable**	0x0	Write	
4	**Video Detect Test Enable**	0x0	Write	
5	**Front End Video Test Enable**	0x0	Write	
6	**DMD JTAG Test Enable**	0x0	Write	
7	**DDR2 BIST Test Type** 1: Long version 0: Short version	0x0	Write	Long version runs the short version multiple times
31:8	Unused	0x0	Write	

Table 7-2. Hardware Tests Status (Register 0x51)

Bit	Description	Reset	Type	Notes
0	Unused	0x0	Read	
1	**System BIST Valid** 1: System BIST is finished and test result is valid 0: System BIST result is not valid	0x0	Read	

Table 7-2. Hardware Tests Status (Register 0x51) (continued)

Bit	Description	Reset	Type	Notes
2	**Flash BIST Valid** 1: Flash BIST is finished and test result is valid 0: Flash BIST result is not valid	0x0	Read	
3	**DDR2 BIST Valid** 1: DDR2 BIST is finished and test result is valid 0: DDR2 BIST result is not valid	0x0	Read	
4	**Video Detect Test Valid** 1: Video detect test is finished and test result is valid 0: Video detect test result is not valid	0x0	Read	
5	**Front End Video Test Valid** 1: Front end video test is finished and test result is valid 0: Front end video test result is not valid	0x0	Read	
6	**DMD JTAG Test Valid** 1: DMD JTAG test is finished and test result is valid 0: DMD JTAG test result is not valid	0x0	Read	
17:7	Unused	0x0	Read	
18	**Flash BIST Status** 1: Flash BIST test passed 0: Flash BIST test failed	0x0	Read	
19	**DDR2 BIST Status** 1: DDR2 BIST test passed 0: DDR2 BIST test failed	0x0	Read	
20	**Video Detect Test Status** 1: Video detect test passed 0: Video detect test failed	0x0	Read	
21	Unused	0x0	Read	
22	**DMD JTAG Test Status** 1: DMD JTAG test passed 0: DMD JTAG test failed	0x0	Read	
31:23	Unused	0x0	Read	

Table 7-3. System BIST Checksum (Register 0x52)

Bit	Description	Reset	Type	Notes
31:0	**System BIST checksum value**	0x0	Read	

Table 7-4. Flash BIST Checksum (Register 0x53)

Bit	Description	Reset	Type	Notes
31:0	**Flash BIST checksum value**	0x0	Read	

Table 7-5. Video Detect Vsync Maximum (Register 0x54)

Bit	Description	Reset	Type	Notes
23:0	**Vsync max period**	0xEE0980	Write	Number of 78-MHz clock cycles
27:24	Unused	0x0	Write	
28	**Command List Execute** 0: Do not execute video detect command list 1: Execute video detect command list Pass or Fail	0x0	Write	
31:29	**Video Detect Mux Select** Specifies what result to be reported 000: Vsync count in # 78-MHz clock cycles 001: Pclk count in # of pclk clock cycles 010: Number of active lines in a frame 011: Number of pixels in first active line of frame	0x0	Write	

Table 7-6. Video Detect Vsync Minimum (Register 0x55)

Bit	Description	Reset	Type	Notes
23:0	**Vsync min period**	0x01298B	Write	Number of 78-MHz clock cycles
31:24	Unused	0x0	Write	

Table 7-7. Video Detect Pclk Configuration (Register 0x56)

Bit	Description	Reset	Type	Notes
31:0	**Vsync Pclk configuration** Set to 0x3FFDBC02 before external video BIST. This configures the BIST to only pass if a PCLK signal is detected in the video signal.	0x3FFDBC0D	Write	BIST may fail if value is not set properly before BIST execution

Table 7-8. Video Detect Active Lines Configuration (Register 0x57)

Bit	Description	Reset	Type	Notes
31:0	**Vsync Active Lines configuration** Set to 0x08000001 before external video BIST. This configures the BIST to only pass if at least one active line is detected in the video signal.	0x08000002	Write	BIST may fail if value is not set properly before BIST execution

Table 7-9. Video Detect Active Pixels Configuration (Register 0x58)

Bit	Description	Reset	Type	Notes
31:0	**Vsync Active Lines configuration** Set to 0x0FFF0001 before external video BIST. This configures the BIST to only pass if at least one active pixel is detected in the first line of the video signal.	0x0	Write	BIST may fail if value is not set properly before BIST execution

Table 7-10. Video Detect Fail Command List Address (Register 0x59)

Bit	Description	Reset	Type	Notes
24:0	**Video detect fail command list address** Base address in flash memory Will not trigger unless bit 28 of register 0x54 is set	0x0	Write	
31:25	Unused	0x0	Write	

Table 7-11. Video Detect Pass Command List Address (Register 0x5A)

Bit	Description	Reset	Type	Notes
24:0	**Video detect pass command list address** Base address in flash memory Will not trigger unless bit 28 of register 0x54 is set	0x0	Write	
31:25	Unused	0x0	Write	

Table 7-12. Video Detect Status (Register 0x5B)

Bit	Description	Reset	Type	Notes
0	**Vsync found error** 1: Error – Vsync is not found within the Vsync max period 0: Vsync is found within max period	0x0	Read	
1	**Vsync out of range error** 1: Error – Vsync period is out of range 0: Vsync period is in the specified range	0x0	Read	Range: 57.4 Hz – 62.6 Hz

Table 7-12. Video Detect Status (Register 0x5B) (continued)

Bit	Description	Reset	Type	Notes
2	**Pixel clock found error** 1: Error – Pixel Clock is not found within pixel clock test period 0: Pixel clock found within pixel clock test period	0x0	Read	
3	**Pixel clock out of range error** 1: Error – Pixel clock count is out of range 0: Pixel count is in the specified range	0x0	Read	
4	**Active lines out of range error** 1: Error – Number of active lines per frame is out of range 0: Number of active lines per frame is in the specified range	0x0	Read	
5	**Pixels per line out of range error** 1: Error – Number of pixels per line is out of range 0: Number of pixels per line is in the specified range	0x0	Read	
31:6	Unused	0x0	Read	

Table 7-13. Video Detect Report (Register 0x5C)

Bit	Description	Reset	Type	Notes
23:0	**Video Detect Report** Value is determined by mux in register 0x54	0x0	Read	
31:24	Unused	0x0	Read	

Table 7-14. Front End Video Test Start Position (Register 0x5D)

Bit	Description	Reset	Type	Notes
10:0	**Front end video test start column**	0x0	Write	0-based
15:11	Unused	0x0	Write	
26:16	**Front end video test start row**	0x0	Write	0-based
31:27	Unused	0x0	Write	

Table 7-15. Front End Video Test End Position (Register 0x5E)

Bit	Description	Reset	Type	Notes
10:0	**Front end video test end column**	0x0	Write	0-based
15:11	Unused	0x0	Write	
26:16	**Front end video test end row**	0x0	Write	0-based
31:27	Unused	0x0	Write	

Table 7-16. Front End Video Test Checksum (Register 0x5F)

Bit	Description	Reset	Type	Notes
31:0	**Front end video checksum value**	0x01234567	Read	

Table 7-17. Flash BIST Start Address (Register 0x61)

Bit	Description	Reset	Type	Notes
24:0	**Flash BIST checksum start address**	0x0	Write	
31:25	Unused	0x0	Write	

Table 7-18. Flash BIST Byte Count (Register 0x62)

Bit	Description	Reset	Type	Notes
24:0	**Number of bytes to checksum from starting address**	0x100	Write	
31:25	Unused	0x0	Write	

Table 7-19. DMD Device ID (Register 0x64)

Bit	Description	Reset	Type	Notes
31:0	**DMD Device ID**	0x0	Read	Value will be populated after DMD JTAG Test is performed.

Figure 7-2 below indicates where each test is performed.

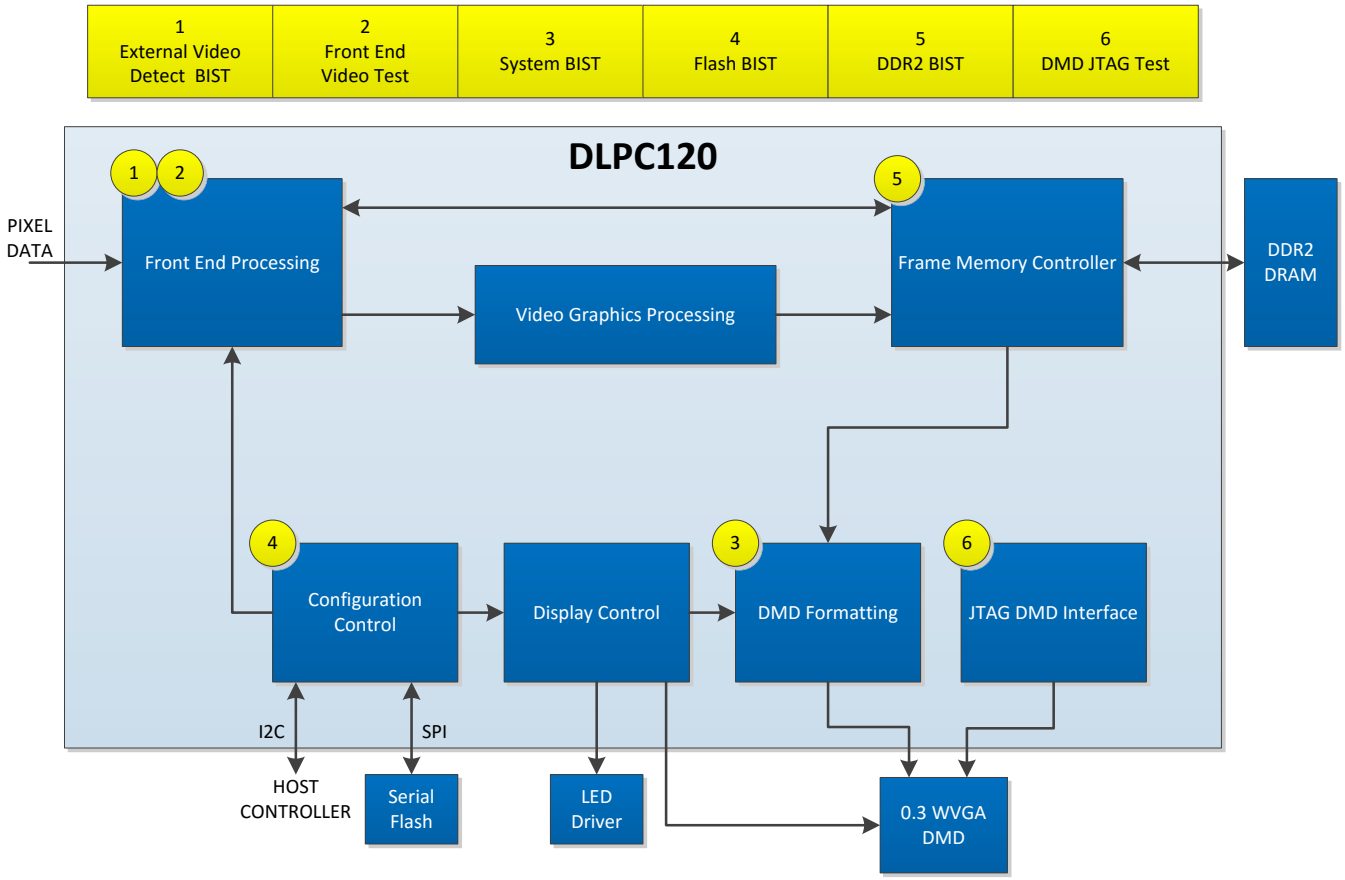


Figure 7-2. Hardware Test Locations

7.2 DDR2 Built-In Self Test

This test checks that the DDR memory is functioning correctly and can be accessed correctly. This test must be run before an input mode command list is selected and output.

The following registers are used for the DDR2 Built-In Self Test. These registers are described in more detail in Chapter 7:

Table 7-20. DDR2 Built-In Self Test Registers

Register	Address
Hardware Tests Enable	0x50
Hardware Tests Status	0x51

The flow chart below indicates the procedure for running this test:

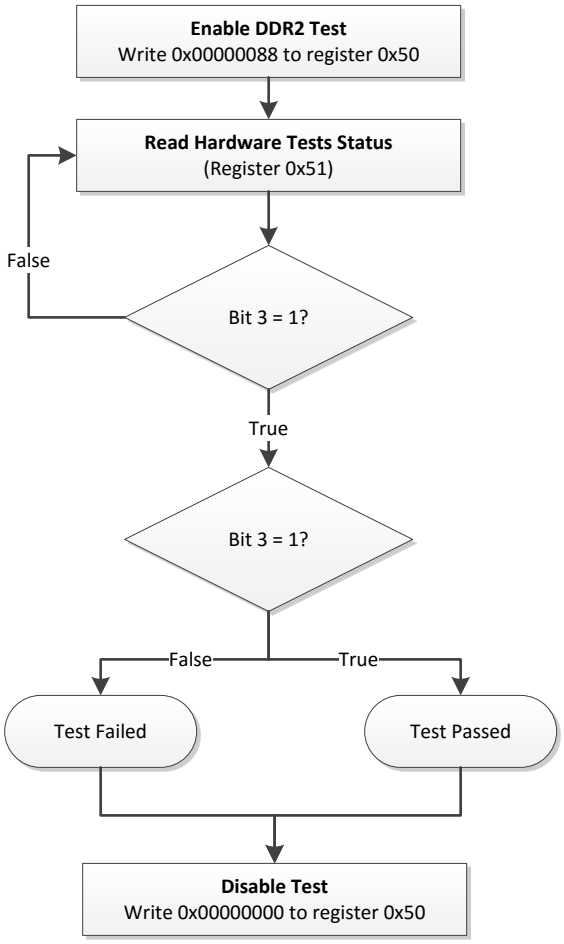


Figure 7-3. DDR2 Test

7.3 Flash Built-In Self Test

This test uses a specified range of flash memory data to generate a checksum result. The checksum can be compared from system to system to verify the flash integrity.

The following registers are used for the Flash Built-In Self Test. These registers are described in more detail in Chapter 7:

Table 7-21. Flash Built-In Self Test Registers

Register	Address
Hardware Tests Enable	0x50
Hardware Tests Status	0x51
Flash BIST Checksum	0x53
Flash BIST Start Address	0x61
Flash BIST Byte Count	0x62

The flow chart below indicates the procedure for running this test:

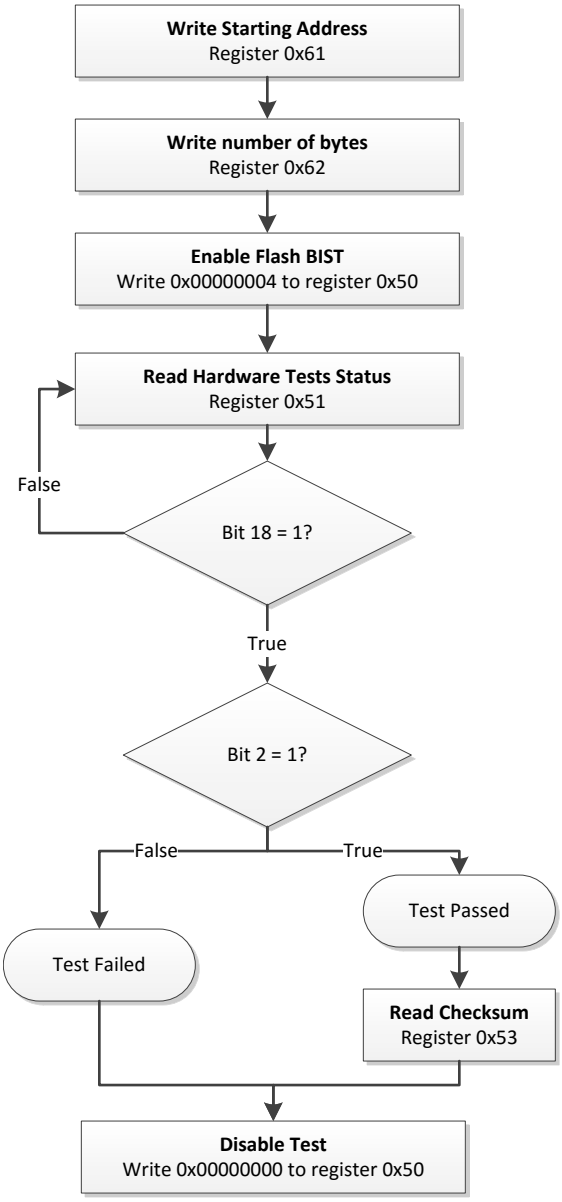


Figure 7-4. Flash BIST Procedure

Note that the Valid (bit 2) and Status (bit 18) bit-checking is reversed for the Flash BIST compared to the other BISTs. It is important that the Status bit is read first for this BIST.

7.4 DMD JTAG Test

This test can detect connection faults between the DLPC120-Q1 and the DMD. This test must be run before an input mode command list is selected and output.

The following registers are used for the DMD JTAG Test. These registers are described in more detail in Chapter 7:

Table 7-22. DMD JTAG Test Registers

Register	Address
Hardware Tests Enable	0x50
Hardware Tests Status	0x51
DMD Device ID	0x64

The flow chart below indicates the procedure for running this test:

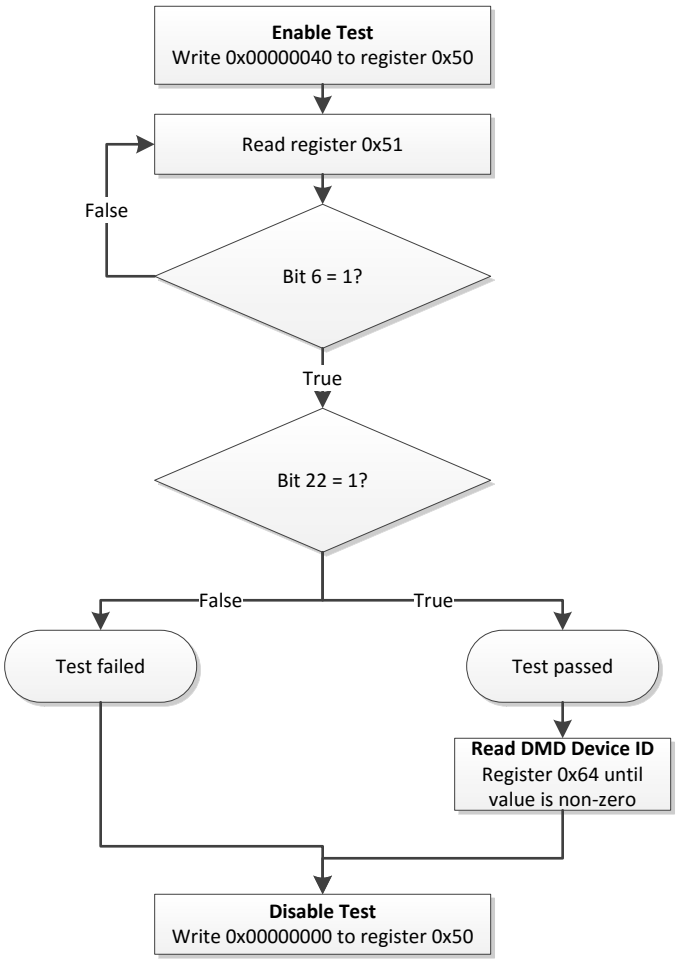


Figure 7-5. DMD JTAG Test Procedure

7.5 System Built-In Self Test

This test generates a checksum result using processed display data at the point just before it is sent to the DMD. This checksum can be compared from system to system. The same input image must be used for each test. Two unique checksums will output from any given input image so both checksums should be acknowledged as correct values.

The following registers are used for the System Built-In Self Test. These registers are described in more detail in Chapter 7:

Table 7-23. System Built-In Self Test Registers

Register	Address
Hardware Tests Enable	0x50
Hardware Tests Status	0x51

Table 7-23. System Built-In Self Test Registers (continued)

System BIST Checksum	0x52

The flow chart below indicates the procedure for running this test:

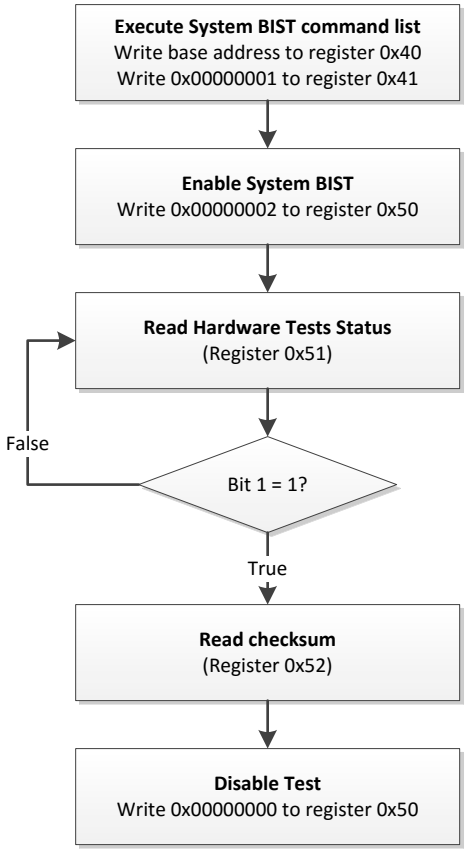


Figure 7-6. System BIST Procedure

7.6 Front End Video Test

This test generates a checksum result using the data read directly from an external video source. The checksum can be compared from system to system to verify the video source. A static image must be displayed from an external video source in order to generate a consistent checksum on each system.

The start pixel location and end pixel location must be written to registers 0x5D and 0x5E.

Figure 7-7 defines the coordinate space for these pixel locations with respect to the active video region of the DMD.

The following registers are used for the Front End Video Test. These registers are described in more detail in Chapter 7:

Table 7-24. Front End Video Test Registers

Register	Address
Hardware Tests Enable	0x50
Hardware Tests Status	0x51
Front End Video Test Start Position	0x5D
Front End Video Test End Position	0x5E

Table 7-24. Front End Video Test Registers (continued)

Front End Video Test Checksum	0x5F

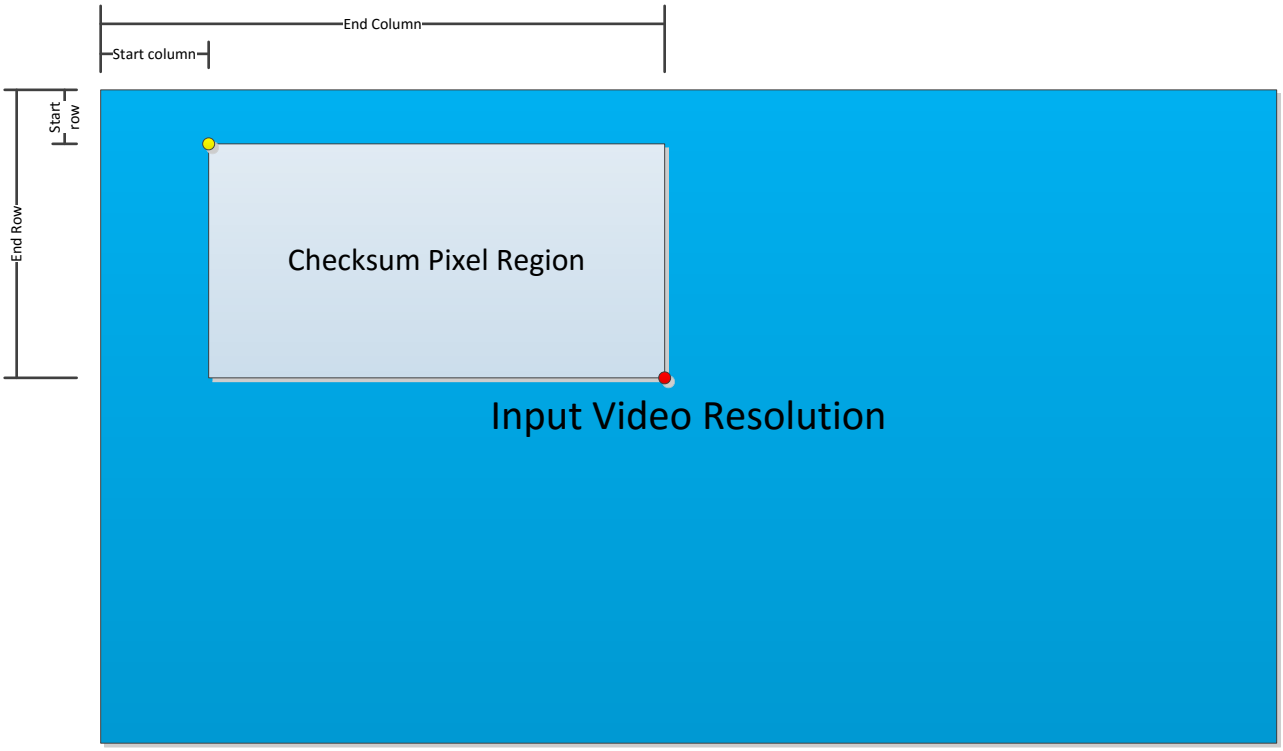


Figure 7-7. Front End Checksum Region

Figure 7-8 shows the procedure for running this test.

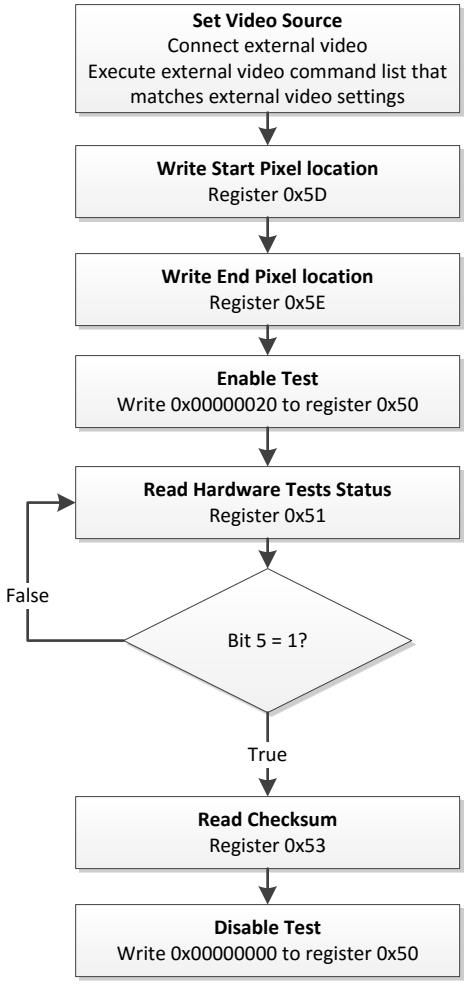


Figure 7-8. Front End Video Test Procedure

7.7 External Video Detect Test

A video detect test can be run to confirm that the external Vsync is within a specified range. Additionally, this test verifies that there is a nonzero pixel clock, input lines, and pixels per line.

This test can be performed at any time during system operation after Power-On initialization (as specified in Chapter 3).

The following registers are used for the External Video Detect Test. These registers are described in more detail in Chapter 7:

Table 7-25. External Video Detect Test Registers

Register	Address
Hardware Tests Enable	0x50
Hardware Tests Status	0x51
Video Detect Vsync Maximum	0x54
Video Detect Vsync Minimum	0x55
Video Detect Pclk Configuration	0x56
Video Detect Active Lines Configuration	0x57
Video Detect Active Pixels Configuration	0x58
Video Detect Fail Command List Address	0x59

Table 7-25. External Video Detect Test Registers (continued)

Video Detect Pass Command List Address	0x5A
Video Detect Status	0x5B
Video Detect Report	0x5C

Table 7-26. Video Detect Test Parameter Ranges

Parameter	Minimum	Maximum
Vsync	Determined by bits 23:0 of register 0x55	Determined by bits 23:0 of register 0x54

Figure 7-9 shows how active lines and active pixels per line are defined.

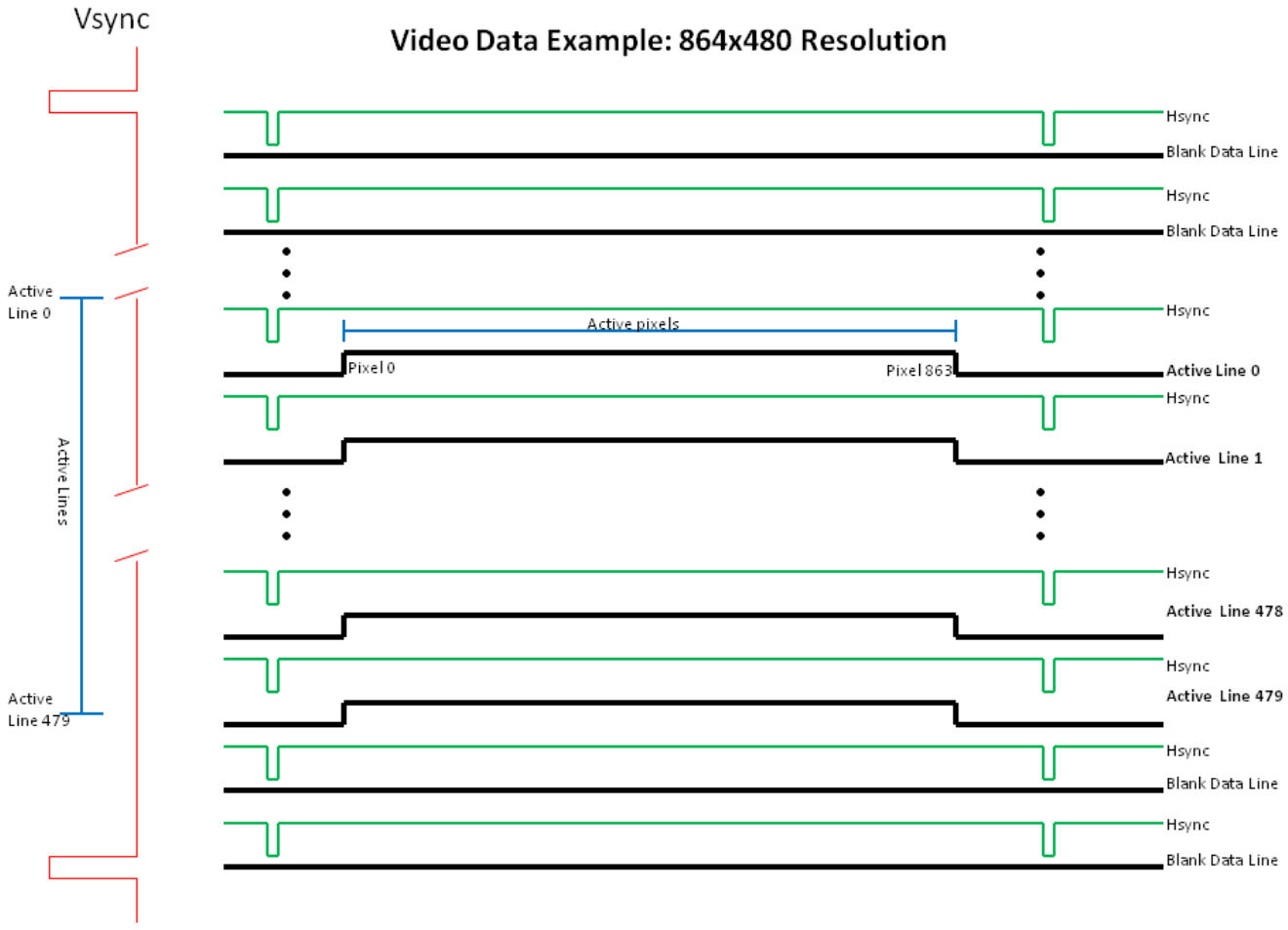


Figure 7-9. Video Data Lines Example

7.7.1 Use Case

The External Video Detect Test may be used to detect the lack of a video signal source.

If configured to do so, the DLPC120-Q1 will execute a pre-defined command list if video signal is not found and the test fails. For example, the DLPC120-Q1 can be configured to execute an internal test pattern command list, such as a Solid Black image, when the test fails.

Note: If the DLPC120-Q1 is in external video mode and a video signal is not being passed to the DLPC120-Q1 then the DMD is in an unsafe state. For this reason the External Video Detect Test should be run any time the DLPC120-Q1 is set to external video mode and a command list should be set to execute if the test fails in order to switch the DMD to a safe operating state.

The vsync range that is considered a “pass” condition is configurable and should be set based on the expected frame sync jitter from the video source. For example, the video source is intended to generate frame sync jitter from 58 Hz to 60 Hz then these values should be used as the vsync maximum and minimum for the test configuration.

7.7.2 Setting up registers

In order to run the video detect test, the values below must be configured:

1. Set register 0x56 to 0x3FFDBC02. This configures the BIST to only pass if a PCLK signal is detected in the video signal.
2. Set register 0x57 to 0x08000001. This configures the BIST to only pass if at least one active line is detected in the video signal.
3. Set register 0x58 to 0x0FFF0001. This configures the BIST to only pass if at least one active pixel is detected in the first line of the video signal.
4. Maximum vsync period that should be allowed (register 0x54)
5. Minimum vsync period that should be allowed (register 0x55)
6. Whether a command list should be executed when the test fails (register 0x54)
    1. Test PASS command list base address in flash (register 0x5A)
    2. Test FAIL command list base address in flash (register 0x59)
7. Video Detect Report mux select (register 0x54)
    1. This value determines what the Video Detect report provides. It can provide: vsync, pixel clock, active lines, or active pixels

7.7.3 Example: Setting up Video Detect Test registers

- External video is 60 Hz so test should allow 58.8 Hz to 61.2 Hz
- Command list for external video 864x480 should operate when the test PASSES
- Command list for splash screen should operate when the test FAILS
- Video Detect should report vsync value when it completes

Step 1: Set configuration values

Set Pclk configuration	Register 0x56 = 0x3FFDBC02
Set Active lines configuration	Register 0x57 = 0x08000001
Set Active pixels configuration	Register 0x58 = 0x0FFF0001

Step 2: Calculate Vsync minimum period

Calculate 61.2 Hz period in 78-MHz clocks	78 MHz / 61.2 Hz = 1274510 clocks
Convert clocks to hex	Vsync min = 1274510 = 0x13728E

Step 3: Calculate Vsync maximum period

Calculate 58.8 Hz period in 78-MHz clocks	78 MHz / 58.8 Hz = 1326531 clocks
Convert clocks to hex	Vsync max = 1326531 = 0x143DC3

Step 4: Set Test command list base addresses

Determine command list base address for Input 864x480 command list from .cfg file for app flash file	Pass Base address = 0x123456(example)
Determine command list base address for splash image command list from .cfg file for app flash file	Fail Base address = 0x789ABC (example)

Step 5: Set Video Detect Report mux to vsync

From details of register 0x54, mux should be set to 000 in order to report vsync	Mux = 000 = 0x0

Step 6: Set Registers with found values

From details of register 0x54: combine vsync max period, command list enable, and mux settings	Command list enable = 0x1 Mux = 0x0 Vsync max = 0x143DC3 Register 0x54 = 0x1F143DC3
From details of register 0x55: write vsync min period	Register 0x55 = 0x0013728E
From details of register 0x59: write command list FAIL address	Register 0x59 = 0x00789ABC
From details of register 0x5A: write command list PASS address	Register 0x5A = 0x00123456

7.7.4 Operating Procedure

Figure 7-10 indicates the procedure for running this test.

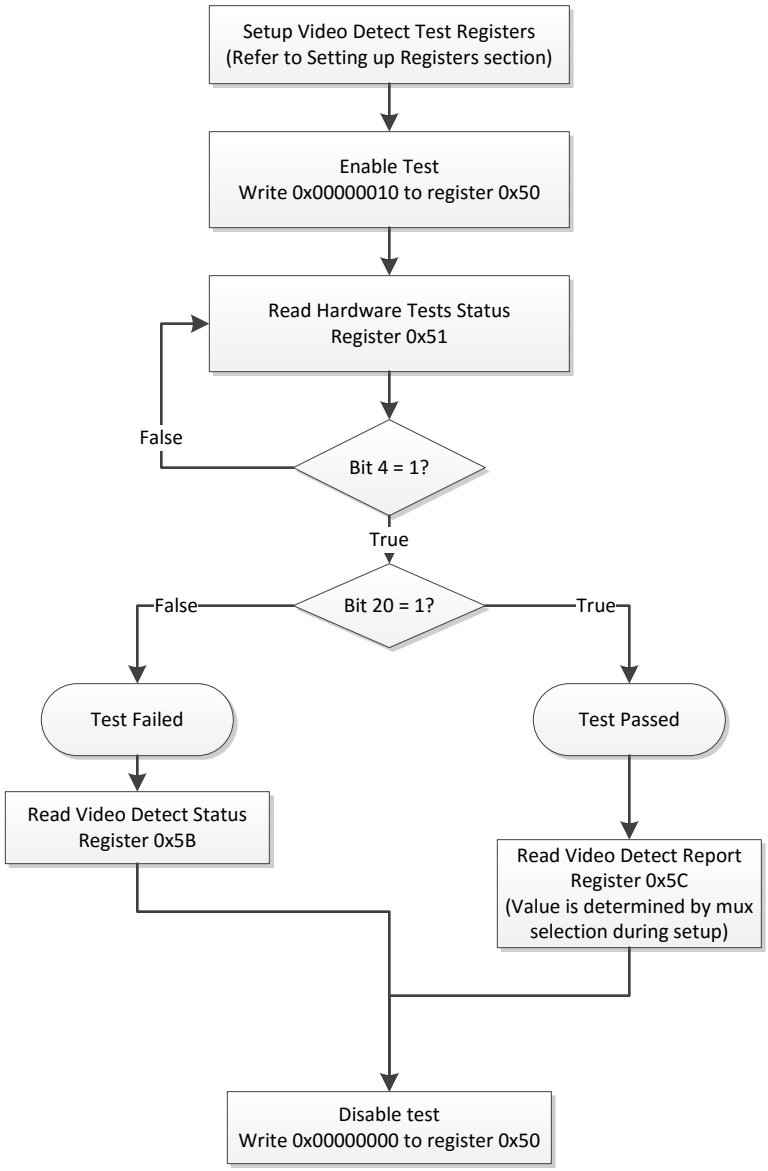


Figure 7-10. External Video Detect Test Procedure

7.7.5 Reading Video Detect Report Mux (0x5C)

The Video Detect Report can provide a value for vsync, pixel clock, active lines, and active pixels in the input video. The chart below indicates when this information is valid:

Table 7-27. Valid Video Detect Test Parameter Results

Ext Video Signal Connected	Displaying	Vsync	pixel clock	Lines	Pixels
YES	Video	VALID	VALID	VALID	VALID
YES	Internal pattern/Splash	VALID	INVALID	INVALID	INVALID
NO	Internal pattern/Splash	VALID	INVALID	INVALID	INVALID

Vsync Mux = 000_2	The report provides the number of 78-MHz clocks in the vsync period
Pixel Clock Mux = 001_2	The report provides the number of pixel clocks that were found in a 13.1154 µs period
Active Lines Mux = 010_2	The report provides the exact number of lines in an active frame
Active Pixels Mux = 011_2	The report provides the exact number of pixels in the first active line of an active frame

Note: PLL spreading configuration (Section 3.2) will affect these results. For example, with a 2% PLL clock down-spreading, the Vsync value should be expected to decrease roughly 1% on average.

Flash Programming

8.1 Flash Programming Overview

The flash programming registers provide access to the flash memory through the DLPC120-Q1 I^2C interface. Note that it is also possible to program the flash memory via its SPI interface, in which case the DLPC120-Q1 shall be held in reset.

There are 5 commands that are used for flash access via the DLPC120-Q1:

1. Write Enable
2. Sector Erase
3. Page Write
4. Read Flash Memory
5. Read Internal Flash Status

Write Enable

Must be performed before any modifications can be made to flash memory. Modifications include Sector Erase and Page Write commands.

Sector Erase

Erases one sector of flash memory. The address must be a multiple of the sector erase size. The examples below use the Winbond W25Q64 SPI flash which has a sector erase size of 4096 bytes. Examples of valid starting addresses for W25Q64 include 0x00000000 and 0x00001000.

Page write

Writes one page of flash memory. The first address must be a multiple of the page size. The examples below use the Winbond W25Q64 SPI Flash which has a page write size of 256 bytes. Examples of valid starting addresses include 0x00000000 and 0x00000100.

Read Flash Memory

Reads memory and stores 4 bytes in a register at a time.

Read Internal Flash Status

Reads the status bits of the internal flash memory. These bits indicate whether write is enabled and whether the flash memory is currently busy performing a requested action such as a sector erase.

The following registers are used to read and write the flash memory:

Table 8-1. Flash Configuration (Register 0x43)

Bit	Description	Reset	Type	Notes
3:0	**Transfer Data Byte Enable**	0xF	Write	
11:4	**UCA OP Code**	0xB	Write	
14:12	**Address Length**	0x3	Write	In bytes
15	Unused	0x0	Write	
21:16	**Dummy Bytes Length**	0x1	Write	In bytes
23:22	Unused	0x0	Write	
31:24	**Dummy Byte Pattern**	0x0	Write	

Table 8-2. Flash Write Data Length (Register 0x44)

Bit	Description	Reset	Type	Notes
24:0	**Write Data Length**	0x0	Write	
31:25	Unused	0x0	Write	

Table 8-3. Flash Read Data Length (Register 0x45)

Bit	Description	Reset	Type	Notes
24:0	**Read Data Length**	0x1FFFFFF	Write	Defaults to max length.
31:25	Unused	0x0	Write	

Table 8-4. Flash Starting Address (Register 0x46)

Bit	Description	Reset	Type	Notes
24:0	**Starting Address**	0x0	Write	
31:25	Unused	0x0	Write	

Table 8-5. Flash Programming Configuration (Register 0x47)

Bit	Description	Reset	Type	Notes
0	**Flash Programming Enable** 1: Enable Programming Mode 0: Disable Programming Mode	0x0	Write	
1	Unused	0x0	Write	
2	**Flash Abort** 1: Abort Flash Command and reset the Serial Flash controller. 0: Don’t abort on a write of zero	0x0	Write	
3	Unused	0x0	Write	
6:4	**Flash Command** 000-010 = Reserved 011 = Preload TXFIFO 100 = Start UCA 101 = Abort UCA 110 = Flush TX FIFO 111 = Flush RX FIFO	0x0	Write	
31:7	Unused	0x0	Write	

Table 8-6. Flash Write Data (Register 0x48)

Bit	Description	Reset	Type	Notes
31:0	**Write data to flash controller**	0x0	Write	Automatically generates a push pulse when written

Table 8-7. Flash Read Data (Register 0x49)

Bit	Description	Reset	Type	Notes
31:0	**Read data from flash controller**	0x0	Read	Flash Read Pop must be written to read next data

Table 8-8. Flash Read POP (Register 0x4A)

Bit	Description	Reset	Type	Notes
0	**POP request** 1: POP request sent to flash controller to be able to read 4 bytes of flash data	0x0	Write	Self-clearing
31:1	Unused	0x0	Write	

Table 8-9. Flash Status (Register 0x4B)

Bit	Description	Reset	Type	Notes
2:0	**User-supplied command access (UCA) Status** 000: Idle 100: Non-DMA UCA in progress 110: Flush TX FIFO in progress 111: Flush RX FIFO in progress	0x0	Read	
7:3	Unused	0x0	Read	
8	**Flash queue status** 0: Flash queue block is running an instruction 1: Flash queue is in idle state and locked for programming	0x0	Read	
31:9	Unused	0x0	Read	

8.2 Write Enable

This instruction must be executed before any modification to flash memory can be made including sector erase and page write commands.

To execute this instruction, follow the procedure below:

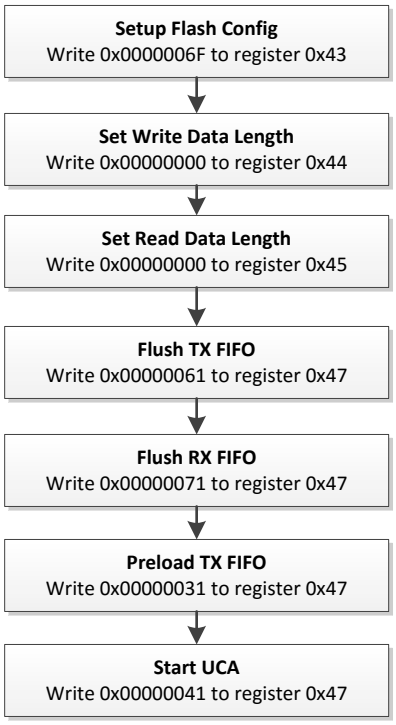


Figure 8-1. Write Enable Instruction

8.3 Sector Erase

This instruction will erase a sector of flash memory (4096 bytes).

The starting address must be a multiple of 0x1000. Examples of valid starting addresses include 0x00000000 and 0x00001000.

To execute this instruction, follow the steps below:

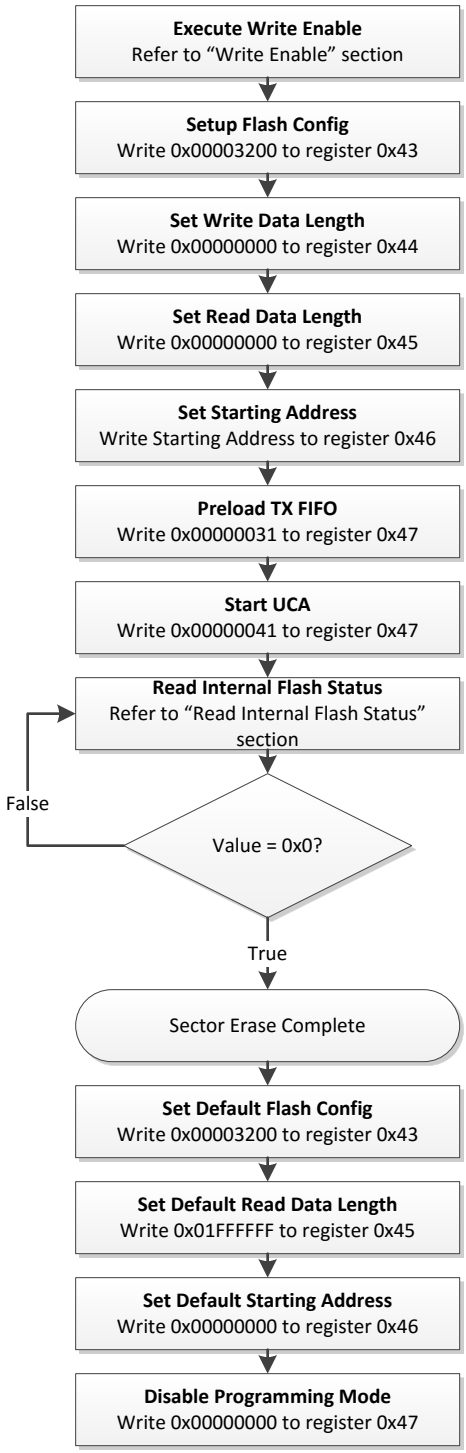


Figure 8-2. Erase Flash Sector

8.3.1 Example: Erase addresses 0x00000000 to 0x00001FFF

See diagram of steps below:

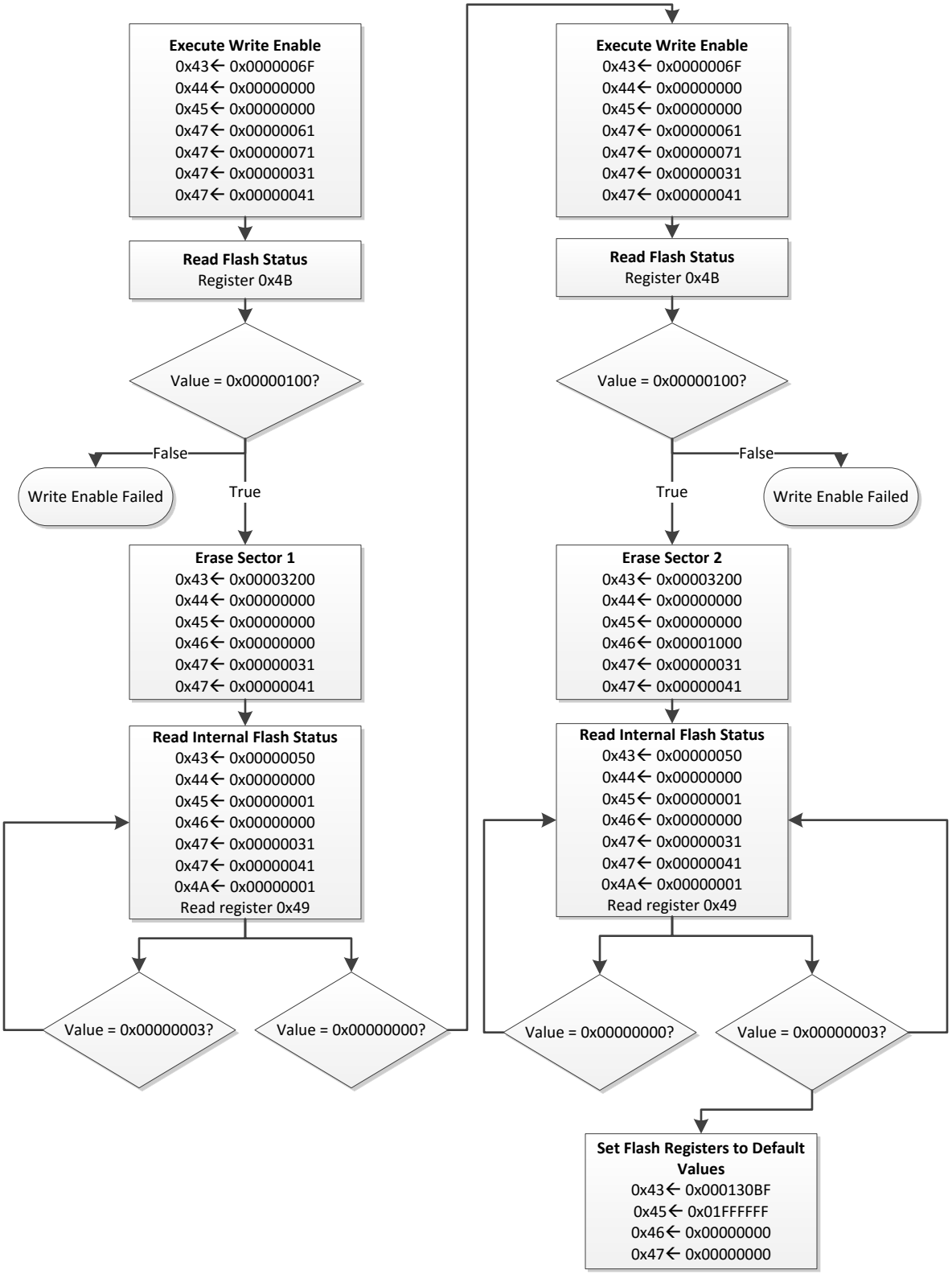


Figure 8-3. Erase Sector Example

8.4 Page Write

This instruction will write a maximum of a page of flash memory (256 bytes).

The first address must be a multiple of 0x100. Examples of valid starting addresses include 0x00000000 and 0x00000100.

To execute this instruction, follow the procedure in Figure 8-4.

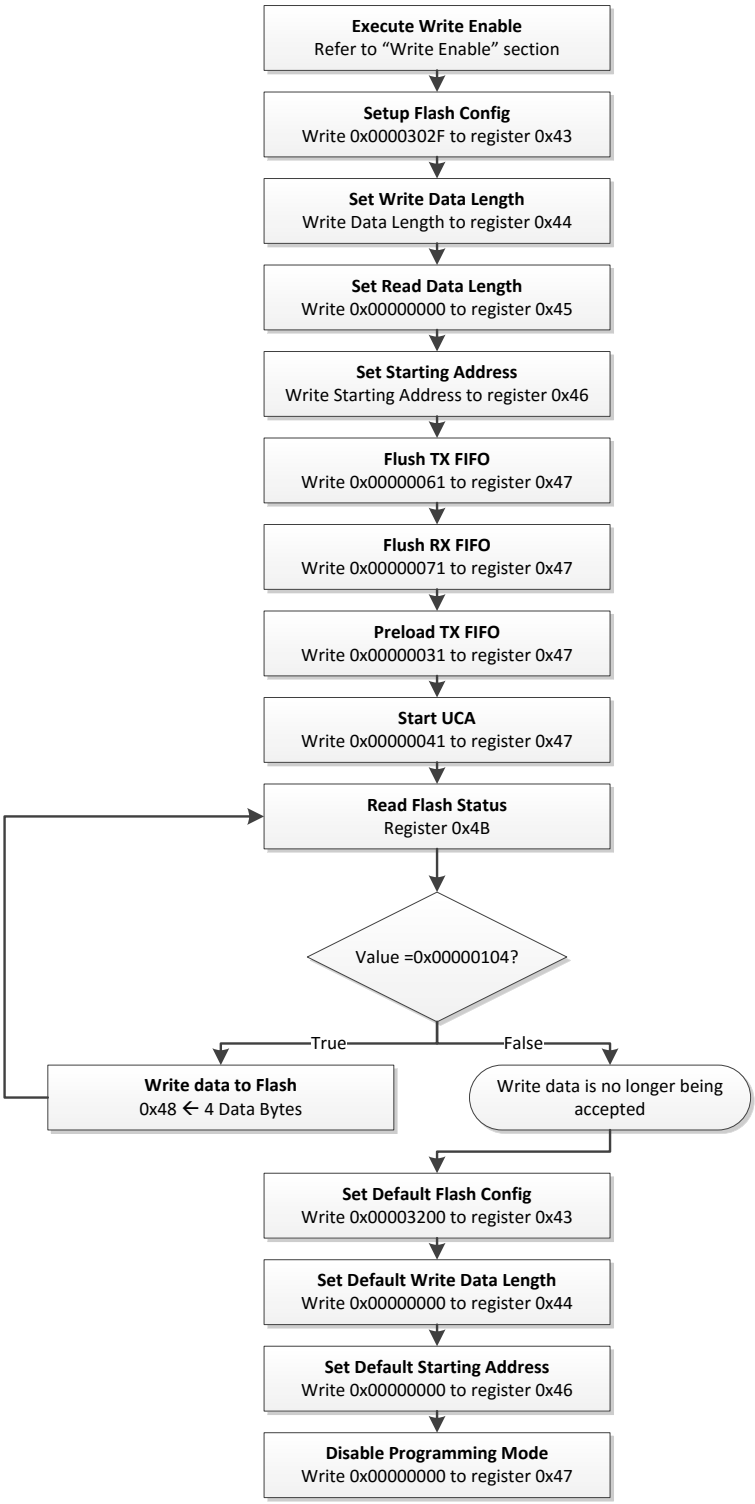


Figure 8-4. Write Flash Memory

8.4.1 Example: Write data (05 06 07 08 09 0A 0B 0C) starting at address 0x000FFF00

This example writes 8 data bytes by writing 4 bytes at a time. The endianness of each 4 bytes of data must be reversed. Therefore, this data is written as (08 07 06 05 | 0C 0B 0A 09).

To execute this example, follow the procedure in Figure 8-5.

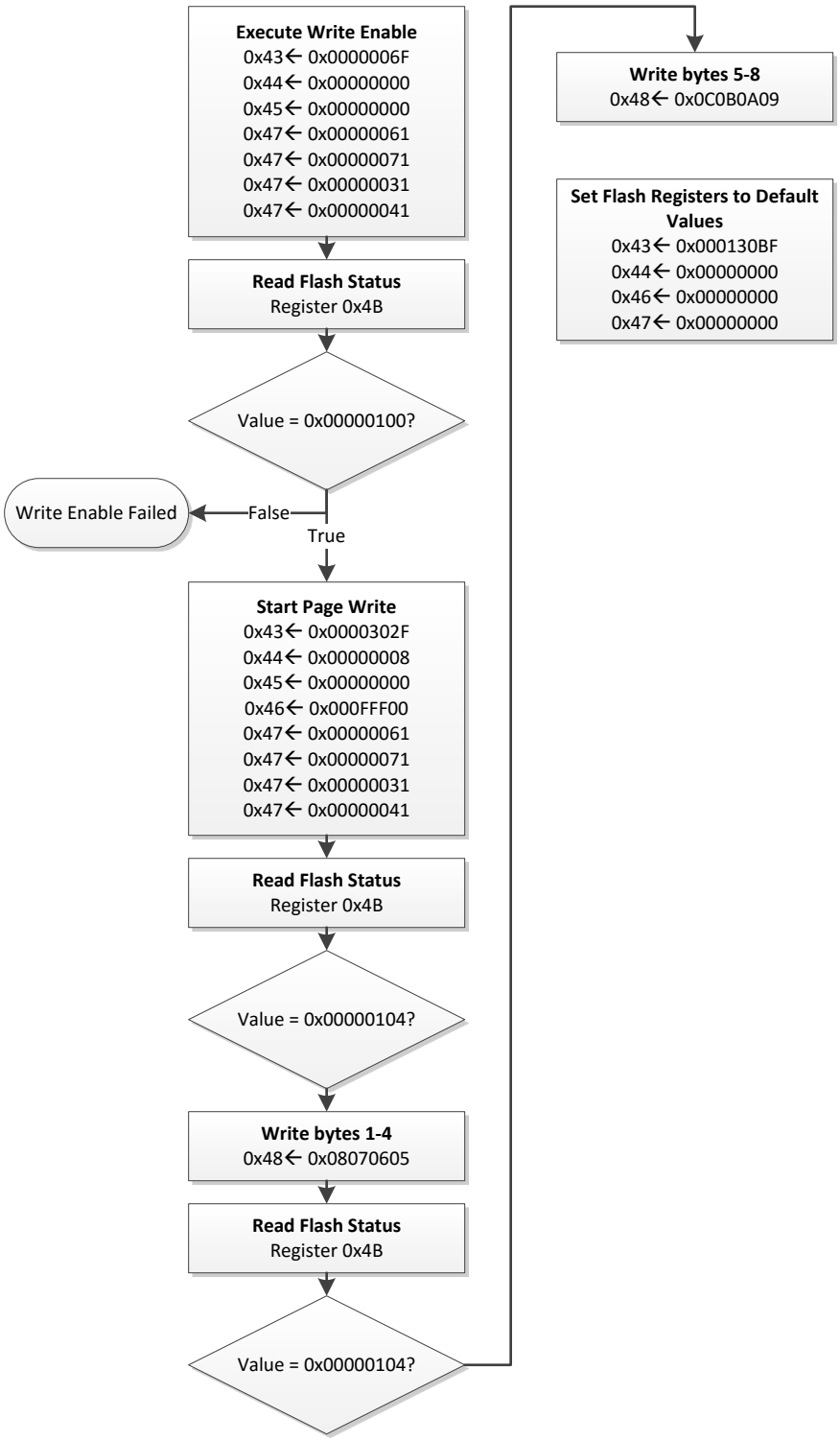


Figure 8-5. Page Write Example

8.5 Read Flash Memory

This instruction will read between 1 and 8388608 bytes of data the flash memory.

To execute this instruction, follow the procedure in Figure 8-6:

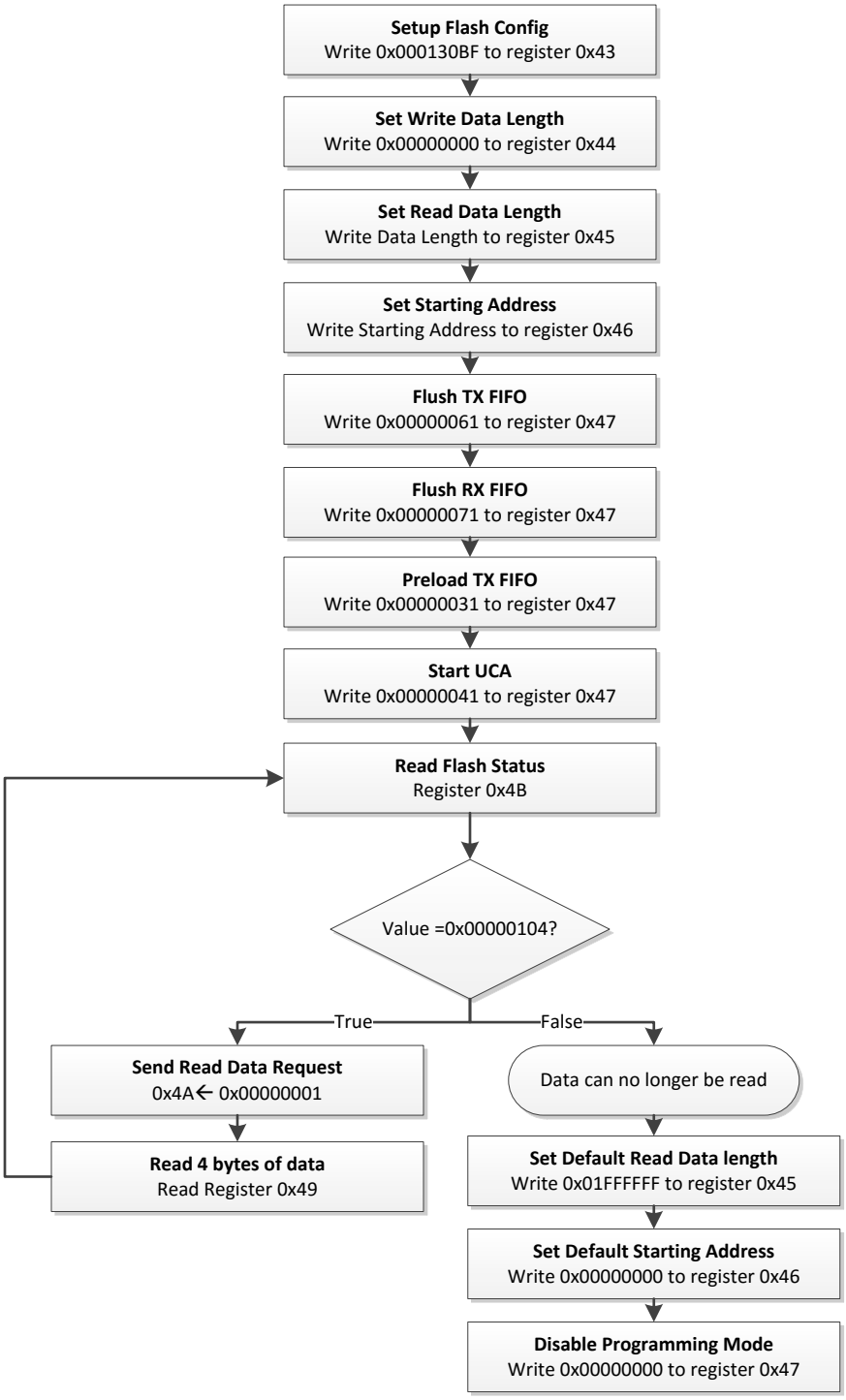


Figure 8-6. Read Flash Memory

8.5.1 Example: Read addresses 0x000FFF00 to 0x000FFF07

To execute this example, follow the procedure in Figure 8-7.

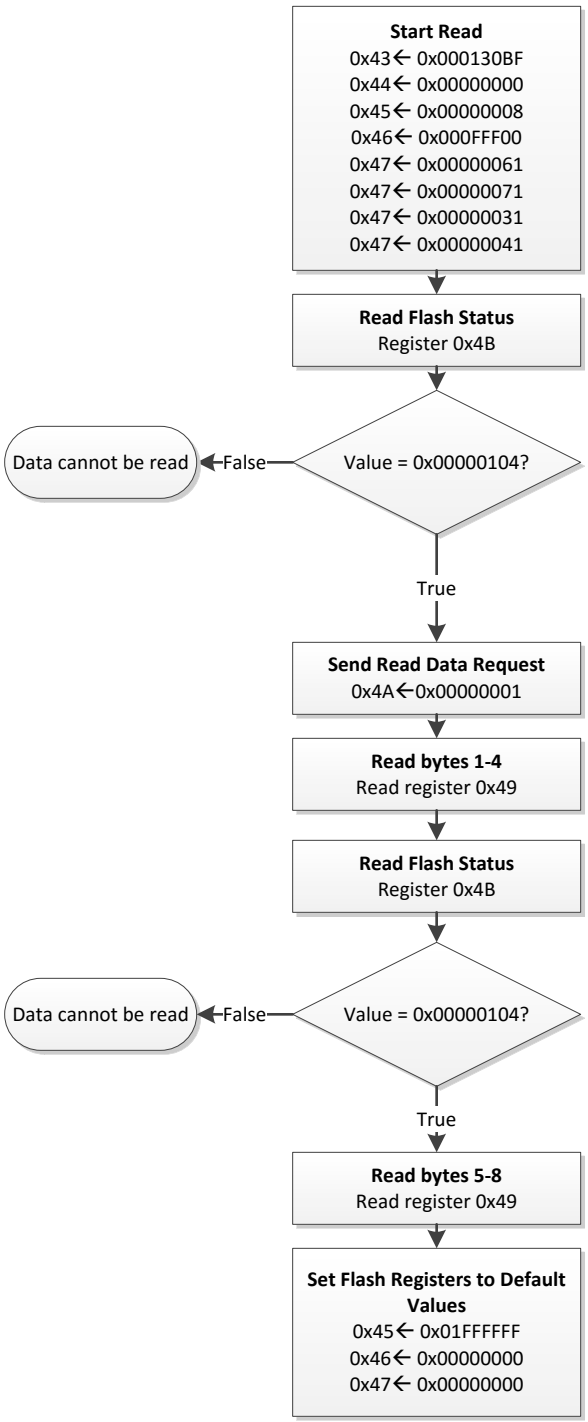


Figure 8-7. Read Flash Example

8.6 Read Internal Flash Status

This instruction reads back the internal flash status register. Executing this instruction will interrupt write and read instructions.

Table 8-10. Internal Flash Status

Bit	Description
0	Erase/Write in progress
1	Write enable flag 1: Write enabled 0: Write disabled

To execute this instruction, follow the procedure in Figure 8-8:

Figure 8-8. Read Internal Flash Status

Dimming LUT Groups

9.1 Dimming LUT Group Overview

Displaying an image requires the use of a dimming LUT group. A “dimming LUT group” consists of sets of three (3) tables stored in flash memory: LDC table, SEQ table, and CMT table. Smooth dimming requires the use of multiple sets of these 3 table types within the same dimming LUT group.

- SEQ Table – Controls the timing of data to the DMD and sets both the RGB and DMD duty cycle
- LDC Table – Controls LED timing attenuation and pulsing
- CMT Table – Controls image dithering and contour mitigation

9.2 SEQ Table

Each SEQ table controls the DMD mirror load and reset timing. For changes to this table refer to a TI Applications Engineer. The system level parameters that SEQ tables control include:

1. DMD Duty Cycle
   a. For example, setting the DMD mirrors to operate at 70/30 or 50/50 at different brightness ranges.
2. Expected video frame rate
   a. For example, when using a 60-Hz input video source, the SEQ table must be setup to drive the DMD for 60-Hz operation.
3. RGB Duty Cycle
   a. For example: Red 37%, Green 43%, Blue 20%.

9.3 LDC Table

Each LDC table controls the LED strobe timing. For changes to this table refer to a TI Applications Engineer. This table must match the SEQ table that is in use because the LED strobe timing must match the DMD timing operations. The system level parameters that LDC tables control include:

1. Continuous or Discontinuous mode
   a. Continuous mode – LDC table specifies the length of time that an LED emits light
   b. Discontinuous mode – LDC table specifies the number of light pulses required for each LED

9.4 CMT Table

Each CMT table controls how input video data is mapped to DMD sequence bits. For changes to this table refer to a TI Applications Engineer. This table must match the SEQ table that is in use because this mapping must match the DMD timing operations. The system level parameters that CMT tables control include:

1. Gamma curve
   a. The gamma curve that is applied to the input video is embedded in the CMT table.
2. Dithering
   a. The spatial dithering that is applied to the output image is determined by the CMT table data. The quantity of dithering seen in an output image is a result of image color bit depth as well as the applied gamma curve.
   b. Temporal dithering is a setting in the DLPC120-Q1 that shifts the offset for the spatial dithering

pattern each frame. This temporal component of dithering can be disabled by executing a command list stored in the application flash. Refer to Section 1.2 for details on command list execution.

9.5 Switching Dimming LUT Group

The following registers are used to switch the dimming LUT group:

Table 9-1. CMT Base Address (Register 0xCF)

Bit	Description	Reset	Type	Notes
24:0	CMT base address in flash	0x0	Write	
31:25	Unused	0x0	Write	

Table 9-2. SEQ Base Address (Register 0xD2)

Bit	Description	Reset	Type	Notes
24:0	SEQ base address in flash	0x0	Write	
31:25	Unused	0x0	Write	

Table 9-3. LDC Base Address (Register 0xD5)

Bit	Description	Reset	Type	Notes
24:0	LDC base address in flash	0x0	Write	
31:25	Unused	0x0	Write	

Each dimming LUT group can have a different RGB LED duty cycle. See a TI Applications engineer to setup the duty cycle of each dimming LUT group.

In order to switch dimming LUT groups, the LDC, SEQ, and CMT base addresses must be changed. These base addresses can be found in the configuration file (see Chapter 2).

9.6 Dimming within Dimming LUT Group

The following registers are used to change CMT, SEQ, and LDC tables within a single dimming LUT group:

Table 9-4. CMT Table Index (Register 0xD0)

Bit	Description	Reset	Type	Notes
7:0	CMT table index	0x0	Write	
31:8	Unused	0x0	Write	

Table 9-5. SEQ Table Index (Register 0xD3)

Bit	Description	Reset	Type	Notes
7:0	SEQ table index	0x0	Write	
31:8	Unused	0x0	Write	

Table 9-6. LDC Table Index (Register 0xD6)

Bit	Description	Reset	Type	Notes
7:0	LDC table index	0x0	Write	
31:8	Unused	0x0	Write	

After the dimming LUT group is chosen the CMT, SEQ, and LDC tables are chosen by writing the Table Index registers (0xD0, 0xD3, 0xD6).

Refer to the Piccolo Software Programmer's Guide for a detailed explanation of table selection.

Interrupt Events

10.1 Interrupt Event Overview

The interrupt registers are used to determine when events occur such as frame sync and splash screen load complete. The following registers are used for interrupt events:

Table 10-1. HUD Interrupt Clear (Register 0x00)

Bit	Description	Reset	Type	Notes
21:0	Reserved	0x0	Write	Set to 0
22	**Fsync Event**	0x0	Write	
24:23	Reserved	0x0	Write	Set to 0
25	**Discontinuous Mode Pulse Count Error**	0x0	Write	
26	**Splash Screen Load Done Event**	0x0	Write	Refer to Section 5.3 for use scenarios
31:27	Reserved	0x0	Write	Set to 0

Table 10-2. HUD Interrupt Set (Register 0x01)

Bit	Description	Reset	Type	Notes
21:0	Reserved	0x0	Read	Value may change during operation.
22	**Fsync Event**	0x0	Read	
24:23	Reserved	0x0	Read	Value may change during operation.
25	**Discontinuous Mode Pulse Count Error**	0x0	Read	
26	**Splash Screen Load Done Event**	0x0	Read	Refer to Section 5.3 for use scenarios
31:27	Reserved	0x0	Read	Value may change during operation.

Table 10-3. HUD Interrupt Enable (Register 0x02)

Bit	Description	Reset	Type	Notes
21:0	Reserved	0x0	Write	Set to 0
22	**Fsync Event**	0x0	Write	
24:23	Reserved	0x0	Write	Set to 0
25	**Discontinuous Mode Pulse Count Error**	0x0	Write	
26	**Splash Screen Load Done Event**	0x0	Write	Refer to Section 5.3 for use scenarios
31:27	Reserved	0x0	Write	Set to 0

Any events that are enabled will trigger an active high voltage on the DLPC120-Q1 HUD_INTR pin. Once the voltage at HUD_INTR becomes high, the HUD Interrupt Set register can be read to determine which event(s) triggered HUD_INTR. After the event has been handled appropriately, the bits can be cleared individually using the HUD Interrupt Clear register. HUD_INTR voltage will remain high until all triggered events are cleared.

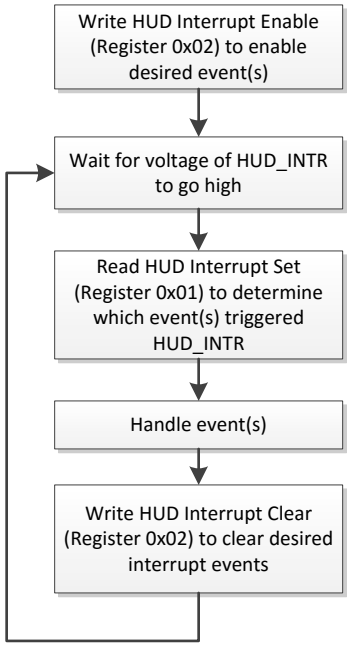


Figure 10-1. Interrupt Event Procedure

10.2 Discontinuous Mode Pulse Count Error

DLPC120-Q1 monitors light pulse count in discontinuous mode by counting falling edges of COMPOUTZ. If the DLPC120-Q1 does not receive the correct pulse count within a bit slice during discontinuous mode operation, it will throw an interrupt due to Pulse Count Error after the bit slice time has elapsed. Refer to Figure 10-3 for an example error situation. System-level conditions that can cause this error include:

- Photodiode disconnected
  - In the event that the photodiode is not connected, the optical feedback will not be able to reach the expected comparison threshold. In this situation, the DLPC120-Q1 will not receive any COMPOUTZ edges.
- Unreachable optical feedback comparison level
  - In the event that the expected optical comparison threshold is set to a level that is higher than the LED is able to achieve, the COMPOUTZ will not be triggered. In this situation, the DLPC120-Q1 will not register any light pulses.
- Slow pulse rise time
  - During normal operation, the expected number of pulses must be able to fit within the illumination time of the frame. If the amount of time that it takes to create a light pulse increases (due to operating condition changes such as temperature variation, LED aging, or input voltage variation) then it may not be possible to create the proper number of light pulses within the frame illumination time.

Refer to the examples below.

10.2.1 Example: No Pulse Count Error

During normal operation of discontinuous mode, the DLPC120-Q1 should receive the expected number of COMPOUTZ falling edges and no pulse count error will occur. An example of DLPC120-Q1 signals during normal discontinuous mode operation is shown below. In this example the DLPC120-Q1 expects four COMPOUTZ edges during the green slice and three COMPOUTZ edges during the blue slice.

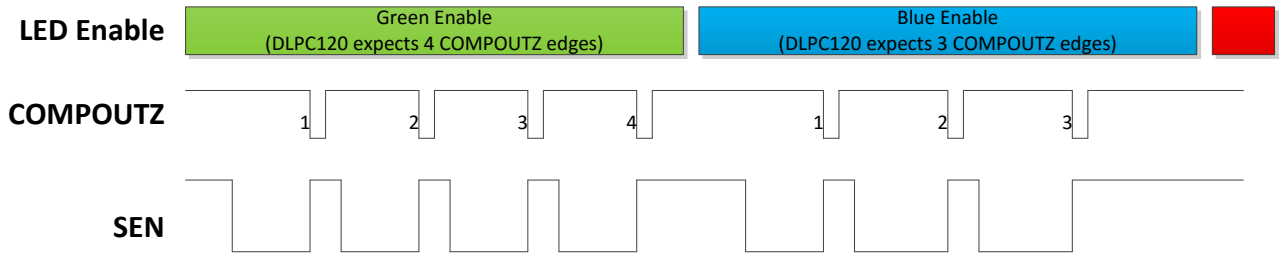


Figure 10-2. No Pulse Count Error

10.2.2 Example: Pulse Count Error Due to Photodiode Disconnect

In the event that the photodiode is not connected and the system is configured to operate in discontinuous mode, the DLPC120-Q1 will trigger a Discontinuous Mode Pulse Count Error interrupt. An example of a system-level scenario in which the photodiode disconnects during normal operation is shown below. Compare to the normal operation example in Figure 10-2.

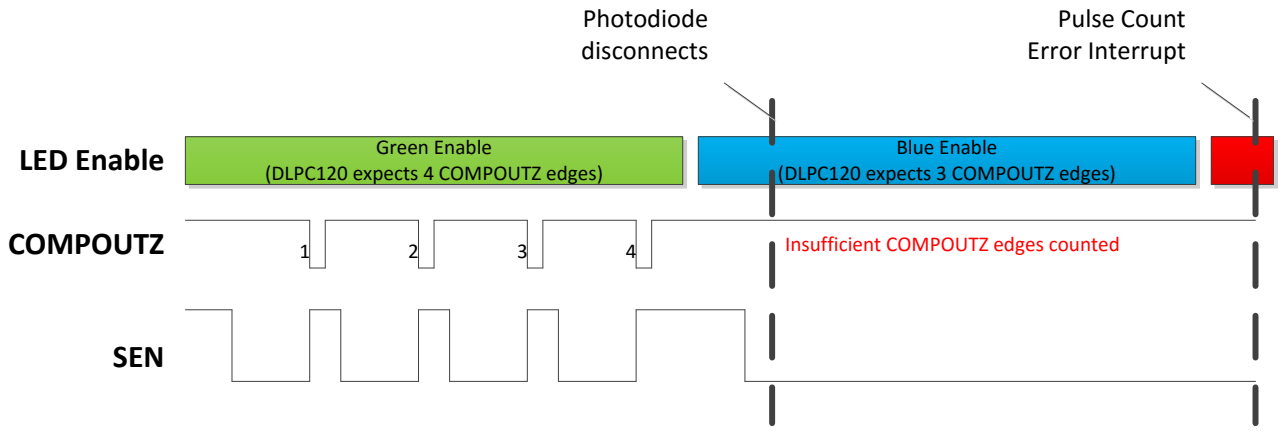


Figure 10-3. Pulse Count Error Photodiode Disconnect

10.3 Example: TI Reference Design Dimming

The TI recommended dimming micro-controller uses interrupt events to update backlight dimming every frame. Figure 10-4 shows the updates that the micro-controller makes each frame.

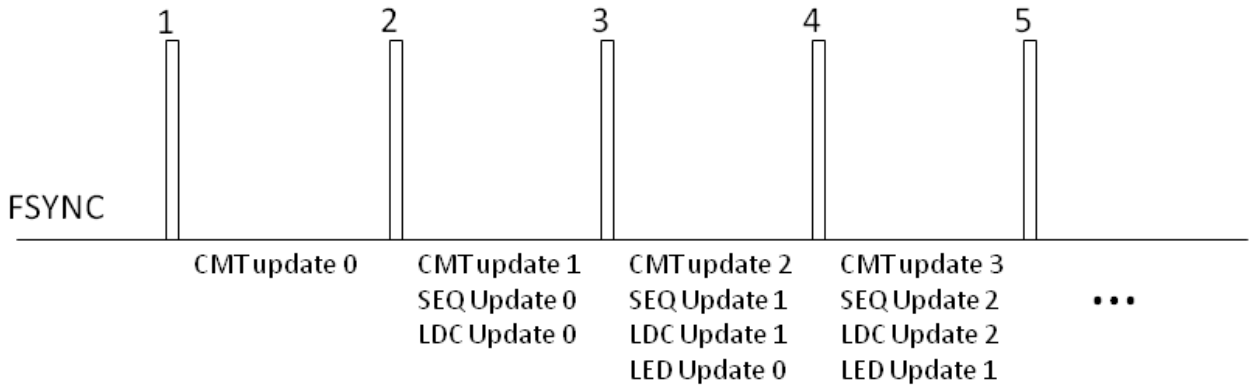


Figure 10-4. TI Reference Micro-Controller Dimming

The following procedure is used to perform these updates:

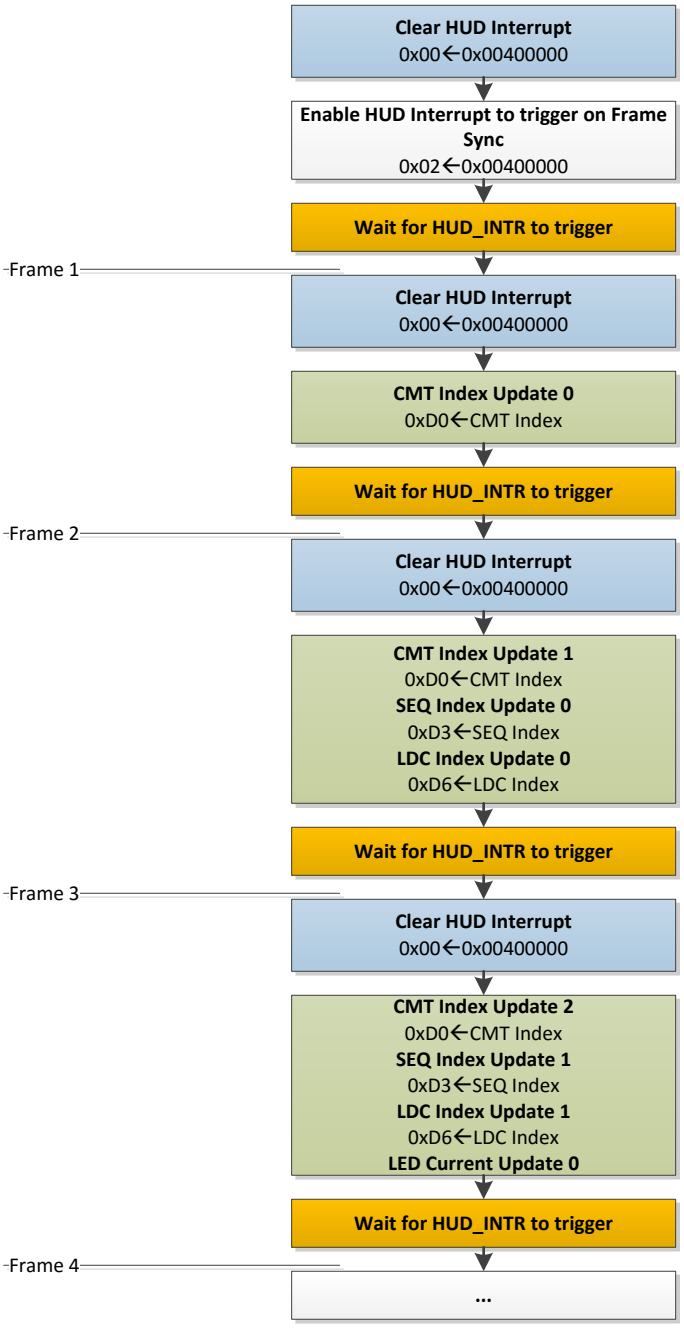


Figure 10-5. TI Recommended Dimming Procedure

DMD Park and Power-Down

11.1 DMD Park and Power-Down Overview

There are two steady state positions for the micro-mirrors of the DMD: on and off. When the DMD is not operating, the micro-mirrors are released from the on or off states, and they are “parked.” It is required that the mirrors are parked prior to power down in order to ensure reliability. This section focuses on the different types of DMD parking and the responsibility of each the DLPC120-Q1 and the host controller to manage these parking types.

11.2 Automatic DMD Temperature Parking

There are two events that can trigger the DMD to park. The first is the automatic action taken by the DLPC120-Q1 due to the DMD temperature surpassing the minimum or maximum limits. Please refer to the DMD datasheet for the operating temperature limits. Figure 11-1 below shows the behavior of the DLPC120-Q1 to automatically manage this temperature parking. Note that the DLPC120-Q1 will also automatically unpark the DMD when it crosses the boundary back into the operating range.

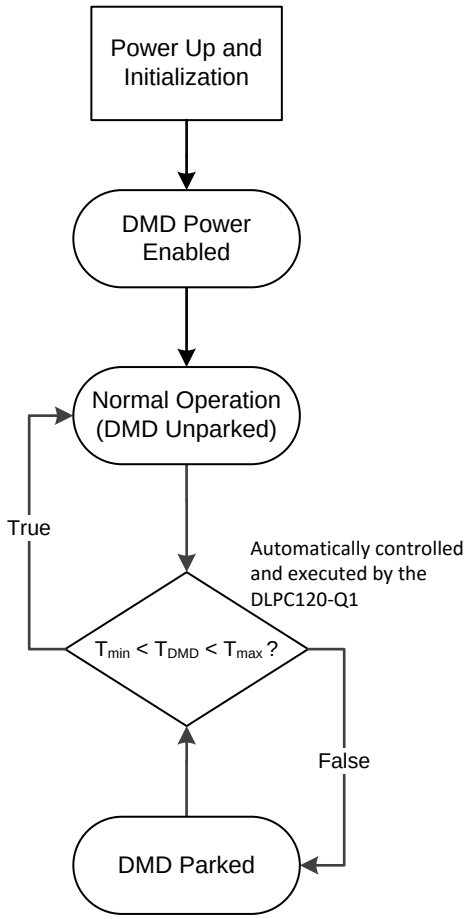


Figure 11-1. DLPC120-Q1 Controlled Temperature Park and Unpark Procedure

11.3 DMD Pre-Conditioning Sequence Parking and Unparking

The second type of park event requires action from the host micro-controller. For any parking event other than the automatic action due to temperature, such as a system power-down event, it is required that the host executes a Pre-Conditioning Sequence prior to parking the DMD. There is an I2C command list set that must be executing in order to ensure the Pre-Conditioning Sequence runs properly before parking. Similarly, a command list can be used to subsequently unpark the DMD. Figure 11-2 below shows the procedure of the host micro-controller managing the Pre-Conditioning parking and unparking.

In order to execute this Pre-Conditioning Sequence when parking and unparking the DMD, a series of Command Lists must be executed by the host micro-controller, as described in Section 11.3.1 and Section 11.3.2. The Generic Command Lists are stored in a fixed location (i.e. Index 0, 1, and 2) in the DLPC120-Q1 application flash, as described in Section A.2.1.

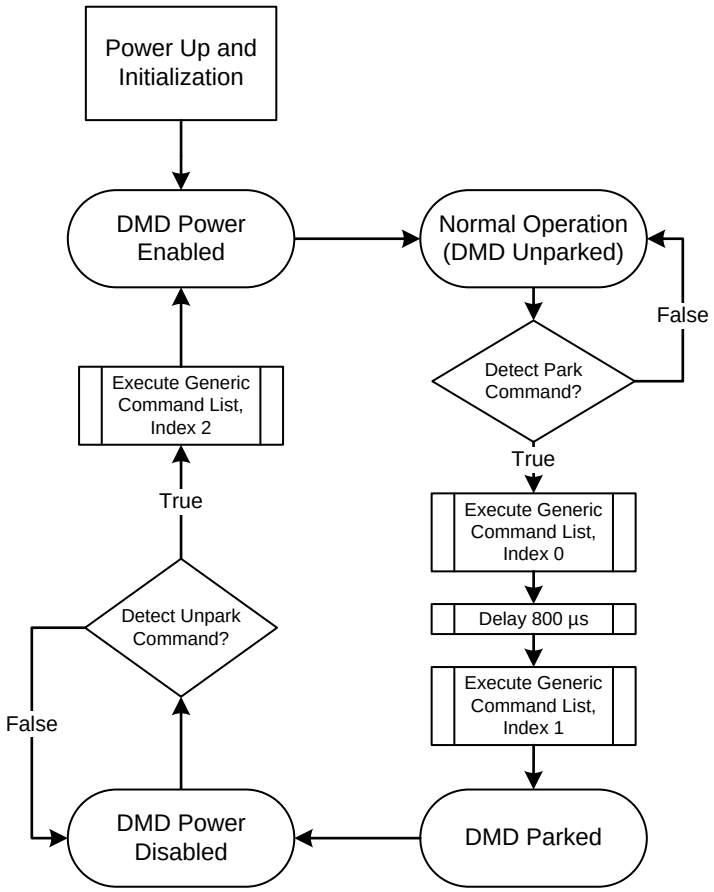


Figure 11-2. Host Controlled Park and Unpark Procedure

11.3.1 Pre-Conditioning Sequence and Park Implementation Example

The Pre-Conditioning Sequence for parking requires the following actions by the host:

1. Execute Generic Command List, Index 0
   a. This command list will disable the Sequencer (Section 12.3) in order to interrupt and halt normal operation of processing input video data
   b. Next, this command list will begin the Pre-Conditioning Sequence of the DMD, which takes 800 µs to execute
2. Delay
   a. This delay will allow the DMD to finish running the Pre-Conditioning Sequence initiated in Command List, Index 0
3. Execute Generic Command List, Index 1

a. This command list will de-assert output pin E14 (PWRGOOD_CNTRL) of the DLPC120-Q1, which shall cause input pin G13 (PWRGOOD) to de-assert and subsequently alert the DLPC120-Q1 to park the DMD.

This section provides a system-level implementation example for executing the Pre-Conditioning Sequence and parking of the DMD. The duration of the interface communication write events is design dependent. The times in the table below assume an I2C frequency of 400 kHz, as tested on a TI Evaluation Module (EVM).

Figure 11-3 shows a functional block diagram of the components involved in the Pre-Conditioning Sequence Park execution and the timing of the signals involved.

TYPE	PARAMETER	NOM	UNIT
Assuming 400-kHz I2C clock	t_{CL0}	560	µs
	t_{CL1}	310	µs
Fixed by design	$t_{PCDuration}$	800	µs
	$t_{Parking}$	200	µs
	$t_{PCReaction}$	15	µs
	$t_{PG1Reaction}$	6	µs
	$t_{DPEReaction}$	500	µs

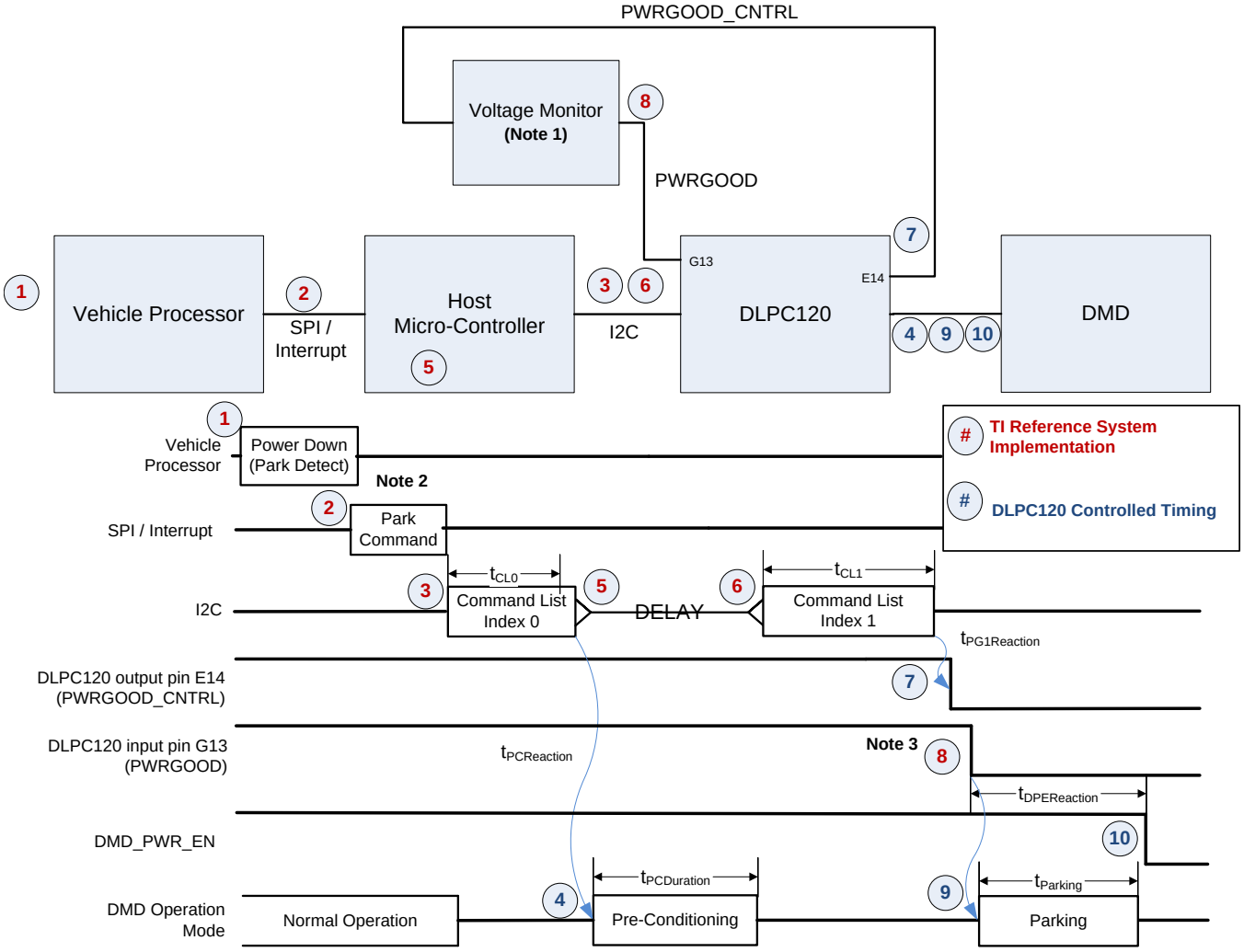


Figure 11-3. Pre-Conditioning Park Implementation Example

Note 1- The voltage monitor functionality may be integrated into micro-controller or implemented as an independent power detection IC.

Note 2 – Depending on the timing requirement, the indication to the Host to begin the Pre-Conditioning Sequence and parking may either be an SPI command or a digital interrupt pin. The recommendation is that the Host treats this indication as high priority.

Note 3 – PWRGOOD requires no delay on its falling edge.

11.3.2 Pre-Conditioning Sequence and Unpark Implementation Example

Upon initialization, the DLPC120-Q1 will automatically assert the PWRGOOD_CNTRL signal, allowing the DMD to unpark and begin normal operation.

In order to unpark the DMD after the Pre-Conditioning park described in Section 11.3.1, the following action is required by the host:

1. Execute Generic Command List, Index 2
   a. This command list will assert output pin E14 (PWRGOOD_CNTRL) of the DLPC120-Q1, which shall cause input pin G13 (PWRGOOD) to assert and subsequently unpark the DMD.
   b. Also, this command list enables the Sequencer (Section 12.3), allowing the DLPC120-Q1 to begin processing the input video mode and controlling the DMD and illumination.

This section provides a system-level implantation example for unparking the DMD after it was parked using the Pre-Conditioning Park method. The duration of the interface communication writes is design dependent. The times in the table below assume an I2C frequency of 400 kHz, as tested on a TI Evaluation Module (EVM).

Figure 11-4 shows a functional block diagram of the components involved in the Pre-Conditioning Sequence Unpark execution and the timing of the signals involved.

TYPE	PARAMETER	NOM	UNIT
Assuming 400-kHz I2C clock	$t_{PG1Start}$	580	µs
Fixed by design	$t_{DPEDelay}$	5	ms

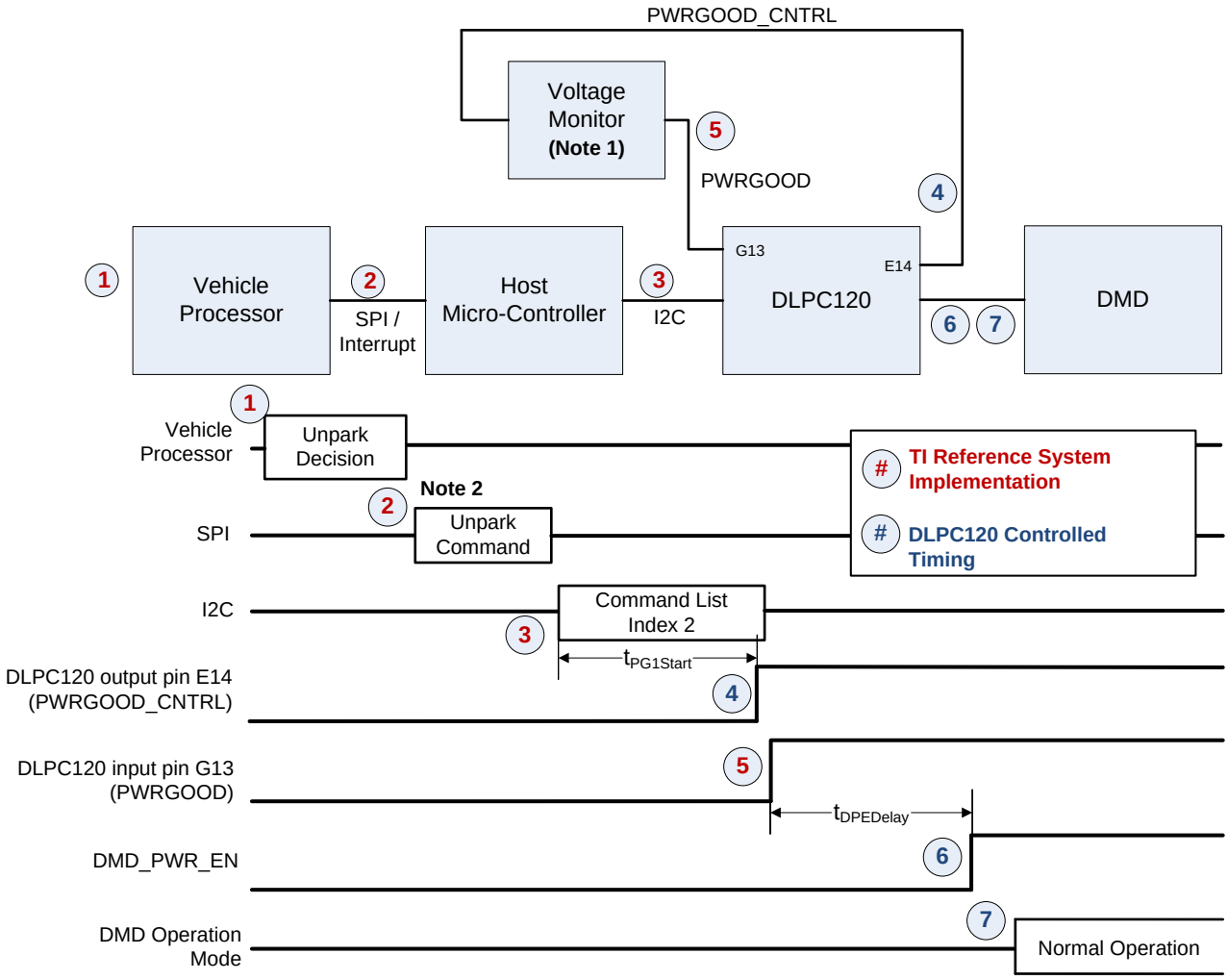


Figure 11-4. Pre-Conditioning Unpark Implementation Example

Note 1- The voltage monitor functionality may be integrated into micro-controller or implemented as an independent power detection IC.

Note 2 – Depending on the timing requirement, the indication to the Host to begin the Pre-Conditioning Unparking may either be an SPI command or a digital interrupt pin. The recommendation is that the Host treats this indication as high priority.

Support Functions

12.1 Support Function Overview

Support Functions are available to control specific actions of the chipset. Their function and usage in the sample Piccolo software is described below.

Note: Due to reserved bits located in these registers, the procedure for modifying the register values is as follows:

1. Read value stored in register
2. Modify fields of register value using bitwise operators
3. Write modified value to register

Table 12-1. Support Functions Control (Register 0xA0)

Bit	Description	Reset	Default Config	Type	Notes
3:0	Reserved	0x0	0x5	Write	This value will change during operation and can also vary based on default flash configuration.
4	**PLL Reset** 1: Force PLL to reset	0x0	0x0	Write	Refer to Section 12.2
31:5	Unused	0x0	0x0	Write	

Table 12-2. Sequencer Control (Register 0xBA)

Bit	Description	Reset	Type	Notes
0	**Sequencer Enable** 0: Disabled 1: Enabled	0x0	Write	Refer to Section 12.3
7:1	Reserved	0x0	Write	
31:8	Unused	0x0	Write	

Table 12-3. LDC Mode (Register 0xC0)

Bit	Description	Reset	Type	Notes
0	Unused	0x0	Read	
1	Reserved	0x1	Read	Value may change during operation.
2	**LDC Operating Mode** 0: Discontinuous Mode 1: Continuous Mode	0x0	Read	Refer to Section 12.4
31:3	Unused	0x0	Read	

12.2 PLL Reset Bit (Register 0xA0 bit 4)

This bit is set during the PLL initialization process described in Section 3.2. Writing this bit will force the PLL to reset and update PLL parameters. When this is done, the ASIC settings will reset as well and all registers will revert to their default state. Therefore this bit should only be set during system startup.

12.3 Sequencer Enable Bit (Register 0xBA bit 0)

NOTE: The DMD must be parked before sequencer is disabled. If sequencer is disabled without parking the DMD, the DMD mirrors will remain in their last known positions. This can decrease the lifetime of the DMD.

This bit will enable or disable the DLPC120-Q1 read of the DDR frame buffers. While sequencer is disabled sequence selection will continue to be read by the ASIC from flash memory, but no output image will be created.

The TI reference Piccolo software enables sequencer during the initialization procedure to ensure that an image will be displayed when dimming settings are applied.

For the Pre-Conditioning Parking and Unparking events described in Section 11.3, the sequencer is automatically disabled for parking and enabled for unparking as part of the command list execution.

12.4 LDC Operating Mode (Register 0xC0 bit 2)

This bit may be read to determine the current dimming operating mode: Continuous or Discontinuous.

The TI reference Piccolo software reads this register to determine the current operating mode when an Operating Mode read request is sent by the host through SPI.

ADC Sampling Timer (AST)

13.1 AST Overview

The AST function provides the ability to set a sequence-referenced interrupt signal. The settings in these registers will alter the output waveforms on the AST_CLR0, AST_HLD0, and AST_INTR0 pins of the DLPC120-Q1. This function is provided for future use for evaluation purposes only. The following registers are used to setup the AST function:

Table 13-1. AST Hold Start (Register 0xC9)

Bit	Description	Reset	Type	Notes
20:0	**Window Start** Number of 39-MHz clocks to wait from frame sync before de-asserting HOLD signal	0x1FFFFF	Write	
31:21	Unused	0x0	Write	

Table 13-2. AST Hold End (Register 0xCA)

Bit	Description	Reset	Type	Notes
20:0	**Window End** Number of 39-MHz clocks to wait from frame sync before re-asserting HOLD signal	0x1FFFFF	Write	
31:21	Unused	0x0	Write	

Table 13-3. AST Interrupt Position Control (Register 0xCC)

Bit	Description	Reset	Type	Notes
7:0	**Interrupt Point A** Number of sequence intervals after CLEAR signal before Interrupt A is asserted.	0xFF	Write	
15:8	**Interrupt Point B** Number of sequence intervals after CLEAR signal before Interrupt B is asserted.	0xFF	Write	
28:16	**Interrupt Delay** Number of 39-MHz clocks to delay both Interrupt A and Interrupt B after HOLD is reasserted.	0x1FFF	Write	
31:29	Unused	0x0	Write	

Table 13-4. AST Width Control (Register 0xCD)

Bit	Description	Reset	Type	Notes
12:0	**Interrupt Width** Number of 39-MHz clocks to assert the Interrupt signal	0x0	Write	
15:13	Unused	0x0	Write	
28:16	**Clear Width** Number of 39-MHz clocks to assert the Clear signal	0x1FFF	Write	
31:29	Unused	0x0	Write	

Table 13-5. AST Control (Register 0xCE)

Bit	Description	Reset	Type	Notes
0	**AST Function Enable** 0: Disable 1: Enable	0x0	Write	
1	Reserved	0x0	Write	Set to 0
3:2	Unused	0x0	Write	
7:4	Reserved	0x0	Write	Set to 0
8	**AST Clear Polarity** 0: Active high 1: Active low	0x1	Write	
9	**AST Hold Polarity** 0: Active high 1: Active low	0x1	Write	
10	**AST Interrupt Polarity** 0: Active high 1: Active low	0x1	Write	
31:11	Unused	0x0	Write	

The figure below shows how these register settings will affect the output signals:

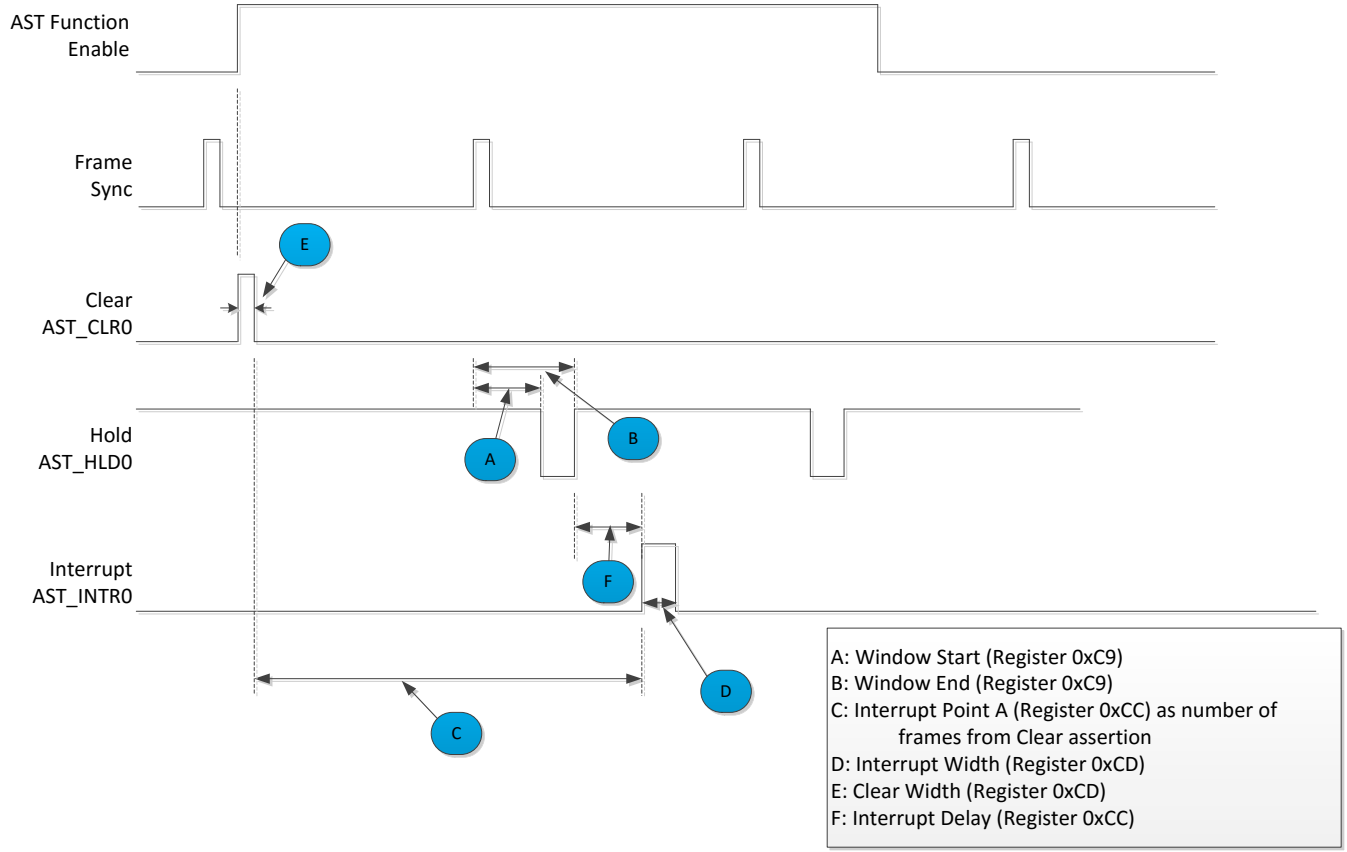


Figure 13-1. AST Signal Timing Diagram

13.2 Example: Set AST Interrupt rising edge to occur 1500 µs from sequence start

Calculate Interrupt start time in number of 39-MHz clocks	(1500*10^-6)*(39*10^6) = 58500 clocks
Calculate Window End. When setting Interrupt delay to 0, Window end should equal Interrupt start time.	Window End = 58500 clocks = 0xE484
Calculate Window Start. Window Start can occur any time before Window End. In this example it will be set to 1000 clocks before Window End.	Window Start = 58500-1000 = 57500 = 0xE09C
Calculate Interrupt Width. In this example only the Interrupt rising edge timing is critical, but the interrupt width will be set to 1024 clocks.	Interrupt width = 1024 = 0x400
Calculate Clear Width. In this example Clear width is not critical, but the Clear width will be set to 2048 clocks.	Clear width = 2048 = 0x800
Disable AST before setting up function	Write 0x00000000 to register 0xCE
Set Window Start Register	Write 0x0000E09C to register 0xC9
Set Window End Register	Write 0x0000E484 to register 0xCA
Set Interrupt Position Control Register • Interrupt delay = 0 (0 clocks after window end) • Interrupt point A = 1 (frame after Clear signal) • Interrupt point B = Interrupt point A = 1 (frame after Clear signal)	Write 0x00000101to register 0xCC
Set Width Control Register • Calculated Interrupt width = 0x400 • Calculated Clear width = 0x800	Write 0x08000400 to register 0xCD
Set AST Control Register • Enable = 1 • Clear = Active High • Hold = Active Low • Interrupt = Active High	Write 0x00000201 to register 0xCE

Appendix

A.1 List of Registers

Address	Name
0x00	HUD Interrupt Clear
0x01	HUD Interrupt Set
0x02	HUD Interrupt Enable
0x03	HUD Status
0x04	HUD Version
0x05	Flash Version
0x0D	VAC Control
0x38	Bezel Adjustment Function Enable
0x39	Bezel Adjustment Offset
0x3C	DMD Temperature
0x3D	Temperature Status
0x40	Command List Base Address
0x41	Command List Execution
0x43	Flash Configuration
0x44	Flash Write Data Length
0x45	Flash Read Data Length
0x46	Flash Starting Address
0x47	Flash Programming Configuration
0x48	Flash Write Data
0x49	Flash Read Data
0x4A	Flash Read POP
0x4B	Flash Status
0x50	Hardware Tests Enable
0x51	Hardware Tests Status
0x52	System BIST Checksum
0x53	Flash BIST Checksum
0x54	Video Detect Vsync Maximum
0x55	Video Detect Vsync Minimum
0x56	Video Detect Pclk Configuration
0x57	Video Detect Active Lines Configuration
0x58	Video Detect Active Pixels Configuration
0x59	Video Detect Fail Command List Address
0x5A	Video Detect Pass Command List Address
0x5B	Video Detect Status
0x5C	Video Detect Report
0x5D	Front End Video Test Start Position
0x5E	Front End Video Test End Position

Address	Name
0x5F	Front End Video Test Checksum
0x61	Flash BIST Start Address
0x62	Flash BIST Byte Count
0x64	DMD Device ID
0x65	DMD Drive Control
0x8E	Display Mode
0x98	Splash Function Control
0xA0	Support Functions Control
0xA4	PLL Control
0xA5	PLL Clock Configuration
0xAC	UMC Control
0xAE	UMC Read Data 1
0xAF	UMC Read Data 2
0xBA	Sequencer Control
0xC0	LDC Mode
0xC9	AST Hold Start
0xCA	AST Hold End
0xCC	AST Interrupt Position Control
0xCD	AST Width Control
0xCE	AST Control
0xCF	CMT Base Address
0xD0	CMT Table Index
0xD2	SEQ Base Address
0xD3	SEQ Table Index
0xD5	LDC Base Address
0xD6	LDC Table Index
0xE9	Image Curtain

A.2 DLPC120-Q1 Flash Layout and Description

The tables below describe the flash layout utilized by the DLPC120-Q1 and descriptions of the core data blocks. A DLPC120-Q1 flash binary and its block addresses and sizes can be provided by a TI applications engineer. A description of each flash file and its command lists will be provided.

A.2.1 Block Addressing

Block	Sub-block	Start address	Size
Default Configuration	Defconfig	0x0	Variable (1)
DDR Initialization	DDR Init 1	0 byte offset minimum (2)	Variable (1)
	DDR Init 2	0 byte offset minimum (2)	Variable (1)
	DDR Init 3	0 byte offset minimum (2)	Variable (1)
	DDR Init 4	0 byte offset minimum (2)	Variable (1)

(1) Variable size data blocks will be an integer multiple of 8 bytes and then terminated by the pattern 0xFFFFFFFF. Example: smallest size is 12 bytes (1*8+4), next largest size is 20 bytes (2*8+4).

(2) The start address of every sub-block will occur on a 16 byte boundary (example: 0x0, 0x10, 0x20, etc.).

Block	Sub-block	Start address	Size
Dimming LUT Group 1	LDC 1	0 byte offset minimum (3)	0x800
	LDC 2	0 byte offset minimum (3)	0x800
	LDC n	0 byte offset minimum (3)	0x800
	SEQ 1 (4)	0 byte offset minimum (3)	0x4000
	SEQ 2 (4)	0 byte offset minimum (3)	0x4000
	SEQ n (4)	0 byte offset minimum (3)	0x4000
	CMT 1	0 byte offset minimum (3)	0x4000
	CMT 2	0 byte offset minimum (3)	0x4000
	CMT n	0 byte offset minimum (3)	0x4000
Dimming LUT Group n	LDC 1	0 byte offset minimum (3)	0x800
	LDC 2	0 byte offset minimum (3)	0x800
	LDC n	0 byte offset minimum (3)	0x800
	SEQ 1 (4)	0 byte offset minimum (3)	0x4000
	SEQ 2 (4)	0 byte offset minimum (3)	0x4000
	SEQ n (4)	0 byte offset minimum (3)	0x4000
	CMT 1	0 byte offset minimum (3)	0x4000
	CMT 2	0 byte offset minimum (3)	0x4000
	CMT n	0 byte offset minimum (3)	0x4000
Splash Data	Splash 1	0 byte offset minimum (3)	864x480 resolution: 0x12FC00 608x684 resolution: 0x130980
	Splash 2	0 byte offset minimum (3)	864x480 resolution: 0x12FC00 608x684 resolution: 0x130980
	Splash n	0 byte offset minimum (3)	864x480 resolution: 0x12FC00 608x684 resolution: 0x130980
Command List: Splash	Splash 1	0 byte offset minimum (3)	Variable (5)
	Splash 2	0 byte offset minimum (3)	Variable (5)
	Splash n	0 byte offset minimum (3)	Variable (5)
Command List: Test Pattern	Test Pattern 1	0 byte offset minimum (3)	Variable (5)
	Test Pattern 2	0 byte offset minimum (3)	Variable (5)
	Test pattern n	0 byte offset minimum (3)	Variable (5)
Command List: External Video	External Video 1	0 byte offset minimum (3)	Variable (5)
	External Video 2	0 byte offset minimum (3)	Variable (5)
	External Video n	0 byte offset minimum (3)	Variable (5)
Command List: Generic	Pre-Conditioning 0	0 byte offset minimum (3)	0x44
	Pre-Conditioning 1	0 byte offset minimum (3)	0x24
	Pre-Conditioning 2	0 byte offset minimum (3)	0x14
	List 3	0 byte offset minimum (3)	Variable (5)
	List n	0 byte offset minimum (3)	Variable (5)

(3) The start address of every sub-block will occur on a 16 byte boundary (example: 0x0, 0x10, 0x20, etc.).

(4) Each SEQ table includes two embedded SEQ vectors for the DLP3030-Q1 system.

(5) Variable size data blocks will be an integer multiple of 8 bytes and then terminated by the pattern 0xFFFFFFFF. Example: smallest size is 12 bytes (1*8+4), next largest size is 20 bytes (2*8+4).

Note: Fixed-size data is padded with 0xFF bytes to fill the necessary size allocation.

Note: All start address "offsets" refer to the distance from the last byte of the preceding block.

A.2.2 Flash Block Descriptions

Block	Description
Default Config	Automatically run by ASIC at startup. Contains initialization settings necessary for ASIC operation.
DDR Initialization	Set of command lists that must be executed as part of the startup procedure in order to initialize the DDR2 memory used by the ASIC
SEQ	Contains DMD mirror load and reset timing. This timing determines the DMD mirror duty cycle, video frame rate, and RGB duty cycle.
LDC	Contains LED timing data that corresponds to the SEQ DMD timing. For Continuous mode LDCs, it contains LED drive timing information. For Discontinuous mode LDCs, it contains LED pulse count information.
CMT	Controls the mapping of input video data to a specific sequence. The desired video gamma curve also affects the contents of this table.
Splash Data	BGR 888 uncompressed splash image data
Command List: Splash	Command lists to execute that display splash data. There must be one command list per splash image data.
Command List: Test Pattern	Command lists to execute that display ASIC internal test patterns.
Command List: External Video	Command lists to execute that change the ASIC's expected external video resolution.
Command List: Generic	Additional command lists to for functions not described above. Examples include image flip settings and temporal dithering control. There will be three Generic Command Lists at fixed locations, as discussed in Section 11.3 .

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265