SN65DSI86 SW Examples

Mike Campbell *CCI*

ABSTRACT

The document contains examples of how to program the SN65DSI86 for different purposes. All examples in the document are written using the Total Phase Aardvark I2C controller script language. Software engineers should be able to take the scripts and port them to their specific environment.

Contents

Introduction

The SN65DSI86/96 DSI to embedded DisplayPort (eDP) bridge features a dual-channel MIPI® D-PHY receiver front-end configuration with 4 lanes per channel operating at 1.5Gbps per lane; a maximum input bandwidth of 12Gbps. The bridge decodes MIPI® DSI 18bpp RGB666 and 24bpp RGB888 packets and converts the formatted video data stream to DisplayPort with up to four lanes at either 1.62Gbps, 2.16Gbps, 2.43Gbps, 2.7Gbps, 3.24Gbps, 4.32Gbps, or 5.4Gbps.

This document provides examples of how to program the SN65DSI86 for various different purposes. The scripts below are for a Total Phase Aardvark I2C controller. Details on the Total phase Aardvark I2C controller can be obtained from the Total Phase website.

References

1. *SN65DSI86 Datasheet* (SLLSEH2)
2. *Aardvark Adapter User Manual*
3. *VESA DisplayPort Standard Version 1, Revision 2a. May 23, 2012*

Read Sink's EDID (Direct Method)

This script will read 256 bytes of the EDID.

```
<aardvark>
<configure i2c="1" spi="1" gpio="0" tpower="1" pullups="0" />
<i2c_bitrate khz="100" />

======Enable I2C_ADDR_CLAIM1======
<i2c_write addr="0x2D" count="1" radix="16">60 A1</i2c_write> />

======Write EDID base of 00 ======
<i2c_write addr="0x50" count="0" radix="16">00</i2c_write> />

======Read 256 bytes of EEID======
<i2c_read addr="0x50" count="256" radix="16">00</i2c_read> />

</aardvark>
```

Read Sink's DPCD Register

This example will read 16-bytes from sink's DPCD registers 0x00000.

```
<aardvark>
<configure i2c="1" spi="1" gpio="0" tpower="1" pullups="0" />
<i2c_bitrate khz="100" />

======Clear Status Registers for AUX Request======
```

<i2c_write addr="0x2D" count="1" radix="16">F4 FF</i2c_write> />

======Send AUX Request for 16 bytes from DPCD 0x00000 ======
==========DPCD Address is 0x00000 ======
<i2c_write addr="0x2D" count="1" radix="16">74 00 </i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">75 00 </i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">76 00 </i2c_write> />
==========Number of Bytes to Read is 16 ======
<i2c_write addr="0x2D" count="1" radix="16">77 10 </i2c_write> />
==========Send AUX Read Request ======
<i2c_write addr="0x2D" count="1" radix="16">78 91 </i2c_write> <sleep ms="20" />

======Read Status of AUX Request======
======Make sure SEND_INT is set and no errors======
<i2c_write addr="0x2D" count="0" radix="16">F4</i2c_write> />
<i2c_read addr="0x2D" count="1" radix="16">00</i2c_read> />

======Clear Status Registers for AUX Request======
<i2c_write addr="0x2D" count="1" radix="16">F4 FF</i2c_write> />

======Read 16 bytes from AUX_RDATA======
<i2c_write addr="0x2D" count="0" radix="16">79</i2c_write> />
<i2c_read addr="0x2D" count="16" radix="16">00</i2c_read> />

</aardvark>

Write Sink's DPCD Register

This example will write a 0x01 to the Sink's DPCD 0x0010A register.

<aardvark>
<configure i2c="1" spi="1" gpio="0" tpower="1" pullups="0" />
<i2c_bitrate khz="100" />

======Clear Status Registers for AUX Write Request======
<i2c_write addr="0x2D" count="1" radix="16">F4 FF</i2c_write> />

======Write DPCD Register 0x0010A in Sink to Enable ASSR======
==========Data to Write is 0x01. ======
<i2c_write addr="0x2D" count="1" radix="16">64 01</i2c_write> />
==========DPCD Register is 0x0010A. ======
<i2c_write addr="0x2D" count="1" radix="16">74 00</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">75 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">76 0A</i2c_write> />
==========Number of Bytes to Write is 1 ======
<i2c_write addr="0x2D" count="1" radix="16">77 01</i2c_write> />
==========Send AUX Write Request ======
<i2c_write addr="0x2D" count="1" radix="16">78 81</i2c_write> <sleep ms="10" />

======Read Status of AUX Request======
======Make sure SEND_INT is set and no errors======
<i2c_write addr="0x2D" count="0" radix="16">F4</i2c_write> />

<i2c_read addr="0x2D" count="1" radix="16">00</i2c_read> />

======Clear Status Registers for AUX Request======
<i2c_write addr="0x2D" count="1" radix="16">F4 FF</i2c_write> />

</aardvark>

1920x1080 2DP @ HBR

This example will configure the DSI86 for the following configuration:

REFCLK: 19.2MHz
DSI mode: Single
# of DSI Lanes: 4
DSI Clock Frequency: 422.5MHz
# of DP Lanes: 2
DP Datarate: HBR (2.7Gbps)
bpp: 24
Hactive: 1920
Vactive: 1080
HSync: 16 negative
VSync: 14 negative
HBP: 152
VBP: 19
HFP: 16
VFP: 3

<aardvark>
<configure i2c="1" spi="1" gpio="0" tpower="1" pullups="0" />
<i2c_bitrate khz="100" />

======REFCLK 19.2MHz ======
<i2c_write addr="0x2D" count="1" radix="16">0A 02</i2c_write> />

======Single 4 DSI lanes======
<i2c_write addr="0x2D" count="1" radix="16">10 26</i2c_write> />

======DSIA CLK FREQ 422.5MHz======
<i2c_write addr="0x2D" count="1" radix="16">12 55</i2c_write> />

======enhanced framing and ASSR======
<i2c_write addr="0x2D" count="1" radix="16">5A 05</i2c_write> />

======2 DP lanes no SSC======
<i2c_write addr="0x2D" count="1" radix="16">93 20</i2c_write> />

======HBR======
<i2c_write addr="0x2D" count="1" radix="16">94 80</i2c_write> />

======PLL ENABLE======
<i2c_write addr="0x2D" count="1" radix="16">0D 01</i2c_write> <sleep ms="10" />

======Verify PLL is locked======
<i2c_write addr="0x2D" count="0" radix="16">0A</i2c_write> />
<i2c_read addr="0x2D" count="2" radix="16">00</i2c_read> <sleep ms="10" />

======POST-Cursor2 0dB ======
<i2c_write addr="0x2D" count="1" radix="16">95 00</i2c_write> />

======Write DPCD Register 0x0010A in Sink to Enable ASSR======
<i2c_write addr="0x2D" count="1" radix="16">64 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">74 00</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">75 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">76 0A</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">77 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">78 81</i2c_write> <sleep ms="10" />

======Semi-Auto TRAIN ======
<i2c_write addr="0x2D" count="1" radix="16">96 0A</i2c_write> <sleep ms="20" />

======Verify Training was successful======
<i2c_write addr="0x2D" count="0" radix="16">96</i2c_write> />
<i2c_read addr="0x2D" count="1" radix="16">00</i2c_read> <sleep ms="10" />

=====CHA_ACTIVE_LINE_LENGTH is 1920 =======
<i2c_write addr="0x2D" count="2" radix="16">20 80 07</i2c_write> />

=====CHA_VERTICAL_DISPLAY_SIZE is 1080 =======
<i2c_write addr="0x2D" count="2" radix="16">24 38 04</i2c_write> />

=====CHA_HSYNC_PULSE_WIDTH is 16 negative =======
<i2c_write addr="0x2D" count="2" radix="16">2C 10 80</i2c_write> />

=====CHA_VSYNC_PULSE_WIDTH is 14 negative=======
<i2c_write addr="0x2D" count="2" radix="16">30 0E 80</i2c_write> />

=====CHA_HORIZONTAL_BACK_PORCH is 152=======
<i2c_write addr="0x2D" count="1" radix="16">34 98</i2c_write> />

=====CHA_VERTICAL_BACK_PORCH is 19=======
<i2c_write addr="0x2D" count="1" radix="16">36 13</i2c_write> />

=====CHA_HORIZONTAL_FRONT_PORCH is 16=======
<i2c_write addr="0x2D" count="1" radix="16">38 10</i2c_write> />

=====CHA_VERTICAL_FRONT_PORCH is 3=======
<i2c_write addr="0x2D" count="1" radix="16">3A 03</i2c_write> />

======DP- 24bpp======
<i2c_write addr="0x2D" count="1" radix="16">5B 00</i2c_write> />

=====COLOR BAR disabled=======
<i2c_write addr="0x2D" count="1" radix="16">3C 00</i2c_write> />

======enhanced framing, ASSR, and Vstream enable======

<i2c_write addr="0x2D" count="1" radix="16">5A 0D</i2c_write> />

</aardvark>

2560x1440 2DP @ HBR2

This example will configure the DSI86 for the following configuration:

REFCLK: 19.2MHz
DSI mode: Dual Odd/Even
# of DSI Lanes: 8
DSI Clock Frequency: 362.5MHz
# of DP Lanes: 2
DP Datarate: HBR2 (5.4Gbps)
bpp: 24
Hactive: 2560
Vactive: 1440
HSync: 32 positive
VSync: 5 negative
HBP: 80
VBP: 33
HFP: 48
VFP: 3

<aardvark>
<configure i2c="1" spi="1" gpio="0" tpower="1" pullups="0" />
<i2c_bitrate khz="100" />

======REFCLK 19.2MHz ======
<i2c_write addr="0x2D" count="1" radix="16">0A 02</i2c_write> />

======Dual Odd/Even 4 DSI lanes======
<i2c_write addr="0x2D" count="1" radix="16">10 00</i2c_write> />

======DSI CLK FREQ 362.5MHz======
<i2c_write addr="0x2D" count="1" radix="16">12 48</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">13 48</i2c_write> />

======enhanced framing and ASSR======
<i2c_write addr="0x2D" count="1" radix="16">5A 05</i2c_write> />

======2 DP lanes and no SSC======
<i2c_write addr="0x2D" count="1" radix="16">93 20</i2c_write> />

======HBR2======
<i2c_write addr="0x2D" count="1" radix="16">94 E0</i2c_write> />

======PLL ENABLE======
<i2c_write addr="0x2D" count="1" radix="16">0D 01</i2c_write> <sleep ms="10" />

======Verify PLL is locked======
<i2c_write addr="0x2D" count="0" radix="16">0A</i2c_write> />

<i2c_read addr="0x2D" count="2" radix="16">00</i2c_read> <sleep ms="10" />

======POST-Cursor2 0dB ======
<i2c_write addr="0x2D" count="1" radix="16">95 00</i2c_write> />

======Write DPCD Register 0x0010A in Sink to Enable ASSR======
<i2c_write addr="0x2D" count="1" radix="16">64 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">74 00</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">75 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">76 0A</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">77 01</i2c_write> />
<i2c_write addr="0x2D" count="1" radix="16">78 81</i2c_write> <sleep ms="10" />

======Semi-Auto TRAIN ======
<i2c_write addr="0x2D" count="1" radix="16">96 0A</i2c_write> <sleep ms="20" />

======Verify Training is successful======
<i2c_write addr="0x2D" count="0" radix="16">96</i2c_write> />
<i2c_read addr="0x2D" count="1" radix="16">00</i2c_read> <sleep ms="10" />

=====CHA_ACTIVE_LINE_LENGTH is 1280=======
<i2c_write addr="0x2D" count="2" radix="16">20 00 05</i2c_write> />

=====CHB_ACTIVE_LINE_LENGTH is 1280=======
<i2c_write addr="0x2D" count="2" radix="16">22 00 05</i2c_write> />

=====CHA_VERTICAL_DISPLAY_SIZE is 1440=======
<i2c_write addr="0x2D" count="2" radix="16">24 A0 05</i2c_write> />

=====CHA_HSYNC_PULSE_WIDTH is 32 positive=======
<i2c_write addr="0x2D" count="2" radix="16">2C 20 00</i2c_write> />

=====CHA_VSYNC_PULSE_WIDTH is 5 negative=======
<i2c_write addr="0x2D" count="2" radix="16">30 05 80</i2c_write> />

=====CHA_HORIZONTAL_BACK_PORCH is 80=======
<i2c_write addr="0x2D" count="1" radix="16">34 50</i2c_write> />

=====CHA_VERTICAL_BACK_PORCH is 33=======
<i2c_write addr="0x2D" count="1" radix="16">36 21</i2c_write> />

=====CHA_HORIZONTAL_FRONT_PORCH is 48=======
<i2c_write addr="0x2D" count="1" radix="16">38 30</i2c_write> />

=====CHA_VERTICAL_FRONT_PORCH is 3=======
<i2c_write addr="0x2D" count="1" radix="16">3A 03</i2c_write> />

======DP-24BPP Enable======
<i2c_write addr="0x2D" count="1" radix="16">5B 00</i2c_write> />

=====COLOR BAR disabled=======
<i2c_write addr="0x2D" count="1" radix="16">3C 00</i2c_write> />

======enhanced framing, ASSR, and Vstream enable======

<i2c_write addr="0x2D" count="1" radix="16">5A 0D</i2c_write> />

</aardvark>

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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