



# Ultimate IOTG Guide for Displays in Embedded Applications

**Display Configuration, Troubleshooting and Support**

---

**May 2021**

***Intel Confidential***



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: <http://www.intel.com/design/literature.htm>.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at <http://www.intel.com/>.

Intel, the Intel Atom, the Intel Core, the Intel Pentium and Celeron, the Intel logo, the Thunderbolt and the Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2021, Intel Corporation. All rights reserved.



## Contents

<b>1.0</b>	<b>Introduction .....</b>	<b>10</b>
1.1	Terminology.....	11
<b>2.0</b>	<b>Display Architecture Overview – Planes, Pipes and Ports .....</b>	<b>15</b>
2.1	Planes .....	15
2.2	Pipes.....	16
2.3	Ports .....	17
2.4	Display Driver Interface (DDI).....	17
<b>3.0</b>	<b>DisCon Tool.....</b>	<b>20</b>
3.1	Obtaining a VBT .....	20
3.2	Using DisCon .....	20
<b>4.0</b>	<b>Local Flat Panels (LFPs).....</b>	<b>23</b>
4.1	Embedded Display Port (eDP) .....	23
4.1.1	Advantages – eDP .....	24
4.1.2	Disadvantages – eDP .....	25
4.1.3	Cable/Connector – eDP .....	25
4.1.4	Troubleshooting – eDP .....	26
4.1.5	Maximum Resolution Support – eDP .....	27
4.1.6	Panel Self Refresh .....	28
4.1.7	Multi SST Operation (MSO) .....	30
4.1.8	eDP Bifurcation .....	33
4.2	Low-Voltage Differential Signaling (LVDS).....	35
4.2.1	Advantages – LVDS .....	36
4.2.2	Disadvantages – LVDS .....	36
4.2.3	Cable/Connector – LVDS.....	37
4.2.4	Troubleshooting – LVDS.....	38
4.3	MIPI Display Serial Interface (MIPI-DSI).....	38
4.3.1	DisCon Tool Settings to Enable MIPI-DSI Panel.....	39
4.3.2	MIPI-DSI Flow Diagram.....	40
4.3.3	DisCon Tool Settings to configure the selected MIPI-DSI panel .....	41
4.3.4	MIPI Sequence Tool Setting.....	42
4.3.5	Advantages – MIPI-DSI .....	44
4.3.6	Disadvantages – MIPI-DSI .....	44
4.3.7	Cable/Connector – MIPI-DSI .....	44
4.3.8	Troubleshooting – MIPI-DSI.....	45
<b>5.0</b>	<b>External Flat Panels (EFP) .....</b>	<b>46</b>
5.1	DisplayPort (DP) .....	46
5.1.1	Advantages – DP .....	47
5.1.2	Disadvantages - DP .....	47
5.1.3	Cable/Connector – DP .....	47



5.1.4	Troubleshooting – DP .....	49
5.1.5	Maximum Resolution Support on DP .....	50
5.1.6	Active and Passive Adapters on DP .....	51
5.1.7	Dual-Mode DisplayPort (DP with HDMI/DVI Compatible).....	53
5.1.8	DP to VGA (CRT) Adapter .....	55
5.1.9	DP Alt Mode on USB Type-C .....	55
5.1.10	DP MST .....	57
5.1.11	DP-IN .....	61
5.2	High Definition Multimedia Interface (HDMI) .....	62
5.2.1	Advantages – HDMI.....	63
5.2.2	Disadvantages – HDMI.....	63
5.2.3	Cable/Connector – HDMI.....	63
5.2.4	Troubleshooting – HDMI .....	65
5.2.5	Max Resolution Support on HDMI Interface .....	65
5.2.6	Level Shifter and Protocol Converter (LSPCON*).....	66
5.3	Digital Visual Interface (DVI) .....	68
5.3.1	DVI-A / DVI-D / DVI-I.....	68
5.3.2	Advantages – DVI.....	69
5.3.3	Disadvantages -DVI.....	69
5.3.4	Cable/Connector – DVI.....	70
5.3.5	Troubleshooting - DVI.....	70
5.4	Video Graphics Array (VGA) .....	71
5.4.1	Advantages – VGA .....	71
5.4.2	Disadvantages – VGA .....	71
5.4.3	Cable/Connector - VGA.....	71
5.4.4	Troubleshooting - VGA .....	72
<b>6.0</b>	<b>EDID-less Displays.....</b>	<b>73</b>
6.1	LFP EDID-less Settings.....	73
6.1.1	EDID-less settings for Gen11 and newer platforms.....	73
6.1.2	EDID-less settings for Gen9 and older legacy platforms .....	76
6.2	EFP EDID-less Settings.....	79
<b>7.0</b>	<b>Collage Mode aka Combined Desktop .....</b>	<b>81</b>
7.1	What is the Maximum Resolution of Collage Displays without External Splitter? .....	82
7.1.1	Horizontal Display .....	82
7.1.2	Vertical Display .....	83
7.2	Things to Keep in Mind .....	83
7.3	How to Enable Collage/Combined Desktop .....	84
7.4	Combined Desktop Limitations.....	88
7.5	Bezel Correction.....	88
<b>8.0</b>	<b>8K Resolution Support.....</b>	<b>91</b>
8.1	Two Pipe Two Port.....	91
8.2	Two Pipe One Port.....	92
8.3	Future.....	92



<b>9.0</b>	<b>High-bandwidth Digital Content Protection (HDCP).....</b>	<b>93</b>
9.1	HDCP Levels.....	93
9.2	O/S HDCP Support.....	93
9.3	Troubleshooting .....	94
<b>10.0</b>	<b>KVM Remote (KVMR).....</b>	<b>95</b>
10.1	Enabling KVMR with Platforms that do not have the KVMR Option.....	96
<b>11.0</b>	<b>High Dynamic Range (HDR).....</b>	<b>97</b>
11.1	HDR Functional Stack on Intel® Graphics Platforms .....	98
11.2	Activating Native HDR Mode on Windows* .....	99
11.3	Screenshot Showing 'Stream HDR Video' Enabled .....	101
<b>12.0</b>	<b>Display Audio.....</b>	<b>102</b>
12.1	Intel DAC vs ACX DAC.....	103
12.1.1	MSFT* ACX DAC Loading Behavior in Device Manager (DM).....	103
12.1.2	MSFT* ACX DAC Version Details.....	104
12.2	Troubleshooting .....	105
<b>13.0</b>	<b>Discrete Graphics (dGPUs) .....</b>	<b>106</b>
13.1	Troubleshooting .....	107
13.2	Concurrent Mode.....	107
13.2.1	Concurrent Limitations.....	108
13.3	Hybrid (Switchable) Mode .....	108
13.3.1	Hybrid Requirements.....	109
13.3.2	Utilizing Hybrid .....	110
<b>14.0</b>	<b>Hot-Plug Detect (HPD) .....</b>	<b>111</b>
<b>15.0</b>	<b>Redriver / Retimer.....</b>	<b>113</b>
15.1	Redriver.....	114
15.2	Retimer.....	116
<b>Appendix A</b>	<b>117</b>	
A.1	Windows* Registry for Graphics .....	117
A.1.1	Registry Batch.....	117



## Figures

Figure 1:	Digital Signage .....	10
Figure 2:	Industrial Automation.....	10
Figure 3:	Point of Sales (POS).....	10
Figure 4:	Display Architecture Overview .....	15
Figure 5:	Various Planes .....	16
Figure 6:	Example Tiger Lake-UP3 Embedded SKU DDIs and TCPs.....	18
Figure 7:	DisCon LFP Page .....	18
Figure 8:	DisCon Showing Port Limitation .....	19
Figure 9:	Example of Loading .bin and .json Files.....	21
Figure 10:	DisCon Tool Main Page .....	21
Figure 11:	Enabling Embedded Platform Option.....	22
Figure 12:	Example LFP on a Notebook .....	23
Figure 13:	eDP Overview .....	24
Figure 14:	Typical eDP Panel Connector.....	25
Figure 15:	eDP on Intel® CRB.....	26
Figure 16:	Example Commercial eDP Connector .....	26
Figure 17:	PSR Overview .....	29
Figure 18:	Enable PSR Feature in VBT .....	30
Figure 19:	PSR Options in VBT .....	30
Figure 20:	MSO 2x1 .....	31
Figure 21:	MSO 2x2 .....	32
Figure 22:	Chip-On-Glass.....	33
Figure 23:	eDP Bifurcation .....	33
Figure 24:	VBT AUX Channel Configuration .....	34
Figure 25:	LVDS Overview .....	35
Figure 26:	SPWG Data Format Example (18bpp Dual Channel) .....	37
Figure 27:	Example LVDS Panel-side Connector .....	37
Figure 28:	Example LVDS Cable.....	38
Figure 29:	MIPI-DSI Overview .....	39
Figure 30:	Enabling MIPI-DSI in DisCon on Gen9 or earlier platforms .....	39
Figure 31:	Enabling MIPI-DSI in DisCon on Gen11+ platforms.....	40
Figure 32:	Flow Diagram.....	41
Figure 33:	Panel specific settings .....	41
Figure 34:	MIPI Sequence Tool.....	42
Figure 35:	Modify the GPIO Resources ID, Number and Value.....	43
Figure 36:	Saving VBT File .....	44
Figure 37:	MIPI FPC Cable.....	45
Figure 38:	Example EFP .....	46
Figure 39:	DP Overview.....	47
Figure 40:	Standard DP system connector .....	48
Figure 41:	Standard DP Cable Connector .....	48
Figure 42:	Mini DP Cable Connector .....	48
Figure 43:	VBT Link Rate Settings.....	50
Figure 44:	Active Adapter Overview .....	52



Figure 45:	Example Active Adapter .....	52
Figure 46:	Passive Adapter Overview .....	53
Figure 47:	Example Passive Adapter .....	53
Figure 48:	VBT Option for Enabling Dual-mode .....	54
Figure 49:	Example DP to VGA Adapter .....	55
Figure 50:	Type-C System Connector (Female) .....	55
Figure 51:	Type-C Connector Overview .....	56
Figure 52:	USB Type-C Signals .....	56
Figure 53:	VBT Settings for Enabling DP over Type-C .....	57
Figure 54:	MST Hub .....	57
Figure 55:	MST Daisy-Chaining .....	58
Figure 56:	Horizontal 6x1 MST Configuration .....	60
Figure 57:	Vertical 2x3 MST Configuration .....	60
Figure 58:	Example of DP-IN on Tiger Lake-H .....	61
Figure 59:	HDMI Overview .....	63
Figure 60:	Standard HDMI System Connector .....	64
Figure 61:	Standard HDMI Cable Connector .....	64
Figure 62:	Mini HDMI Cable Connector .....	64
Figure 63:	Micro HDMI Cable Connector .....	64
Figure 64:	HDMI Cables/Connectors Size Comparison .....	64
Figure 65:	LSPCON* .....	67
Figure 66:	Difference in Device Organization between DP-HDMI and HDMI LS .....	68
Figure 67:	DVI Variations .....	69
Figure 68:	DVI System Connector .....	70
Figure 69:	DVI Cable Connector .....	70
Figure 70:	DVI to DVI and VGA Splitter .....	70
Figure 71:	VGA System Connector .....	71
Figure 72:	VGA Cable Connector .....	72
Figure 73:	Set "Panel Type #" .....	74
Figure 74:	Generic DTD Timings Selection .....	74
Figure 75:	DTD Entries .....	75
Figure 76:	Setting "Embedded Platform" Option .....	76
Figure 77:	Disabling EDID Read .....	77
Figure 78:	Set "Panel Type #" .....	77
Figure 79:	Generic DTD Timings Selection .....	78
Figure 80:	DTD Entries .....	79
Figure 81:	EFP EDID-less Option .....	80
Figure 82:	EFP DTD Entries .....	80
Figure 83:	Typical Horizontal Collage Display .....	81
Figure 84:	Typical Horizontal Collage Display .....	82
Figure 85:	Arrange Displays and Enable "Combined Desktop" or "Collage" .....	84
Figure 86:	Combined Desktop Enabled in Horizontal .....	85
Figure 87:	Combined Desktop Enabled in Vertical .....	86
Figure 88:	Combined Desktop Enabled in Vertical .....	87
Figure 89:	Bezel correction example .....	88
Figure 90:	Combined Desktop Option in IGCC .....	89



Figure 91:	Config option for Combined Desktop .....	90
Figure 92:	Collage/Combined Desktop Config Options.....	90
Figure 93:	Two Pipe Two Port 8K.....	91
Figure 94:	Two Pipe One Port 8K .....	92
Figure 95:	Example Remote Headless Application.....	95
Figure 96:	KVMR Setting in DisCon.....	95
Figure 97:	HDR Stack .....	98
Figure 98:	Where to Enable HDR in Windows*.....	100
Figure 99:	Example HDR Enabled.....	101
Figure 100:	Example DP Data Flow (from Display Audio SPS).....	102
Figure 101:	One display (Device by type) .....	104
Figure 102:	One display (Device by connection).....	104
Figure 103:	Example Discrete Graphics Cards .....	106
Figure 104:	Concurrent iGPU/dGPU Diagram.....	107
Figure 105:	Example Concurrent iGPU/dGPU .....	108
Figure 106:	Switchable/Hybrid iGPU/dGPU Diagram .....	109
Figure 107:	HPD in VBT .....	112
Figure 108:	Video Signal Degradation .....	114
Figure 109:	Redriver in DisCon.....	115
Figure 110:	Redriver Settings.....	115

## Tables

Table 1.	Terminology.....	11
Table 2:	eDP Max Resolution by Platform.....	27
Table 3:	eDP Bifurcation AUX Port Example .....	34
Table 4:	MIPI Sequence Type.....	43
Table 5:	DP Maximum Resolution by Platform .....	50
Table 6:	Example MST Calculations .....	59
Table 7:	HDMI Maximum Resolution by Platform.....	65



## Revision History

Date	Revision	Description
May 2021	1.0	Content additions and updates.
September 2020	0.8	Initial release.

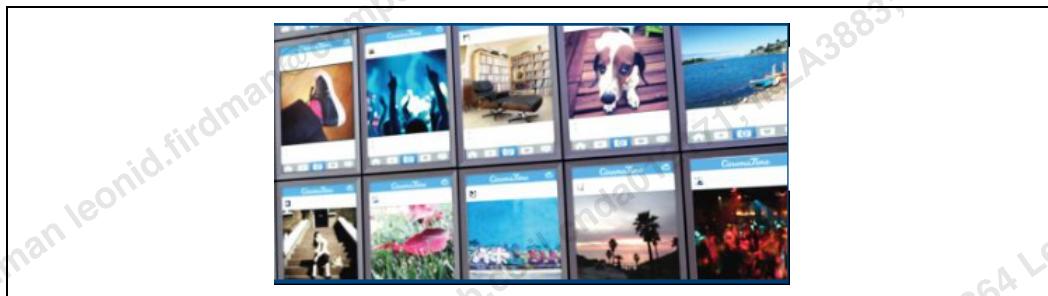
§

## 1.0 **Introduction**

---

The world of embedded applications in IoT (Internet of Things) encompasses a wide and diverse set of applications. These range from sensor chips to full “PC-like” implementations. The wide-ranging applications include systems for Digital signage, Point of Sale (POS), ATMs (Automated Teller Machines), Industrial Automation, portable medical devices or more. A vast majority of these require a display of some sort for a user to interact with the functionality of the device. To address these needs, Intel<sup>®</sup> embedded computing products provide a wide range of display capabilities.

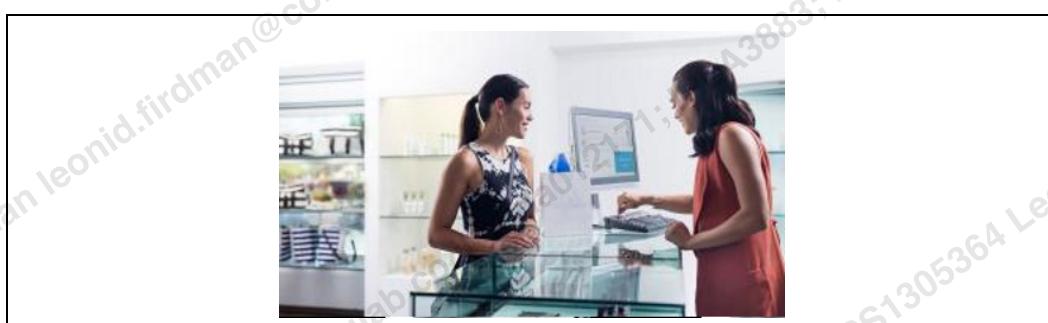
**Figure 1:** **Digital Signage**



**Figure 2:** **Industrial Automation**



**Figure 3:** **Point of Sales (POS)**





## Introduction

In this document, we intend to help you understand these display options, provide tips for choosing the best ones, and generally provide tips to successfully implement an embedded application with display capability.

We will cover:

- Integrated Graphics Processing Unit (iGPU) display technology overview
- Tools for configuring graphics firmware and/or drivers
- Local flat panels and related topics
- External monitors and related topics
- Operation of Discrete GPUs (dGPU) along with iGPU

Along with general information, we will provide information on trouble areas of implementing embedded displays and some common solutions that we can provide. We will also discuss third party hardware used with the Intel® graphics and how to get support for those products.

In the end, we expect you will have a better idea of the general capabilities that Intel® provides and will be more successful implementing your product.

## 1.1 Terminology

**Table 1. Terminology**

Term	Description
AC	Alternating Current
ACX	Audio Class Extension
ADB	Automatic Display Brightness
ALS	Active Level Shifter – used to boost the 3V signals on a port output to 5 V signals required for HDMI or DVI.
Apollo Lake	Intel Atom® Processor E3900 Series, and Intel® Pentium® and Celeron® Processor N- and J-Series.
AUX	Auxiliary
BYT	Intel Atom® Processor E3800 and Z3700 Series, and Intel® Pentium® and Celeron® Processor N- and J-Series.
CEC	Consumer Electronics Controller
Coffee Lake	8 <sup>th</sup> Generation Intel® Core™ Processor Family and Intel® Xeon® Processor E-2100M Family.
Comet Lake	10 <sup>th</sup> Generation Intel® Core™ Processor Family Based on the S-Processor Line and Intel® Pentium® Processors.

Term	Description
CRB	Customer Reference Board
CRLS	Cost Reduced Level Shifter
CTS	Compliance Test Specification (DP and HDMI)
DAC	Display Audio Codec
DDC	Display Data Channel
DDI	Display Driver Interface
DG	Discrete Graphics
dGPU	Discrete GPU
Dongle	A small interface device able to be connected to and used with a computer. Example: DP to HDMI adapter.
DP	Display Port
DPST	Display Power Saving Technology
DRM	Digital Rights Management
DSC	Display Stream Compression
DVD	Digital Versatile Disc
DVI	Digital Visual Interface
DWM	Desktop Windows* Manager
EDID	Extended Display Identification Data
eDP	Embedded Display Port
EDS	External Design Specification
EFP	External Flat Panel
GPU	Graphics Processing Unit
HBR	High Bit Rate
HDCP	High bandwidth Digital Content Protection
HDMI	High Definition Multimedia Interface
HPD	Hot-Plug Detect – GPIO used to detect a that a monitor is plugged in or removed.
IBV	Independent BIOS Vendor

Term	Description
iGPU	Integrated Graphics Processing Unit
Kaby Lake	7 <sup>th</sup> Generation Intel® Core™ Processor Family
LFP	Local Flat Panel
LSPCon*	Level Shifter and Protocol Converter
LVDS	Low-Voltage Differential Signaling
MSO	Multi SST Operation
MST	Multi Stream Transport
NA or N/A	Not Applicable
PC	Personal Computer
PCB	Printed Circuit Board
PCH	Platform Controller Hub
POS	Point of Sale
PSR	Panel Self Refresh
RFB	Remote Frame Buffer
RLS	Retiming Level Shifter – used to resynthesize display output signals and boost them when necessary.
RVP	Reference Validation Platform
SDP	Secondary Data Packet
Skylake	6 <sup>th</sup> Generation Intel® Core™ Processor Family
SST	Single Stream Transport
TBD	To Be Determined
Tiger Lake	11 <sup>th</sup> Generation Intel® Core™ Processor Family
TMDS	Transition Minimized Differential Signaling
TV	Television
UMA	Unified Memory Architecture
VBT	Video Bios Table – a table of settings stored in the bios image that provides many of the configuration details for pre-boot firmware (VBIOS/GOP) and the graphics driver (post-boot). It is edited with the older BMP or newer DisCon tool.



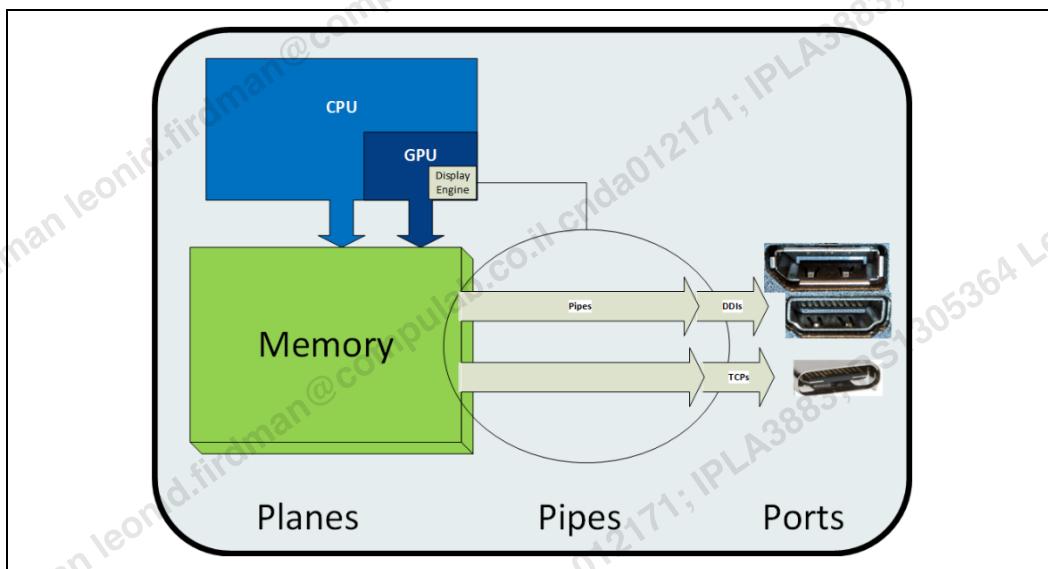
Term	Description
VDSC	VESA Display Stream Compression
VESA	Video Electronics Standards Association
VGA	Video Graphics Array
Whiskey Lake	8 <sup>th</sup> Generation Intel® Core™ U-Series processors
Intel® WiDi	Intel® Wireless Display, which has been replaced by Miracast*

§

## 2.0 Display Architecture Overview – Planes, Pipes and Ports

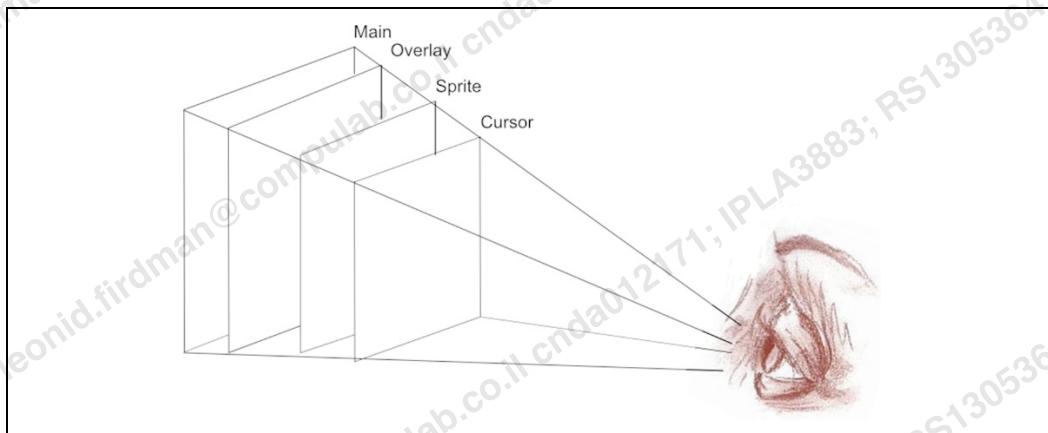
Integrated Graphics Processing Units (iGPU), as Intel® provides in many Intel® processor families, comprises multiple functional areas. Since we are primarily discussing display/monitor capabilities in this document, we will stick to the pertinent areas of the iGPU that are relevant to displays, such as the display engine contained in the iGPU. The display engine in the iGPU provides quite a complex range of capabilities. Fortunately, it can be broken down into straight forward basic functional blocks of Planes, Pipes, and Ports as shown in the following figure:

**Figure 4: Display Architecture Overview**



### 2.1 Planes

Display Planes are regions of main memory allocated to image display data. This includes the main framebuffer where most of the display data resides, but also includes the number of other specialty planes used for specific capabilities, as shown in the following example figure:

**Figure 5: Various Planes**

- Main is the main framebuffer (or framebuffers) for the display where most of the data displayed is manipulated by the CPU and/or GPU.
- Overlay is often used for displaying video allowing the video decode engine to decode directly to a displayable window on the display.
- Cursor can display a hardware generated cursor that overlays the main plane without having to draw the cursor into the main frame as it moves.
- Sprite can display small icon-like objects over the top of Main to enable simple animations or games using small graphical objects that are easy to move on the screen without having to redraw the entire Main plane it sits on.

Refer to the EDS for your specific Intel® processor family for details on which planes are provided on the product you are using.

## 2.2

## Pipes

Display Pipes are the mechanism where the data from the planes (memory) are assembled into a displayable data stream and sent out to an assigned port. They are assigned to and use the port's DDIs (or TCPs for Type C ports) to connect to the port and out to the display. The pipe is where the data timing (including resolution), pipe scaling, color space conversions and any other manipulations or touch-ups of the display information happen. The number of pipes available on the specific Intel® processor family determine the number of simultaneous displays that product supports. If the product you are using has two pipes, then only two simultaneous displays can be enabled. Three pipes can enable three displays, four pipes can enable four displays, and so on. The pipes are assigned to a port via display detection via the "Child Device" table ordering and assigned and configured to the configuration determined by the port DDI configuration. Some pipes can only be used with a specific port – for example on some Intel® processor families, a pipe (or DDI) may only be used with an eDP port, or a TCP pipe can only be used with a Type C port. Keep in mind that for some Intel® product families, higher resolution modes may require the use of

multiple pipes and that decreases the number of simultaneous displays that can be enabled. Also, some products provide "bifurcation" capabilities that adds to the number of simultaneous displays that can be displayed- for example the pipe to the eDP port can be split to support a lower resolution eDP panel and an eDP to VGA converter simultaneously. Refer to the EDS for the specific product you are using to see if this capability is supported on the product you are using.

## 2.3

### Ports

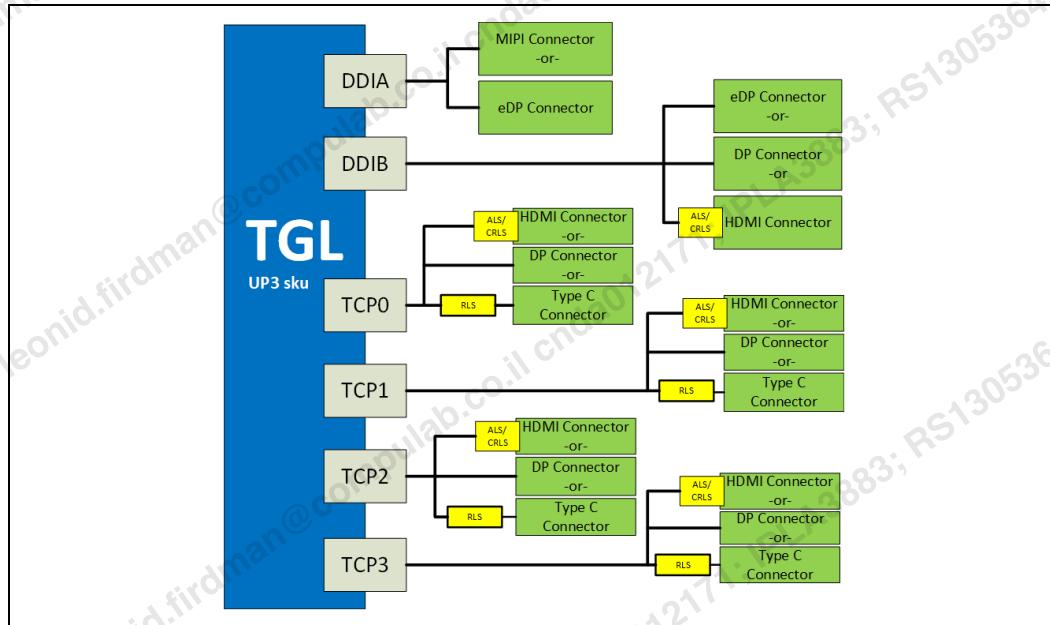
Display Ports are the physical outputs from the system that provides the data to the monitor. It converts the data from the digital signals coming in on the DDIs (or TCPs) to the physical electrical signals for the outside display device. The connectors are typically in a format defined for the type of display connection to which they attach as configured by the DDI interface to the port. There is usually more than one style of connector for each type and range from the more typical full-size form factor down to mini and micro versions for space sensitive applications. Details and examples are shown in the sections of this document discussing the different interfaces. An important thing to keep in mind is that you can have more DDIs/Ports implemented than there are Pipes for the particular Intel® product family, but it is the number of pipes that determines the number of ports than can be active simultaneously. Also, you can have more DDI's than actual ports. For example, on Elkhart Lake, DDI-A/DDI-D are muxed to Port A and DDI-A/DDI-B are muxed to Port B. You should refer to your specific Intel® processor family's EDS to see which types of ports are supported and their capabilities.

With this basic overview of the internal display handling path understood, we can move on to the main topics for embedded display technologies in the rest of this document.

## 2.4

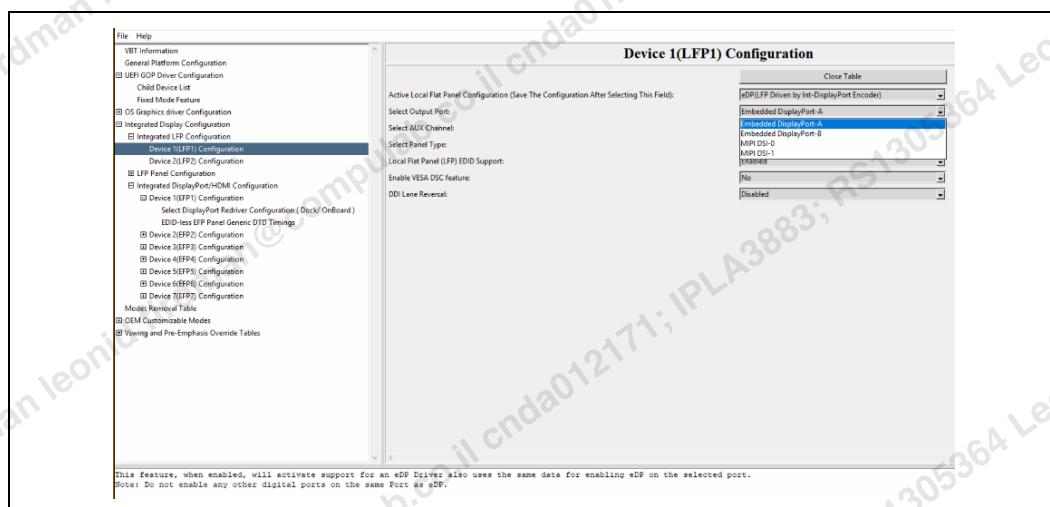
### Display Driver Interface (DDI)

The characteristics for DDI (or TCP for ports that can be used for Type C Alt DP output) are especially important to have in mind as you design and/or configure your platform. They are the "connector" between the Pipe and the physical port for the monitor. They are configured for specific output types (i.e., eDP, DP, HDMI, or more.), depending on the Intel® processor family you are using, there may be limitations on how particular DDIs (or TCPs) can be used. The following figure shows the flexibility that the Tiger Lake product provides for DDI configuration:

**Figure 6: Example Tiger Lake-UP3 Embedded SKU DDIs and TCPs**


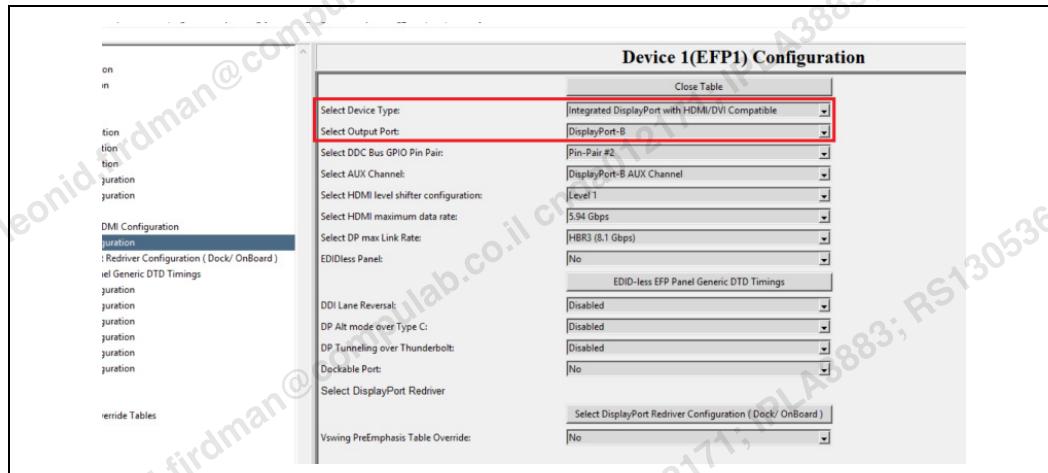
The DDIs typically can be configured as a "traditional" display ports (eDP, DP, HDMI, or more.) and TCPs are typically used for "DP Alt" output on a Type C connector or for traditional DP.

In some processor families, or specific SKUs in the family, the DDI known as DDI-A (or sometimes "Embedded DisplayPort-A" in some product families) may only be configured as an eDP, and sometimes MIPI port – refer to the EDS for your particular part. The VBT setting via the DisCon tool for your specific platform shows how the DDI/TCP channels can be allocated, as shown in the following screenshot:

**Figure 7: DisCon LFP Page**


Other DDIs may have other limitations on the type of EFP it can be (e.g. DP or HDMI). This depends on the Intel® processor family you are using and will be detailed in the EDS (and VBT settings) for the specific processor you are using.

**Figure 8: DisCon Showing Port Limitation**



If you will be using a DDI/TCP for DP output, you may also have the option of setting it for "DP with HDMI/DVI compatibility". This allows you to implement the DP port, but then be able to use an inexpensive passive dongle to also be able to get HDMI or DVI capability via the dongle. The DP port actually outputs HDMI or DVI signals but they need the dongle to tell the port to use HDMI/DVI signals and to level shift those signals to proper HDMI/DVI voltage levels. The DP ports can only drive 3V signals but HDMI/DVI need higher voltage levels to work.

With this basic overview of the configuration of the internal display handling path, we can move to the main topics for embedded display technologies in the following sections of the document.

§



## 3.0 DisCon Tool

The Display Configuration tool (DisCon) is an update of the BMP tool for configuring various graphics and display options in the binary file table (VBT) stored in the BIOS image firmware. It modifies the binary data based upon the responses and settings given by the user on the user interface. The binary table contains information about the configuration of the underlying hardware on which the display firmware/driver runs with the tool. You can modify the configuration present in the binary table when the underlying hardware configuration changes, thus decoupling the driver/firmware from the hardware. Refer to the DisCon Tool User Guide, also available on the Intel® RDC website, for in depth details on using the tool.

Once you have configured the binary table, you will need the ability to merge (or "stitch") the updated table into your UEFI BIOS image and then program (i.e. "flash") it into your system. DisCon does not provide this capability- it just configures the binary table that must be in the firmware.

The merge capability is different from BIOS to BIOS and the tools to merge must be obtained from your BIOS provider. The programming process is system specific and usually you will need to have designed your flash memory to provide this capability.

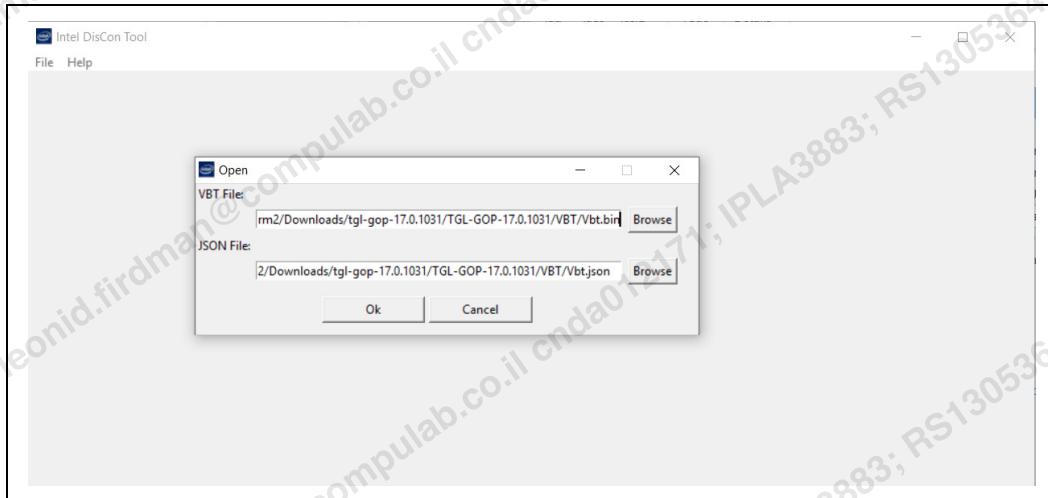
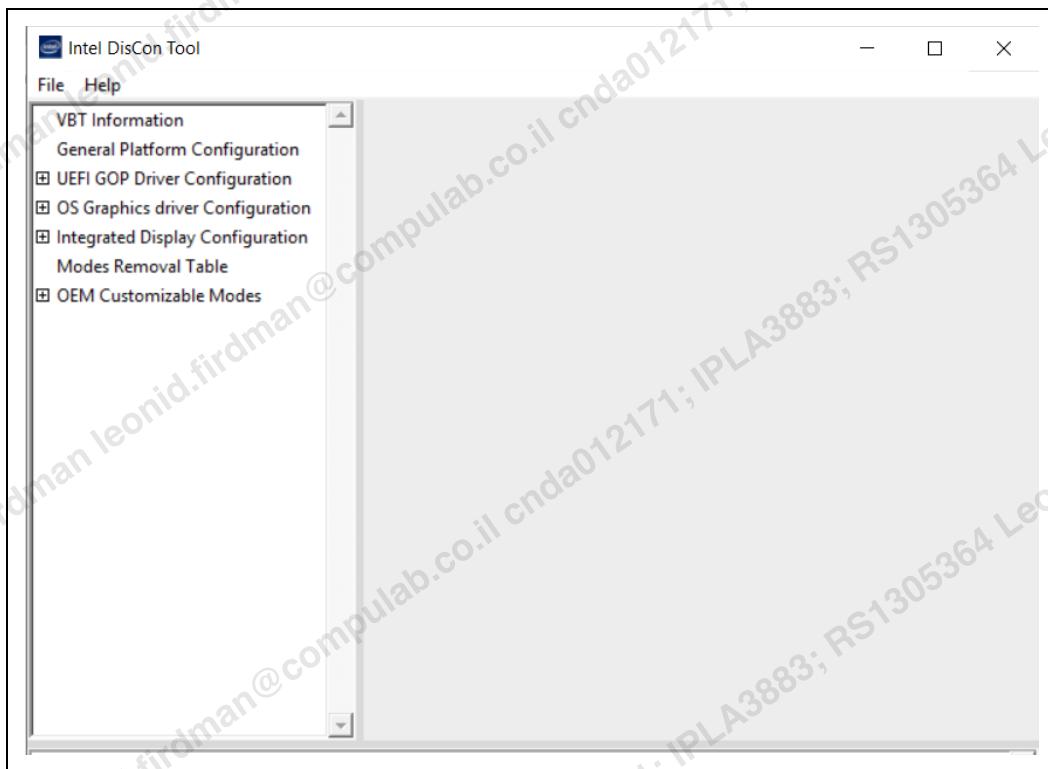
### 3.1 Obtaining a VBT

The default VBT is provided along with the GOP (or VBIOS if you are using older products and much older OS like Windows® OS 7). You might also be able to get the current VBT from your system by using your BIOS specific merge tool. Your BIOS vendor should provide specific instructions on this step.

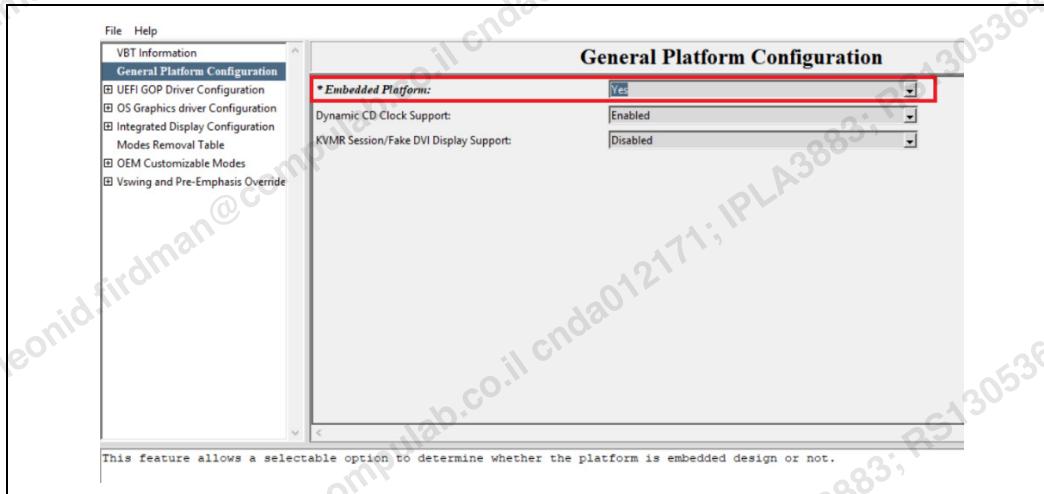
### 3.2 Using DisCon

The first step is to obtain it (typically from the Intel® RDC site), install, and launch the DisCon tool.

Next, you will need to load your processor family specific JSON file (was .bsf with the old BMP tool) and the appropriate VBT.

**Figure 9: Example of Loading .bin and .json Files****Figure 10: DisCon Tool Main Page**

**Note:** At this point, you likely will want to open the "General Platform Configuration" page and make sure that the "Embedded Platform" option is set to **YES** in order to enable all the normally hidden embedded options in the VBT. If you need to set it to **YES**, be sure to do a **File** and then **Save** the file you are working on for the change to take effect and enable the embedded features.

**Figure 11: Enabling Embedded Platform Option**

At this point, any configuration changes to the VBT for your firmware and driver specific to your platform and requirements can be made. The rest of the document will highlight some (but not all) of the different options and/or values that you may need to tune.

§

## 4.0 Local Flat Panels (LFPs)

Local Flat Panels are displays that are comprised of a “raw” LCD panel being driven directly by the iGPU. This is usually through eDP or MIPI. They are differentiated from normal external displays as they require special handling as compared to a typical HDMI or DP monitor. The raw LCD has the following requirements:

- The port must handle power-up and power-down timing.
- The port usually must control the backlight and its dimming.
- LCDs are typically run at fixed resolution, so if a different resolution mode is used with them, the graphics needs to handle scaling the image to remain full screen.

LFP is the display used on a typical notebook computer.

Figure 12: Example LFP on a Notebook



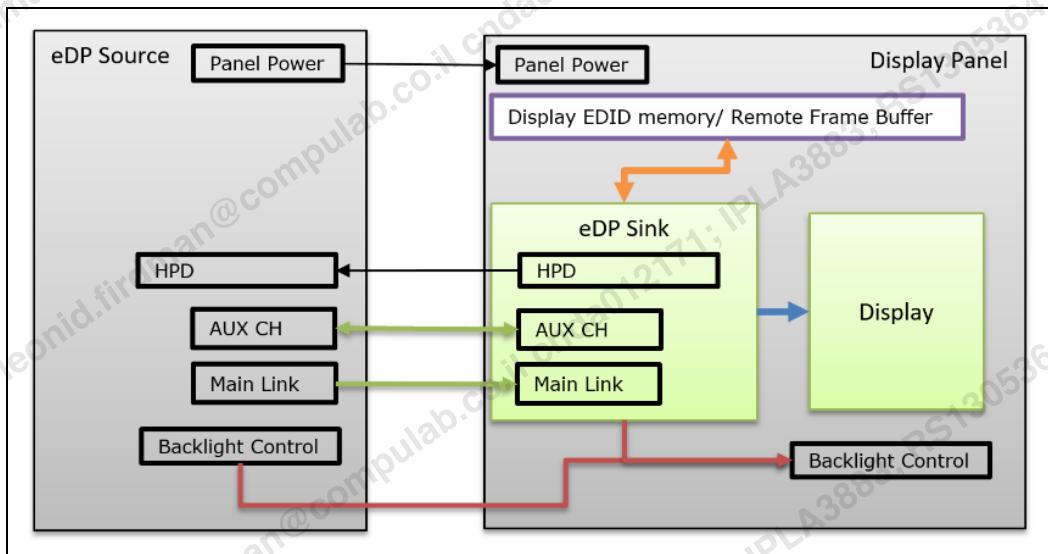
The following sections will discuss topics related to LFPs.

### 4.1

## Embedded Display Port (eDP)

The Embedded DisplayPort\* (eDP\*) is an embedded version of the DisplayPort standard oriented towards applications such as notebook and All-In-One PCs. Like DisplayPort, embedded DisplayPort also consists of a Main Link, Auxiliary channel, and an optional Hot-Plug Detect signal. Note that HPD is not utilized by Intel® graphics driver. eDP\* can be bifurcated (except for U-Processor Line) in order to support VGA display (refer to the [eDP Bifurcation](#) section of this document).

Figure 13: eDP Overview



Something to keep in mind with eDP ports is that only the designated LFP port(s) on its designated DDI(s) on the Intel® product family you are using can be treated as an LFP. If you use a DP to eDP converter on a DP port, you will not have the features that are provided on the native eDP port such as:

- Software Brightness control (backlight) from display control panel (i.e. CUI or IGCC or Windows\* Display Settings).
- Assertive Display Technology (ADT) which includes ADB.
- Software Contrast control.
- iGPU scaling (when the mode selected does not match the panel native resolution).
- Power on/off timing (panel specific).
- LFP pipe specific power saving features such as DPST. (refer to the EDS for the specific Intel® product family for details).

#### 4.1.1

#### Advantages – eDP

The following are the advantages of eDP:

- Requires fewer signals
- Connectors are smaller
- Requires lower power
- Reduced electromagnetic radiation
- Longer connection lengths are allowed vs LVDS.
- Often support EDID for plug and play support by firmware and driver.

## Local Flat Panels (LFPs)

- Standardized (VESA) data format makes interchanging panels easy.

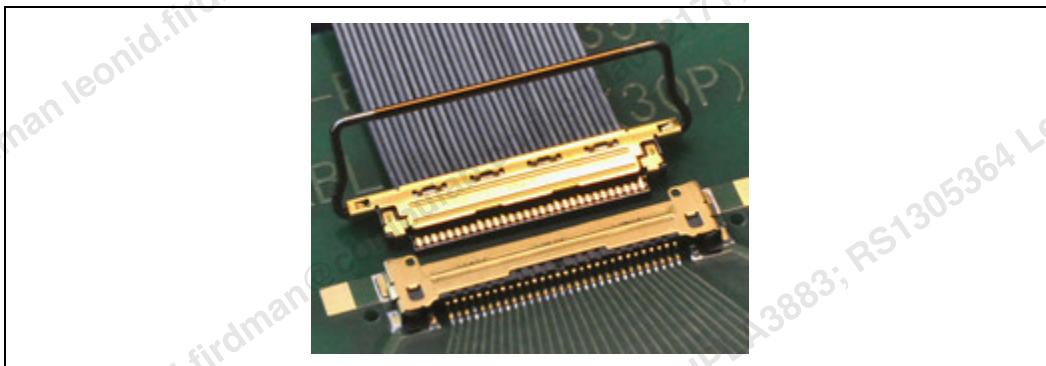
### 4.1.2 Disadvantages – eDP

- There are no standardized connectors for the motherboard, making interconnect cables difficult to design.
- Hot-plug is not supported.
- Cannot enable collage display mode.
- Some platforms do not allow cloning.

### 4.1.3 Cable/Connector – eDP

Example eDP connection on a typical eDP panel:

**Figure 14: Typical eDP Panel Connector**



Cable connector used on an Intel® RVP / CRB motherboard:

**Figure 15: eDP on Intel® CRB**

eDP connector used on some embedded platforms:

**Figure 16: Example Commercial eDP Connector**

#### 4.1.4

#### Troubleshooting – eDP

- For the most part, eDP just works, especially if you use a panel that provides an EDID.
- If you do not use an eDP panel with an EDID, then you will need to configure it for EDID-less operation. Refer to the [EDID-less Displays](#) section of this guide.
- The Intel® firmware (GOP) and driver have a default power sequence timing built in. Some panel manufacturers require longer times for the panel to settle after an eDP port power off and on, thus you may need to adjust this timing in the VBT as per the panel manufacturer's recommendations. The symptoms of incorrect panel power

timing may include black screen, display jitter, display artifacts, or more, especially after a mode-set or sleep/resume cycle.

- The eDP port has a defined backlight control that is connected to a pre-determined GPIO pin and normally just works. If you change the default backlight GPIO on your design, you will need to appropriately adjust the VBT in the LFP section of the VBT configuration.
- Backlight control also may require tuning of the PWM settings base on the requirements of the panel and/or application. The range of the PWM value set in the VBT may need to be adjusted based on the panel's datasheet requirements.
- If you have connected your eDP panel to a non-native port (to a DP port) the LFP will be treated as an EFP and will not have LFP features as discussed before. It is best to use the LFP port for LFP devices.
- For platforms that support multiple eDP ports, it is recommended to use the primary eDP port (in most cases DDI-A) to enable a single eDP. For example, Tiger Lake supports dual eDP, but single eDP is supported only on DDI-A since it is not POR on DDI-B. However, if you have two eDPs, then both DDI-A and DDI-B can be used. This requirement could change in the future, so please refer to the POR document of that specific platform.

#### 4.1.5 Maximum Resolution Support – eDP

The following are the known maximum resolutions for the EmbeddedDisplayPort\* interface for the latest announced Intel® processor families at the time of publication of this document.

**Note:** Refer to the EDS and Spec Updates for any changes to the maximum resolutions that can be supported.

**Table 2: eDP Max Resolution by Platform**

Platform	Spec version	Link rate	Maximum resolution	Maximum resolution with DSC
Skylake	1.3	HBR2	4096x2304 @ 60Hz 24bpp*	NA
Kaby Lake-S	1.4	HBR2	4096x2304 @ 60Hz, 24bpp*	NA
Coffee Lake	1.4	HBR2	4096x2304 @ 60Hz 24bpp*	NA
Whiskey Lake-U	1.4	HBR2	3840x2160 @ 60Hz 30bpp 4096x2304 @ 60Hz 24bpp*	NA
Tiger Lake	1.4b	HBR3	4096x2304 @ 60Hz 36bpp 5120x3200 @ 60Hz 24bpp 7680x4320 @ 60Hz 24bpp	5120x3200 @ 120Hz 30bpp 7680x4320 @ 60Hz 24bpp

Platform	Spec version	Link rate	Maximum resolution	Maximum resolution with DSC
Apollo Lake	1.3	HBR2	4096 x 2160 @ 60hz*	NA
Elkhart Lake	1.3	HBR2	4096 x 2160 @ 60hz*	NA

\* For Gen9 iGPU platforms, there is a special ultra-wide 5K mode that can be supported with limitations. These platforms can support 5120x1440 60 Hz, however, some graphics composition features are not supported in hardware and will need to be handled using slower DWM composition in Windows\*. 5K resolution support if not required can be disabled using the inf key "DisableSinglePipe5kModeSupport"

## 4.1.6 Panel Self Refresh

The Panel Self Refresh feature enables system-level power savings when the displayed image remains static for multiple display frames. It was introduced in the eDP 1.3 spec in 2011 and has been available on Intel® integrated graphics since then. Always check the product specs for the specific product to make sure it is supported. The Sink device (display) must also support PSR for this to work.

### 4.1.6.1 How it Works

The Sink (display) stores a static image locally in the Remote Frame Buffer (RFB) within the Sink and displays this image from the RFB, while the eDP Main Link may be turned off. In order to work with PSR, the panel needs to have a RFB that is implemented according to PSR spec in eDP 1.3. The source can put the sink to PSR active state so panel will refresh image from its local frame buffer. The source needs to take the sink out of PSR active state, so it will display with current stream from source.

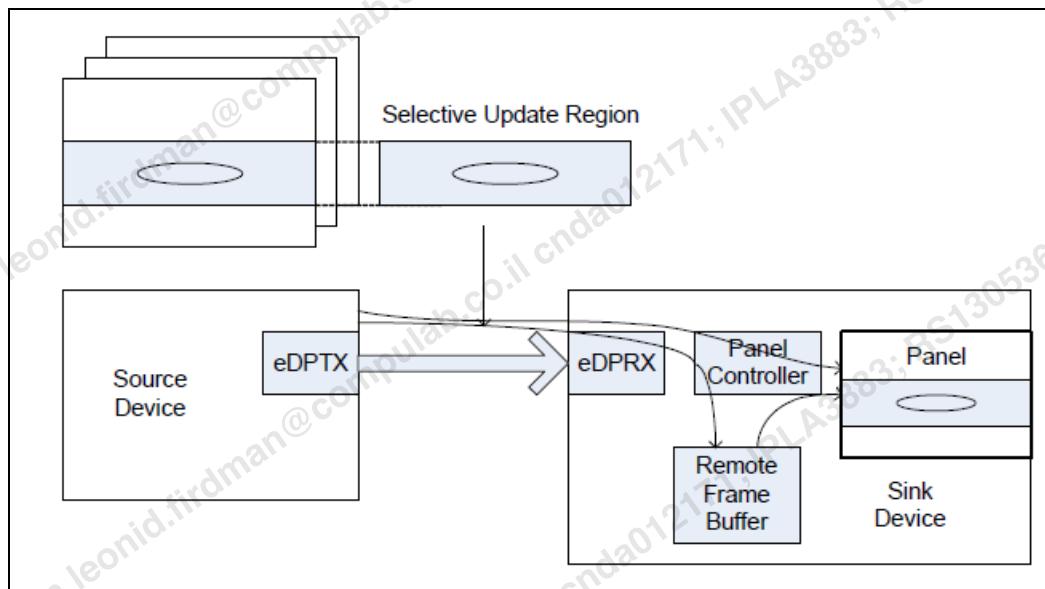
This is an optional feature for both Sinks and Sources. Some additional (smaller) power savings are also achieved by powering down the link to the display device. The source device is responsible for determining when the displayed image is static, sending PSR Source state indication to Sink using Secondary Data Packet (SDP), along with a complete active frame to capture. Source is also responsible for Source transmitter management. Sink is responsible for declaring its capabilities, providing current status, calculating CRC for captured static frame and generating interrupt for transmission or RFB errors to the Source. Sink is responsible for detecting PSR entry and exit events by tracking Source PSR state transitions. Upon detecting entry, Sink must capture the entire static frame into RFB and display from it until Source indicates PSR inactive state.

PSR2 is a superset of the PSR variants already implemented in earlier platforms. PSR2 provides additional power savings by allowing the transmission of modified regions within a video frame (when compared to the previous video frame) on the eDP link. A modified region is referred to as a “selective update region” in eDP v1.4.

A Source captures a full frame in the RFB, before transmitting the selective update region on the eDP link. The following figure shows a high-level conceptual diagram of

PSR2. Native AUX transactions and VSC SDP version #3 are used for transporting commands for PSR2 operation.

**Figure 17: PSR Overview**



#### 4.1.6.2 Potential Issues with PSR

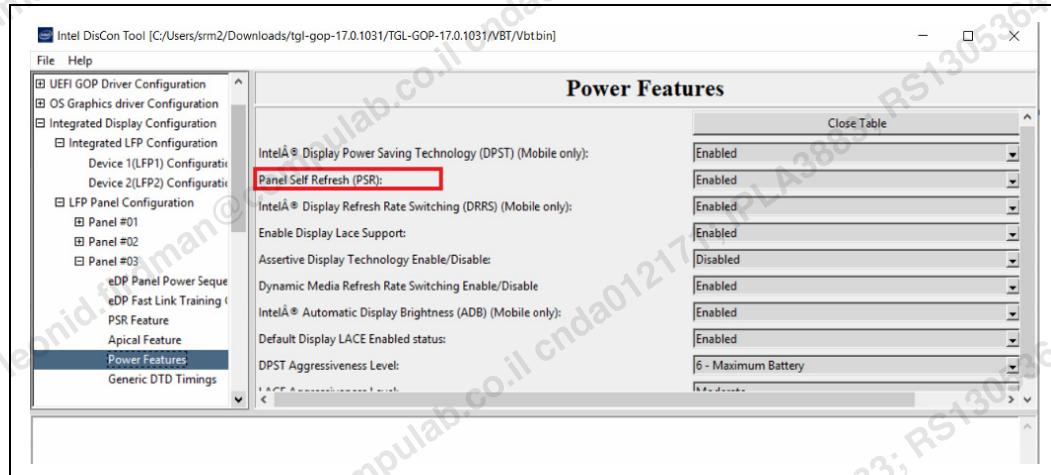
Display flicker – Some panels have the following inherent flicker issues when in PSR mode. If you have flicker, then the recommendation is to disable PSR for those panels.

- Standby/resume issues
- Panel timing problems

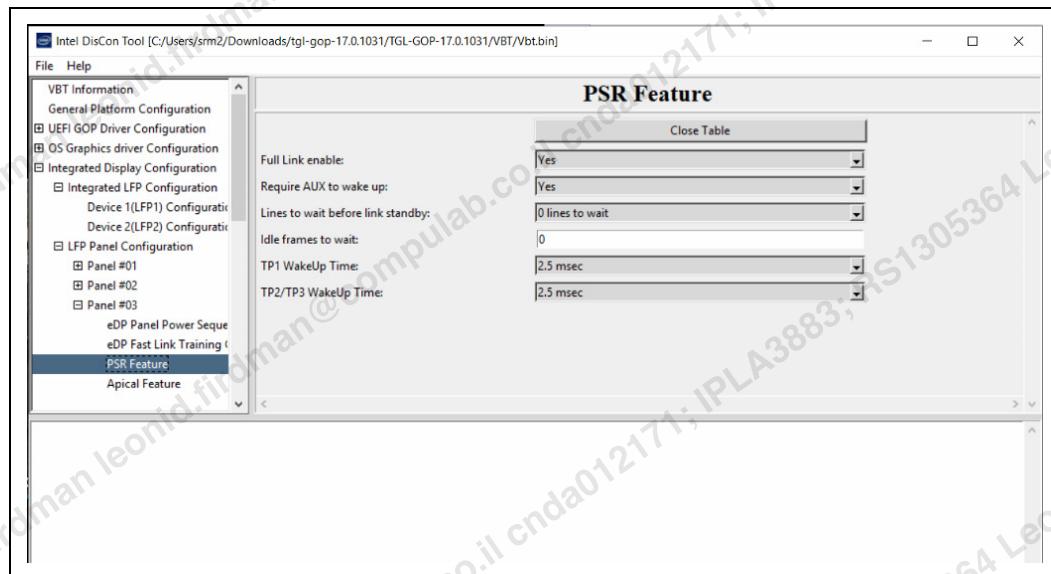
#### 4.1.6.3 Configuring PSR

PSR is configured via the VBT using the DisCon tool. The following screenshots show how to enable the feature:

**Figure 18: Enable PSR Feature in VBT**



**Figure 19: PSR Options in VBT**



How you set these options depend on the panel you are using. Refer to the panel specs and/or panel support provider for details as they should know how to use their panel.

#### 4.1.7

#### Multi SST Operation (MSO)

Multi-SST Operation (MSO) allows one eDP Source device to directly connect to multiple Panel Segments within a single eDP Panel, each separate eDP Main-Link receiver connected to each Panel Segment. The eDP Panel will still have one connection of AUX channel and one connection of HPD. Note that this feature is not supported on all Intel® processor families. Refer to the product specification for the product you are using to verify if this feature is supported.

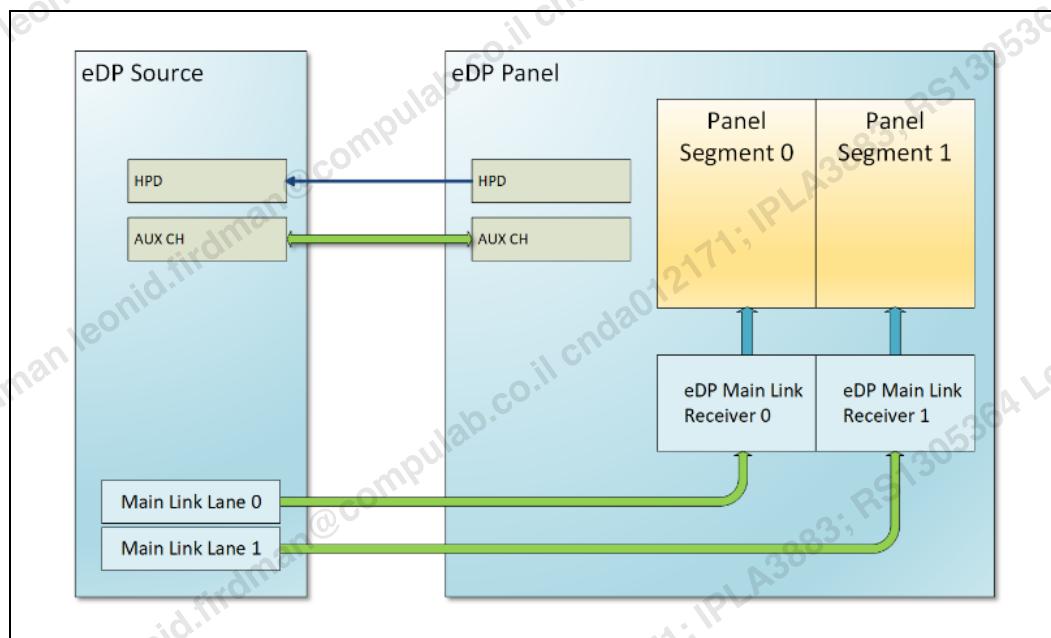
eDP 1.4 defines 3 configurations for MSO topology implementation.

- Two SST Links with one lane each (two lanes total), 2x1
- Two SST Links with two lane each (four lanes total), 2x2
- Intel® UHD Graphics implement MSO with 2x1 and 2x2 configuration.

The following figure illustrates the two application cases:

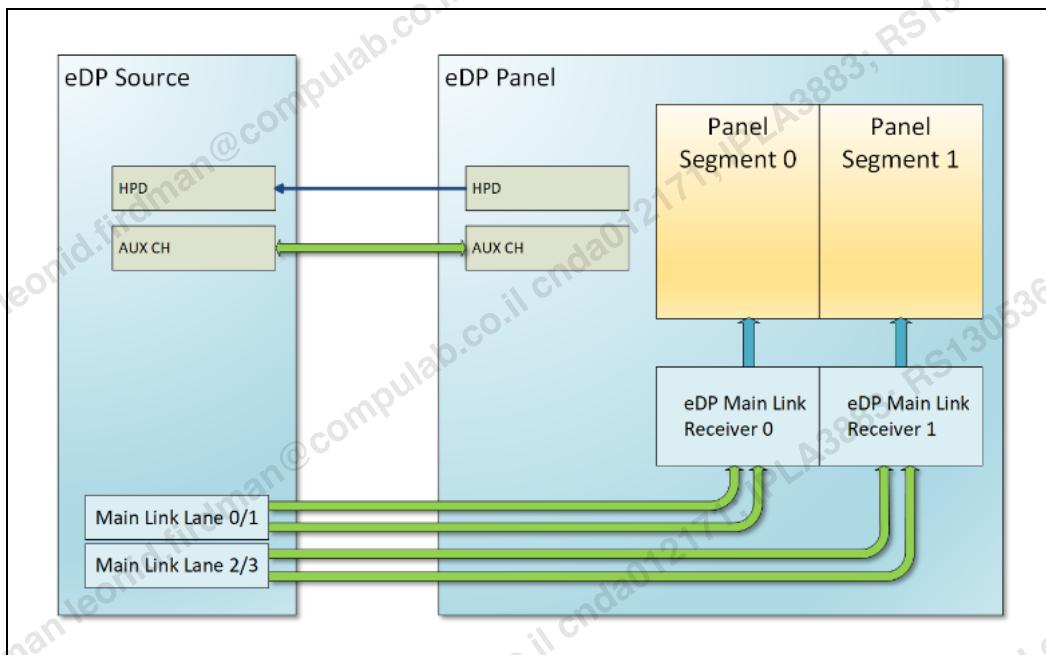
Two SST Links with one lane each (two lanes total), 2x1:

**Figure 20: MSO 2x1**



Two SST Links with two lanes each (four lanes total), 2x2:

**Figure 21: MSO 2x2**



GFX driver reads the **NUMBER\_OF\_LINKS** field in the **MSO\_LINK\_CAPABILITIES** (DPCD address 007A4h). If value of **NUMBER\_OF\_LINKS** = 2h or 4h, then GFX driver will configure MSO. The combination of **MAX\_LANE\_COUNT** (field in DPCD address 00002h) and **NUMBER\_OF\_LINKS** in the **MSO\_LINK\_CAPABILITIES** will determine which MSO lane options as:

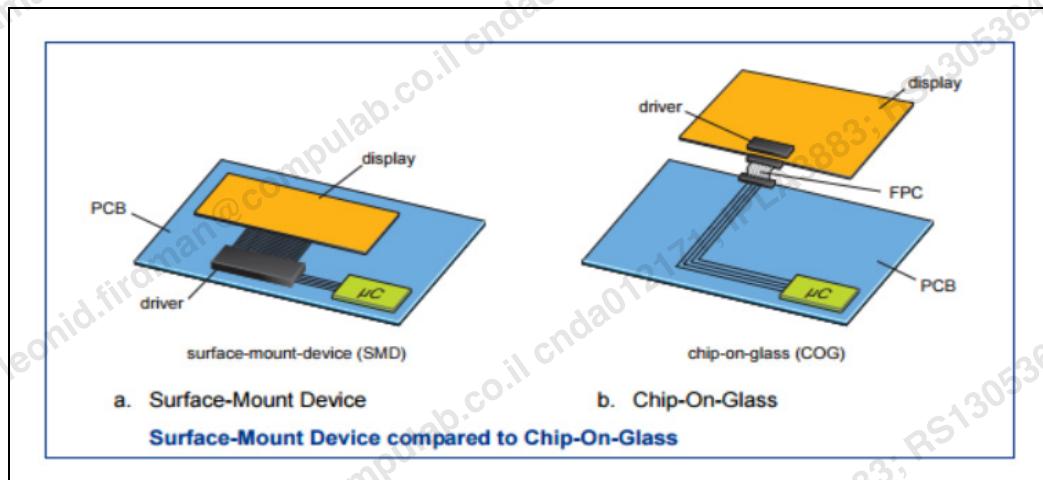
If **MAX\_LANE\_COUNT** = 2h and **NUMBER\_OF\_LINKS** = 2h, GFX driver is restricted to two MSO links, which one Lane for each SST Link (2x1).

If **MAX\_LANE\_COUNT** = 4h and **NUMBER\_OF\_LINKS** = 2h, GFX driver is restricted to two MSO links, which two Lane for each SST Link (2x2).

For detailed MSO definition, refer to VESA Embedded DisplayPort Specification rev 1.4.

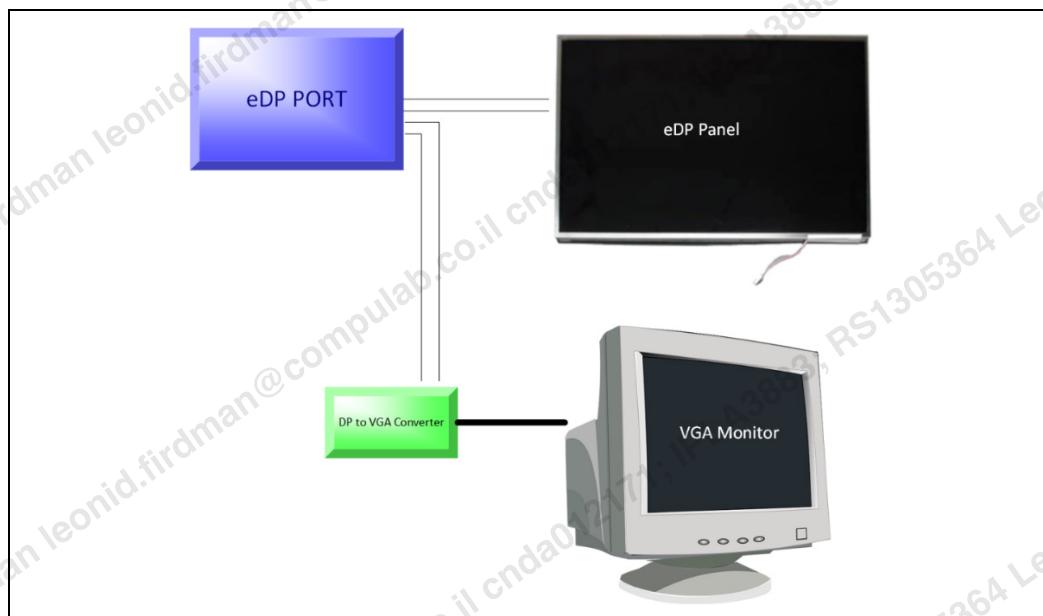
One of the MSO applications from Intel® UHD Graphics is Chip-On-Glass implementation on eDP. COG technology reduces the number of tracks and layers on the PCB, thickness of the Displays, cutting the board cost and complexity. The overall impact is reduction in system cost.

The LCD driver IC will basically be mounted on the display glass directly removing the interconnections between TCON and the driver (shown in the following figure).

**Figure 22: Chip-On-Glass**

#### 4.1.8 eDP Bifurcation

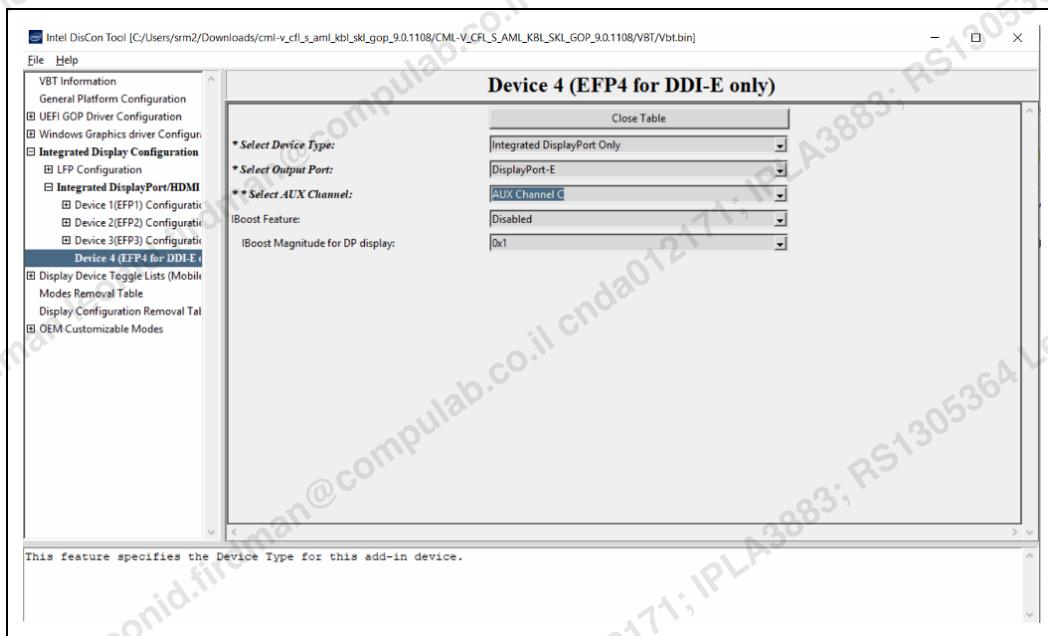
**eDP Bifurcation** is a feature that allows the eDP port to be split into two logical displays by splitting the data lanes of the eDP port so that two lanes can go to an eDP panel and the other two lanes can go to a DP-VGA converter allowing for a VGA port to be supported. Because the data lanes are split, that means the maximum resolution that can be supported on the eDP is decreased due to less data lanes available to the eDP. Refer to the EDS for your specific Intel® product for details.

**Figure 23: eDP Bifurcation**

Note that this feature is not included on all platform solutions. To date, it is included on Skylake, Kaby Lake, Coffee Lake and Comet Lake on the embedded roadmap support bifurcations. If you plan to use this feature, check the current EDS for the product you are using to be sure it is supported. For example, Tiger Lake does not support bifurcation and likely future products will not have support either.

When implementing bifurcation, you will need to configure an AUX channel to act as the DDC for the EDID support. You will need to pick an unused one or take one away from another port depending on your implementation. The following example shows how this can be done:

**Figure 24: VBT AUX Channel Configuration**



The following table shows an example where the eDP port is bifurcated and 1 HDMI and 2 DP's are implemented. When VGA is active, there can only be two other active displays for a platform that supports a maximum of 3 active displays. For example, if the display controller supports 3 active displays, with the bifurcation, you will only be able to have one additional active display along with the bifurcated eDP and DP-VGA active displays.

**Table 3: eDP Bifurcation AUX Port Example**

Device	Display	Port	AUX
LFP	eDP	A	A
EFP1	DP	B	B
EFP2	HDMI/DVI	C	N/A*

Device	Display	Port	AUX
EFP3	DP	D	D
EFP4	DP-VGA	E	C

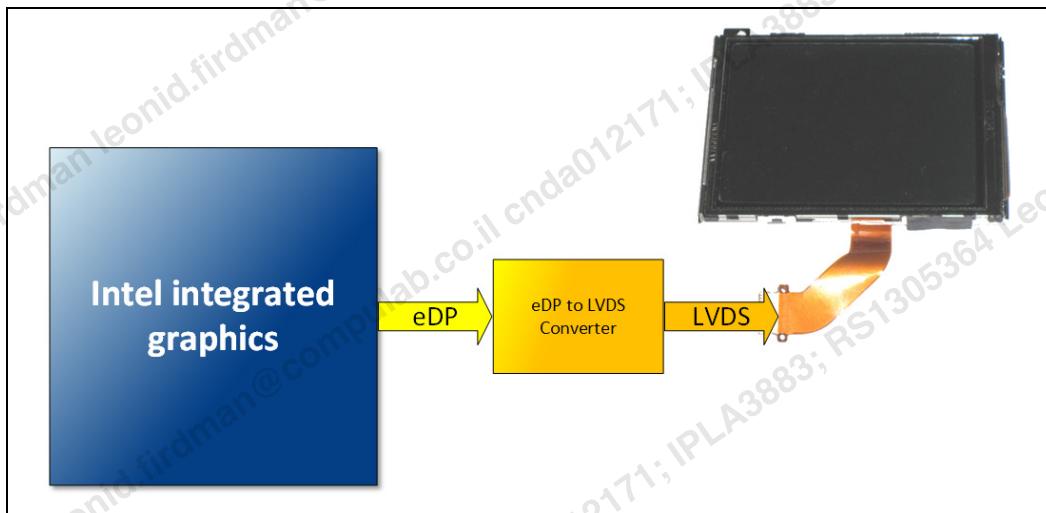
\*In this example, HDMI/DVI does not use an AUX channel and hence AUX-C can be used for EFP4 (DP-VGA). EFP2 cannot be a DP since it uses an AUX channel which is not free and is being used for EFP4.

## 4.2 Low-Voltage Differential Signaling (LVDS)

LVDS is an older interface style of "raw" LCD flat panel. It tends to be used as an integrated display like a notebook display (on older models) or panel mounted displays. It is usually a fixed resolution and relies on the graphics display engine to handle all the display functions such as backlight control, image scaling, and other panel specific functions. The LVDS interface can support single and dual link modes (depending on the required resolution). A variable frequency pixel clock of either 25-112 MHz (single wide) or up to 224 MHz (dual wide) provides support for resolutions up to 1400x1050@60 Hz or 2048x1536@60 Hz respectively.

Intel® hardware stopped native support for this style of panel interface as of around 4<sup>th</sup> generation processors (using 7 series chipsets). However, there are third-party converter products that can convert DP or eDP to LVDS that are available.

Figure 25: LVDS Overview



Intel® hardware and software will support this operation but only if the converters do a proper job of emulating the eDP interface and do a good job of translating the eDP signals to LVDS. Unfortunately, not all the converter products on the market do a good conversion. We have found that most problems with the conversion process is caused

by the conversion product. When this happens, there is nothing that the driver can do to fix issues and any support of these products should be directed to the support organization of the conversion product provider.

Generally, the best solution is to use native eDP panels rather than relying on the often problematic conversion products and outdated LVDS panels.

Something to keep in mind is that only the designated LFP port(s) on designated DDI(s) on the Intel® product family you are using can be treated as an LFP. So, if you were to use a DP to LVDS converter on a DP port, you will not have the features that are provided on the native LFP port such as:

- Software Brightness control (backlight) from display control panel (i.e. IGCC or CUI or Windows\* Display Settings).
- Assertive Display Technology (ADT) which includes ADB.
- Software Contrast control.
- iGPU scaling (when the mode selected does not match the panel native resolution)
- Power on/off timing (panel specific)
- LFP pipe specific power saving features such as DPST (refer to EDS for details based on specific Intel® product family).

#### 4.2.1 Advantages – LVDS

The following are the advantages of LVDS:

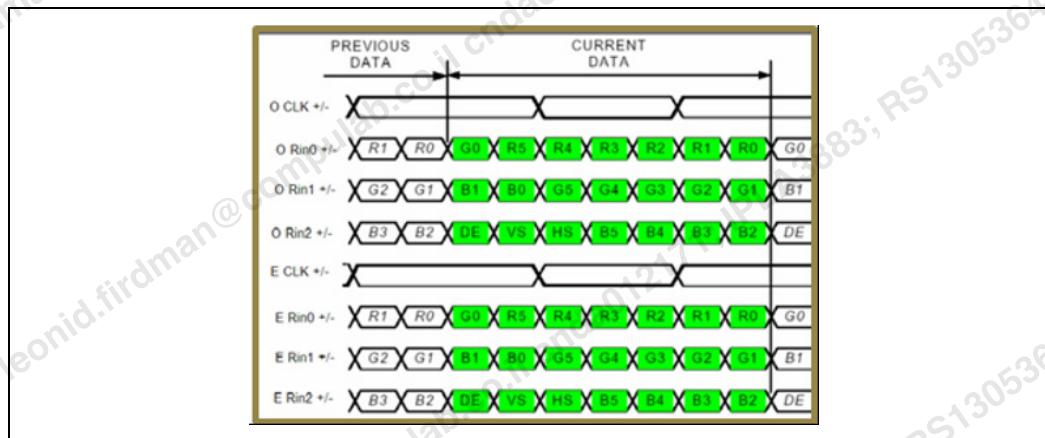
- Wide availability.
- May be slightly lower cost.

#### 4.2.2 Disadvantages – LVDS

The following are the disadvantages of LVDS:

- Replaced by eDP especially on all modern notebook computers.
- Needs third party converter as the interface is no longer supported by Intel® hardware.
- Shorter cable lengths than eDP.
- Rarely support EDID making firmware implementation more difficult.
- Different panel types (16bit, 24bit, etc.) all have different connector pinouts, so implementation can be difficult.
- Different incompatible data format make interchange difficult.
- 18bpp vs 24bpp
- OpenLDI vs SPWG

**Figure 26: SPWG Data Format Example (18bpp Dual Channel)**

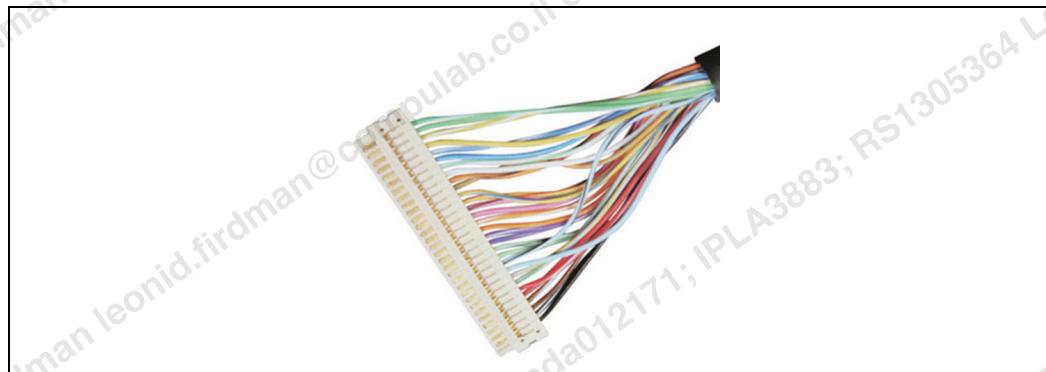


#### 4.2.3

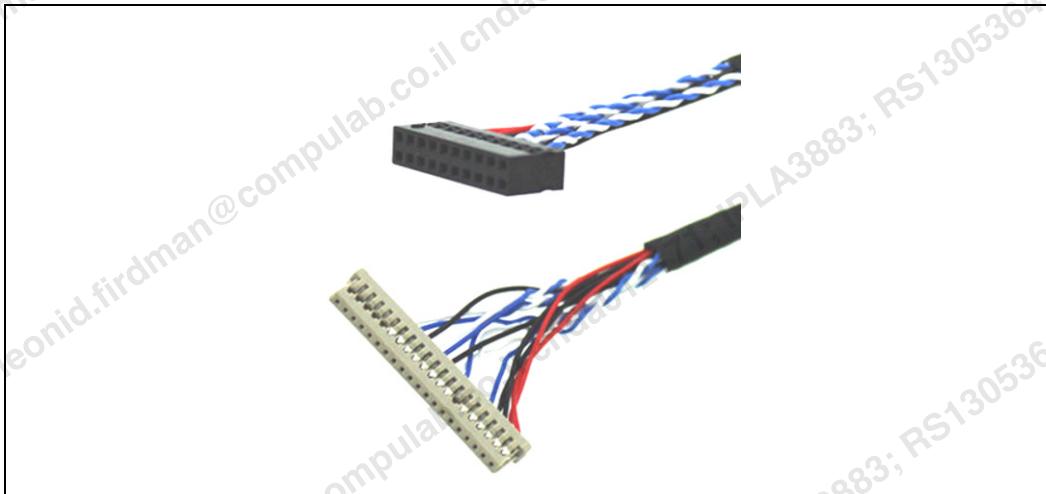
#### Cable/Connector – LVDS

LVDS panels are available with several different styles of connectors making them difficult to interface. Following is an example of one style:

**Figure 27: Example LVDS Panel-side Connector**



The connector on the motherboard also can vary. The top connector on the following LVDS cable is an example of one used on some embedded motherboards and the bottom connector is an example of another LVDS panel connector:

**Figure 28: Example LVDS Cable**

#### 4.2.4 Troubleshooting – LVDS

To troubleshoot issues:

- Intel® does not provide support for a third-party conversion solution. You should contact the vendor that provides the converter for support, and not Intel®.
- Determining the signaling and data format and timing of an LVDS panel is complex. You will need the datasheet for your panel and work with the converter vendor to make sure their product works with the panel you wish to use. The Intel® firmware and driver do not control this, the third-party converter solution needs to support this and only the third-party provider can support the operation of their solution.
- If you have connected your converter to LVDS to a non-native port (i.e. to a DP port) the panel will be treated as a standard external display (monitor) and will not have LFP features. It is best to use the LFP port for LFP devices.

### 4.3 MIPI Display Serial Interface (MIPI-DSI)

Display Serial Interface (DSI) specifies the interface between a host processor and peripherals such as a display module. DSI is a high speed and high-performance serial interface that offers efficient and low power connectivity between the processor and the display module.

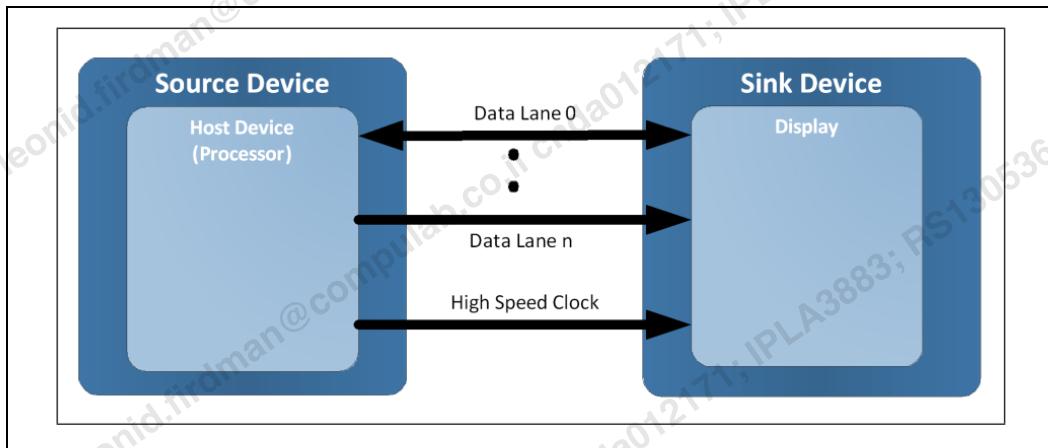
- One link x8 data lanes or two links each with x4 lanes support.
- Supports the Backlight control signal.
- Supports VESA DSC (Data Stream Compression).

One issue with MIPI-DSI is that interface is very complex to configure properly. Information is needed from the panel manufacturer and specifications sheet as well as

## Local Flat Panels (LFPs)

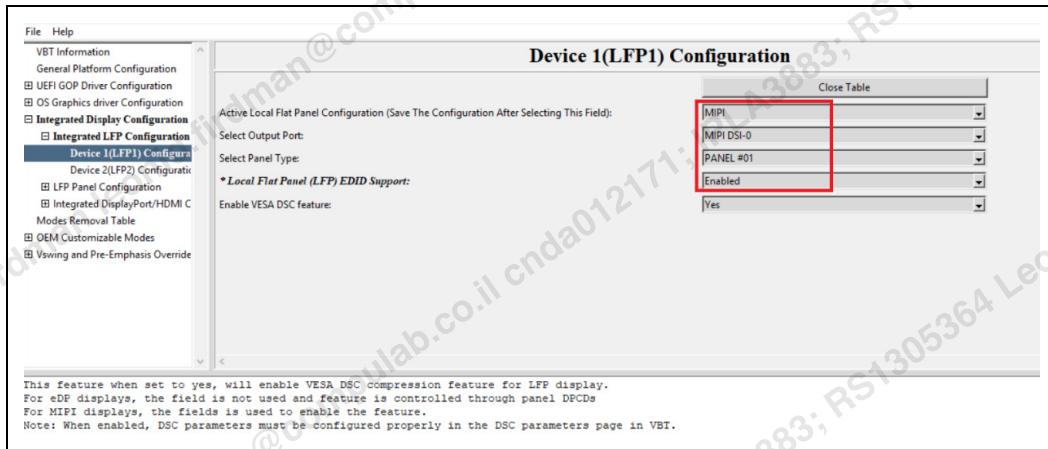
the options required to be tuned in the Intel® firmware and driver settings. For this information, refer to the DisCon Tool User Guide (available on the Intel® RDC website), any white papers on MIPI enabling (example: Doc #604505 on rdc.intel.com) as well as the product specific features shown within the specific GOP for the product you are using. Without all this information, getting a MIPI working is very difficult.

**Figure 29: MIPI-DSI Overview**



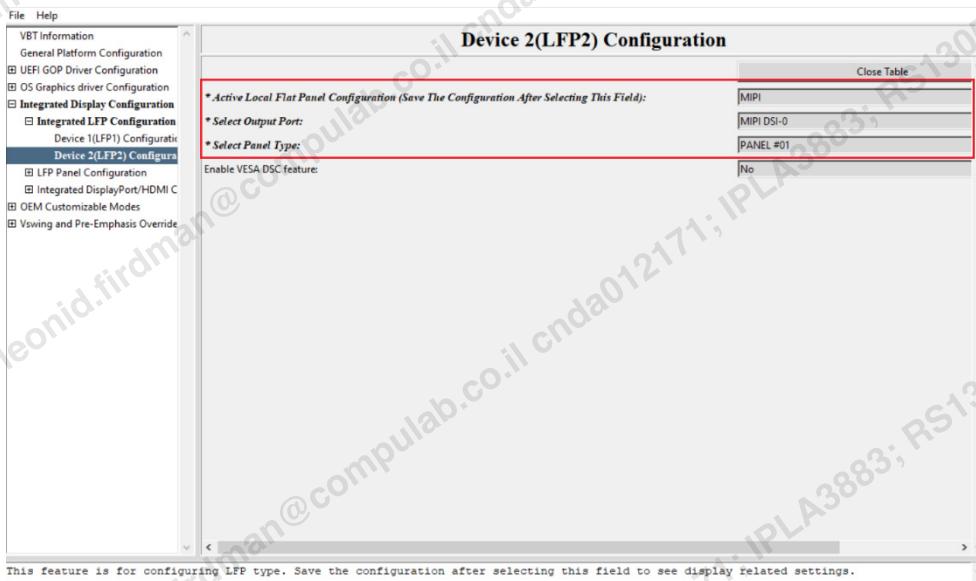
### 4.3.1 DisCon Tool Settings to Enable MIPI-DSI Panel

**Figure 30: Enabling MIPI-DSI in DisCon on Gen9 or earlier platforms**



#### Local Flat Panel (LFP) EDID Support:

- Enable this option when the panel has an EEPROM chip to provide the EDID/DTD details.

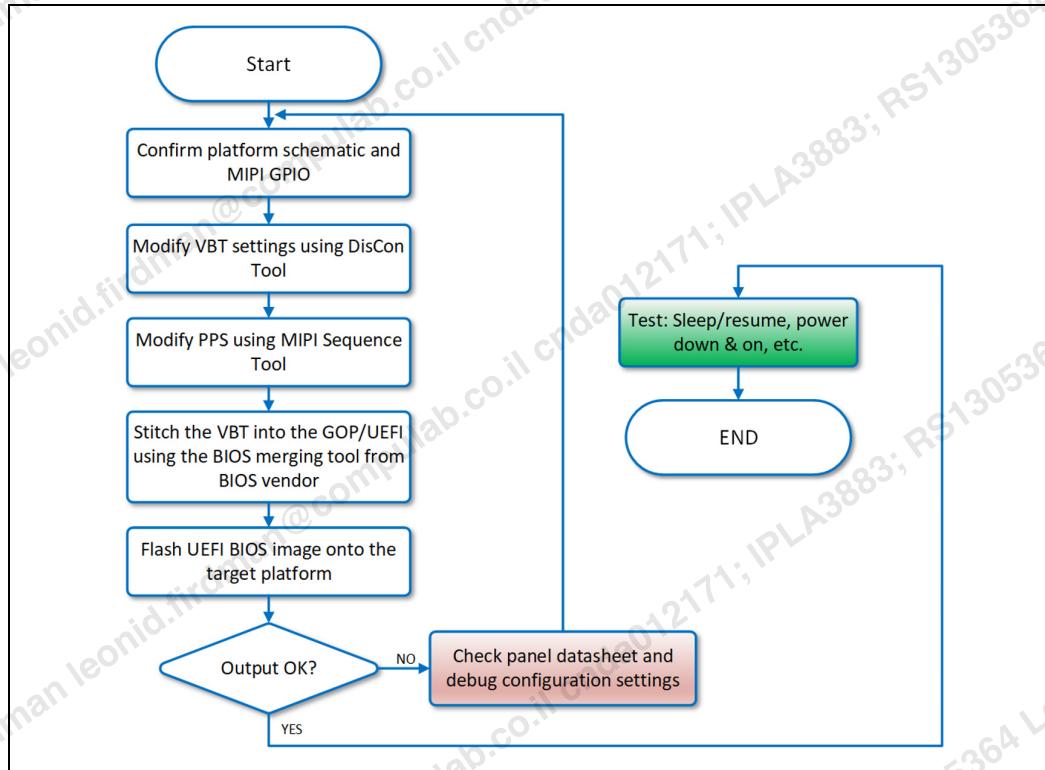
**Figure 31: Enabling MIPI-DSI in DisCon on Gen11+ platforms**

**Note:** If a display does not have an EDID capability to send identification and timing information to the driver, then it requires that DTD information to be provided. For more details, please refer to Section 6.0.

Refer to the DisCon tool user guide for more detailed instructions on the various VBT settings required to enable MIPI-DSI panel.

#### 4.3.2 MIPI-DSI Flow Diagram

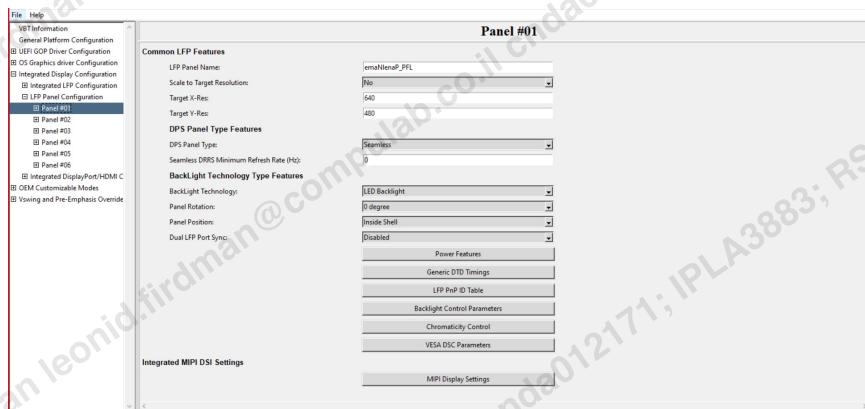
Following is the flow diagram of MIPI-DSI sequence.

**Figure 32: Flow Diagram**

#### 4.3.3

### DisCon Tool Settings to configure the selected MIPI-DSI panel

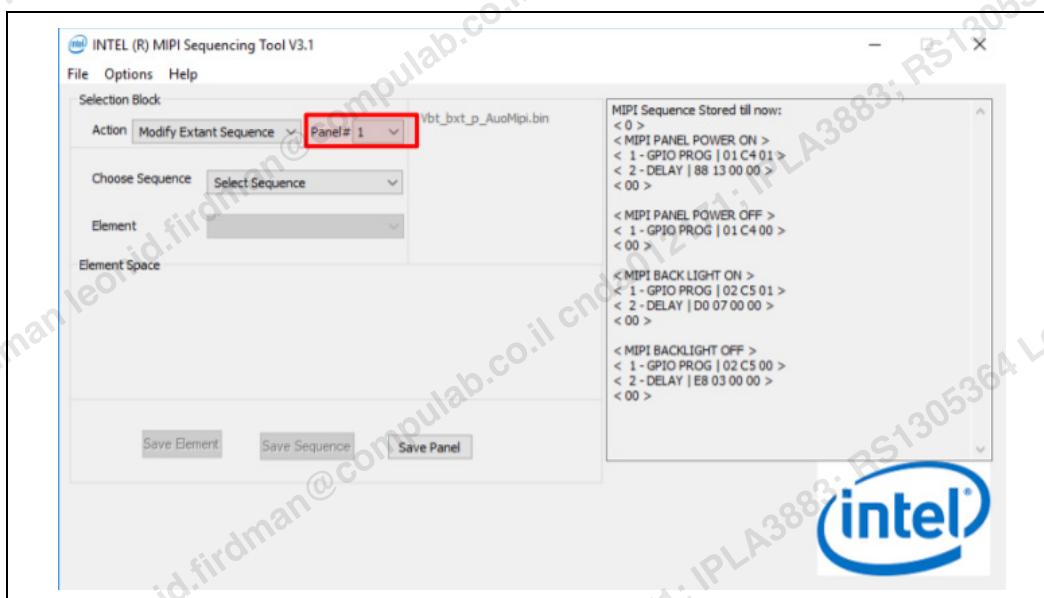
Based on the panel selected as seen in Figure 30 and Figure 31, go to that particular panel # and configure all the values as shown in Figure 33.

**Figure 33: Panel specific settings**

#### 4.3.4 MIPI Sequence Tool Setting

1. Open the VBT.bin file from the VBT GOP driver directory by using "Load .bin/.xml file" option from the MIPI Sequence tool.
2. Check the GPIO Resource ID, GPIO Resource Number, and GPIO value.
3. The following figure shows the default Power Sequence settings for MIPI panel. Users need to modify the settings according to their GPIO and PPS systems for the panel.
4. "MIPI Sequence Type" table shows the MIPI sequence type available in the MIPI Sequence Tool.

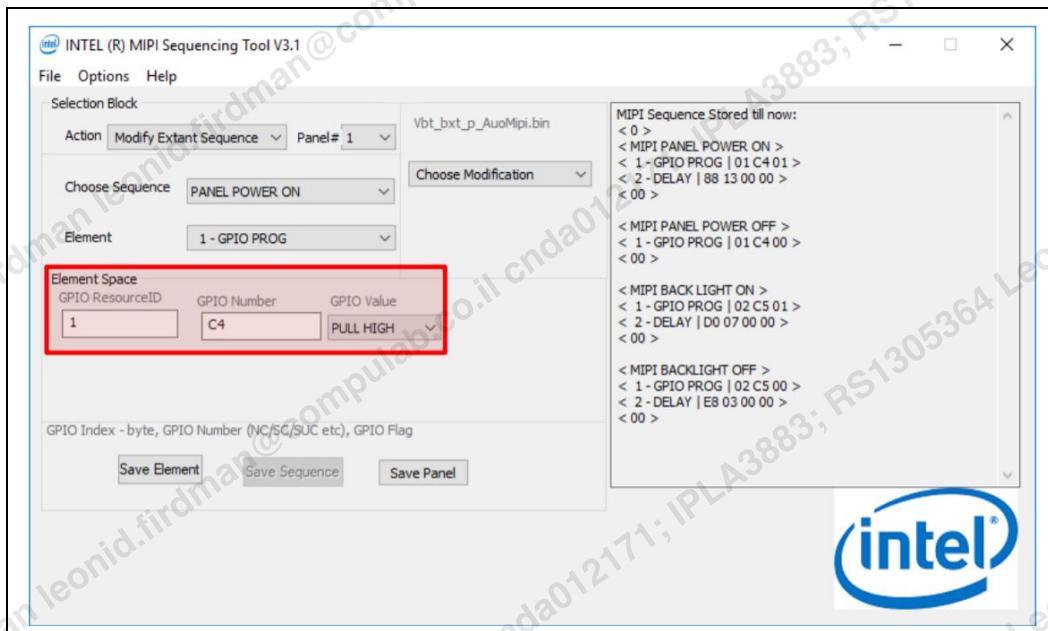
Figure 34: MIPI Sequence Tool



**Table 4: MIPI Sequence Type**

Sequence Type	Reasons for Adding
<b>Panel Power On/Off</b>	Turning On/Off voltage rails (LDOs) for eDP/MIPI. <ul style="list-style-type: none"> <li>Needed before the EDID reads initialization and during mode-set / disable sequence</li> </ul>
<b>Reset Pin Assert/Deassert</b>	Toggling the 'Reset GPIO Pin' for MIPI Only
<b>Display On/Off Commands</b>	To send data in LP/HS mode for MIPI Only. <ul style="list-style-type: none"> <li>Data can be sent in LP/HS mode.</li> <li>Mostly used to send Exit/Enter Sleep, and display On/Off commands to the panel</li> </ul>
<b>Backlight On/Off</b>	Used for MIPI and eDP to: <ul style="list-style-type: none"> <li>Program related and specific backlight panel registers</li> <li>Toggle GPIOs which gate/ungate the Backlight control pin to the panel.</li> </ul>
<b>MIPI Tear On/Off</b>	To program the specific command mode parameters for MIPI only.

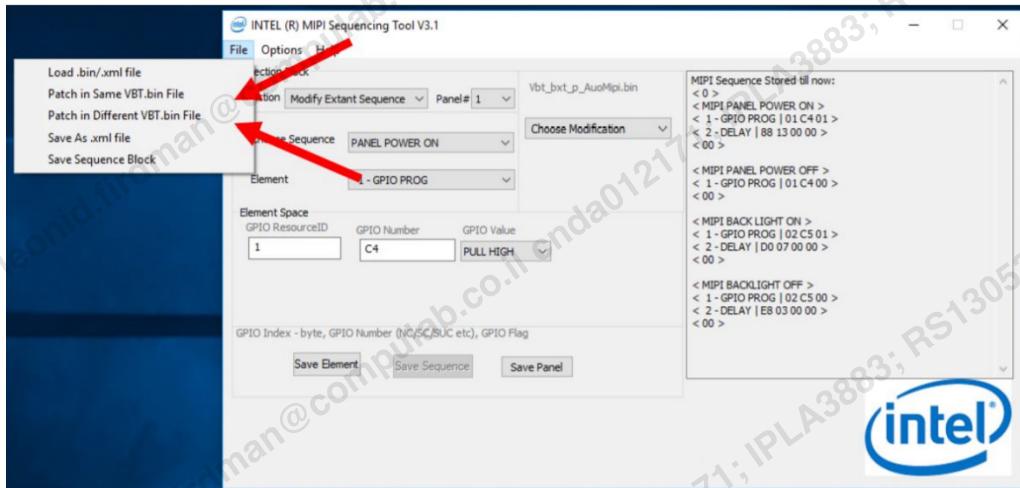
5. Work with a panel vendor to find out the different types of commands to be sent to the panel in order to bring up the panel. The whole set of commands is known as Panel Power Sequence (PPS).
6. Be extra cautious in ensuring that the Panel # chosen matches with the panel type from the DisCon tool as shown in the snapshot below.

**Figure 35: Modify the GPIO Resources ID, Number and Value**

7. Save the sequences after the desired value is modified.

8. Users can select 'Patch in Same VBT.bin File' to save it to the same file or can select 'Patch in Different VBT.bin File' to save the sequence in a different file.

**Figure 36: Saving VBT File**



Refer to MIPISeqUserGuide, also available on the Intel® RDC website, for in depth details on using the tool.

#### 4.3.5 Advantages – MIPI-DSI

Potentially lower power than other LFP panel types.

#### 4.3.6 Disadvantages – MIPI-DSI

The following are the disadvantages of MIPI-DSI:

- Complex to configure properly and get displaying.
- All MIPI panel manufacturers do not provide comprehensive or accurate documentation on the panels.
- Not widely used with computer type applications – more common on devices like printers and mobile phones.
- FPC cables are fragile.
- Short cable lengths.

#### 4.3.7 Cable/Connector – MIPI-DSI

MIPI cables can vary. The following is an example of one used with an Intel® CRB:

**Figure 37: MIPI FPC Cable**

#### 4.3.8

#### Troubleshooting – MIPI-DSI

To troubleshoot issue on MIPI-DSI:

- Most issues with MIPI are with the configuration. That requires getting good information from the panel manufacturer. Intel® is not able to provide the necessary details for third party provided panels and support should start with the panel provider. Another source for information on the options Intel® provides is the DisCon Tool User Guide (on the Intel® RDC website) and any Whitepapers from rdc.intel.com on MIPI (ex. Doc#604505).
- Intel® validates the MIPI port operation both at a hardware and driver level but cannot validate ALL MIPI panels available.
- You may be more successful using a more mainstream type panel such as eDP rather than MIPI.

§

## 5.0 External Flat Panels (EFP)

External flat panels, unlike LFPs, are complete monitors in that they tend to be:

- A separate device from the system.
- Interface via an “external” type connection such as DisplayPort\*, HDMI, DVI, etc.
- Provide their own power.
- And completely manage the display panel for power management, scaling, brightness/contrast or more.

**Figure 38: Example EFP**



### 5.1

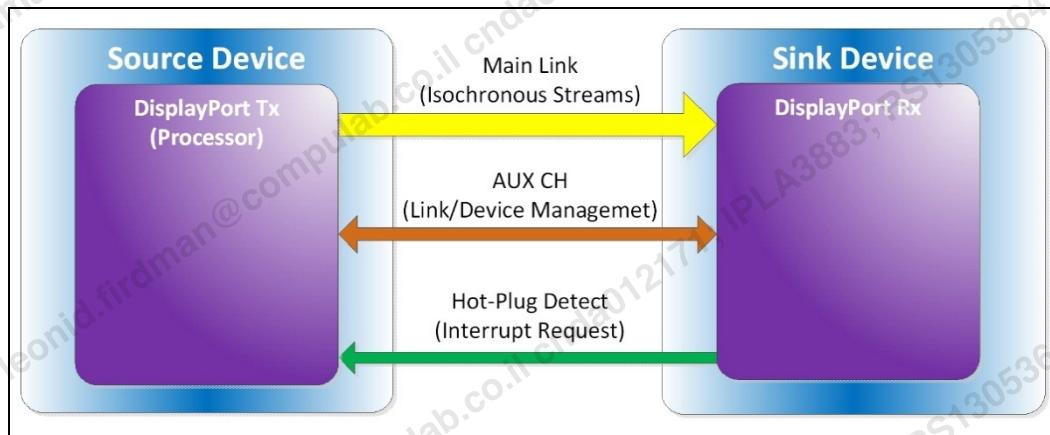
### DisplayPort (DP)

The DisplayPort\* is one type of an EFP interface. It is a digital communication interface that uses differential signaling to achieve a high-bandwidth bus interface designed to support connections between PCs and monitors, projectors, and TV displays.

A DisplayPort\* consists of a Main Link, Auxiliary channel, and a Hot-Plug Detect signal.

- The Main Link is a unidirectional, high-bandwidth, and low-latency channel used for transport of isochronous data streams such as uncompressed video and audio. It is typically comprised of 4 lanes of data.
- The Auxiliary Channel (AUX CH) is a half-duplex bidirectional channel used for link management and device control.
- The Hot-Plug Detect (HPD) signal serves as an interrupt request for the sink device.

Figure 39: DP Overview



### 5.1.1 Advantages – DP

The following are the advantages of DP:

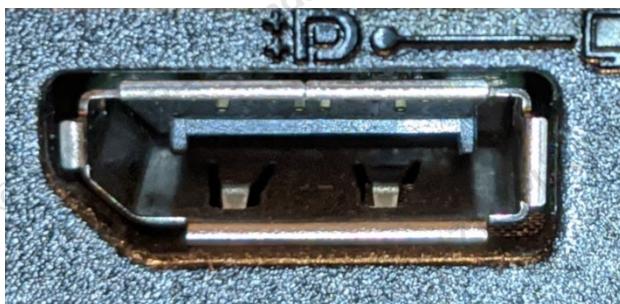
- Royalty free display interface (Logo Certification may cost).
- Is the typical display interface for computer applications (i.e., monitors).
- Easily adapted to other display interfaces such as HDMI and DVI.
- Better scalability than other display interfaces.
- Works more reliably in high interference situations found with many embedded use cases.

### 5.1.2 Disadvantages - DP

Less common for consumer video products (Blu-ray\* / DVD players, etc.) than HDMI.

### 5.1.3 Cable/Connector – DP

DP connectors can be in either standard or mini form factors.

**Figure 40: Standard DP system connector****Figure 41: Standard DP Cable Connector****Figure 42: Mini DP Cable Connector**

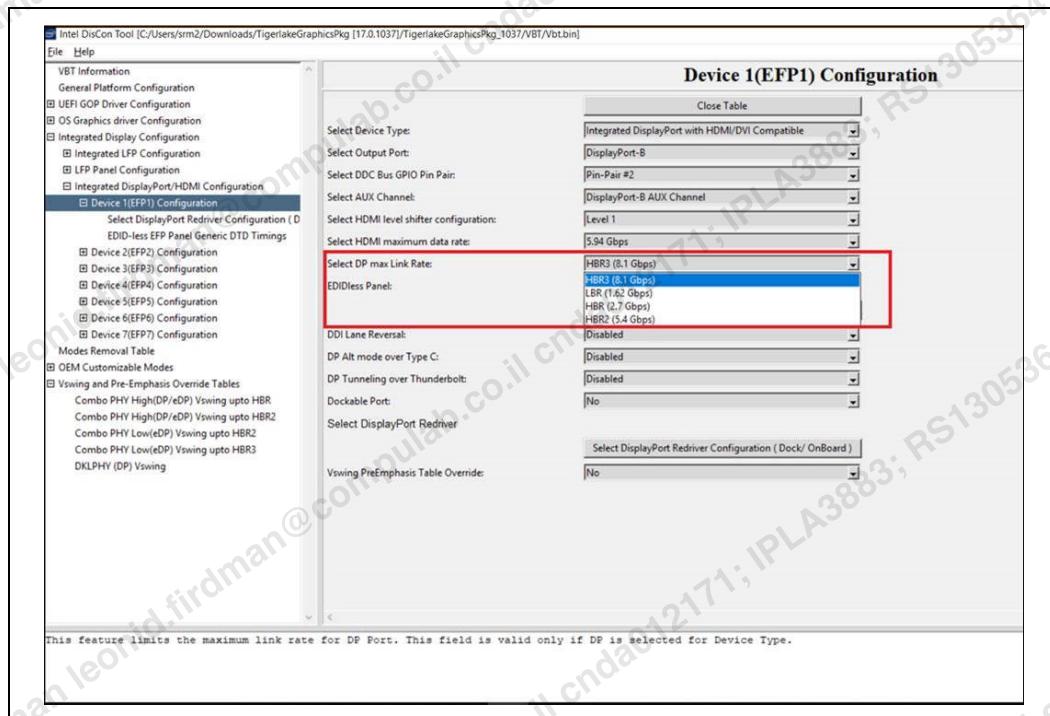
### 5.1.4 Troubleshooting – DP

To troubleshoot issues on DP:

- Some monitors incorrectly control the HPD signal during some power cycling events. The symptoms of this may include black screen, display jitter, display artifact or more, especially after a mode-set or sleep/resume cycle.
- Depending on the board design, some tuning of the DP signals may be required to help compensate for signal loss, noise, etc. Some of the symptoms are failed link training, black screen, display jitter, etc. especially at start-up. Minor correction capabilities may exist in the display engine for DP signal propagation from the processor to the DP displays. Some Intel processor generations have settings like pre-emphasis, voltage swing, DP Port Trace Length, and iBoost options in the VBT to help compensate this. If these options do not show up in the VBT via DisCon tool, then that processor does not support this capability. For example, the DP Port Trace Length and the iBoost options have been removed starting from ICL onwards. Details on the VBT options can be found either in BMP user guide or DisCon user guide on RDC.
- DP compliance issues are often caused due to usage of the wrong test suite version. For example, DisplayPort interfaces on Tiger Lake adheres to the DP1.4a specification, hence it is required to be tested with DP1.4a CTS test suite (not DP1.2 CTS).
- Quality of the cable used may impact the resolution. For example, if you connect 4K monitor to a system capable of outputting 4K but you get a lower resolution, then it is good to check the quality of the cable. Use a higher quality cable for higher resolutions and higher link rates.

The following screenshot shows the DisCon page for setting the link rate options for DP:

**Figure 43: VBT Link Rate Settings**



### 5.1.5 Maximum Resolution Support on DP

The following are the known maximum resolutions for the DisplayPort\* interface for the latest announced Intel® processor families at the time of publication of this document.

**Note:** Refer to the EDS and Spec updates for any changes to the maximum resolutions that can be supported.

**Table 5: DP Maximum Resolution by Platform**

Platform	Spec Version	Link Rate	Maximum Resolution	Maximum Resolution with DSC
Skylake	1.2	HBR2	4096x2304 @ 60 Hz 24bpp*	NA
Kaby Lake-S	1.2	HBR2	4096x2304 @ 60 Hz, 24bpp*	NA
Coffee Lake	1.2	HBR2	4096x2304 @ 60 Hz 24bpp*	NA
Whiskey Lake-U	1.2	HBR2	3840x2160 @ 60 Hz 30bpp 4096x2304 @ 60 Hz 24bpp*	NA

Platform	Spec Version	Link Rate	Maximum Resolution	Maximum Resolution with DSC
Tiger Lake	1.4a	HBR3 (Type C)	4096x2304 @ 60 Hz 36bpp 5120x3200 @ 60Hz 24bpp	5120x3200 @ 120Hz 30bpp 7680x4320 @ 60Hz 24bpp
Apollo Lake	1.2	HBR2	4096 x 2160 @ 60 Hz	NA
Elkhart Lake	1.4	HBR3	4096 x 2160 @ 60 Hz	NA

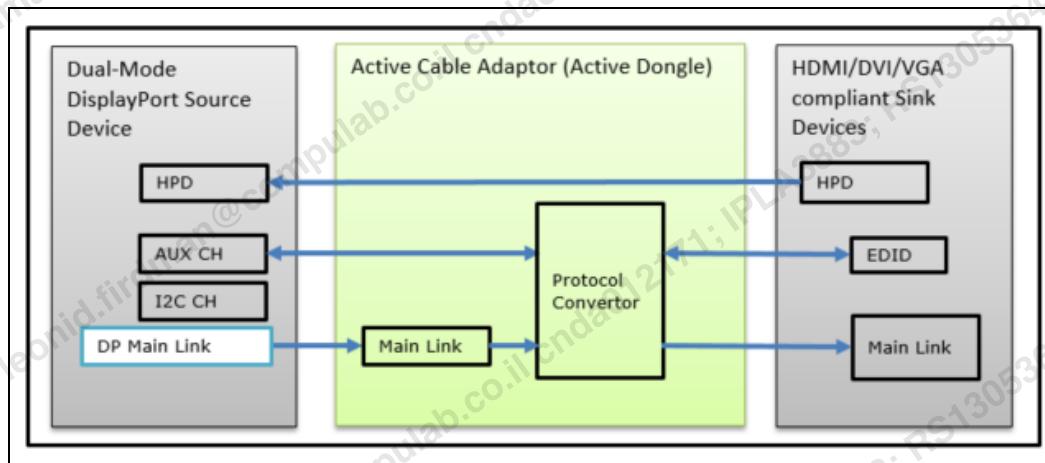
\* For Gen9 iGPU platforms, there is a special ultra-wide 5K mode that can be supported with limitations. These platforms can support 5120x1440 60 Hz, however, some graphics composition features are not supported in hardware and will need to be handled using slower DWM composition in Windows\*. 5K resolution support if not required can be disabled using the inf key "DisableSinglePipe5kModeSupport".

### 5.1.6

### Active and Passive Adapters on DP

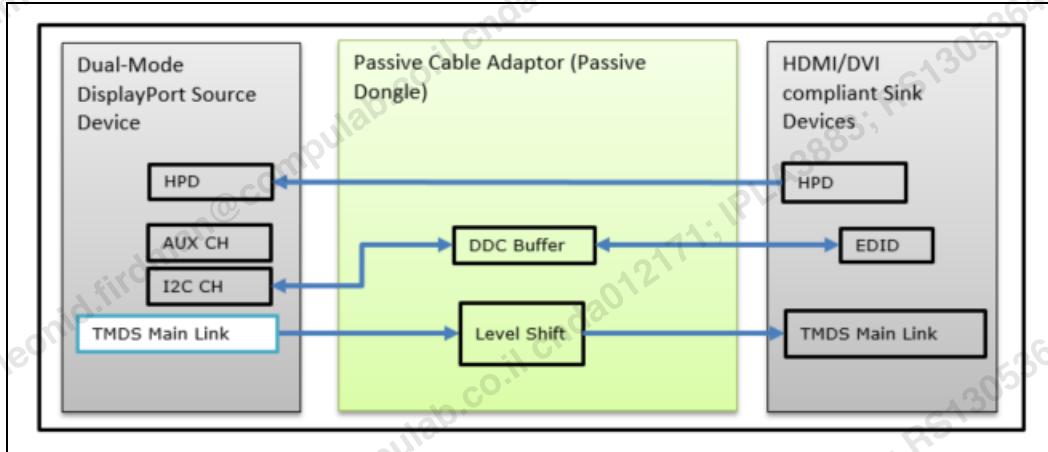
The DisplayPort connection is a very versatile interface. Included in its capabilities is the ability to be easily adapted to other display interfaces. A DP port can be converted to older HDMI, DVI or VGA display interfaces via the use of readily available adapter "dongles". DP dongles typically come in two main styles- active and passive.

Active dongles take in DP signaling and do active conversion to the target output. They work by acting like a DP display is connected, allowing the DP port and driver to operate as intended, as a DP output. On the dongle output, the display data is actively converted to the desired older display interface. They normally are a bit larger and may need to have external power to drive the conversion circuitry they use. They also may be slightly more expensive due to the extra circuitry needed to do the active conversion. Intel® recommends this type of converter as they tend to be higher quality than passive dongles and are less likely to cause issues with the conversion process. Active dongles seem to always work.

**Figure 44: Active Adapter Overview****Figure 45: Example Active Adapter**

Passive dongles rely on the DP port and driver to recognize that the dongle is attached and switch the output from DP signaling to the target interface signaling (i.e., HDMI or DVI TDMS). The native DP interface operates at 3V but HDMI and DVI need to be 5V so the passive dongle must level-shift the 3v DP port output to the target monitor 5V. The DP port must support dual-mode operation (sometimes known as DP++) which is an optional capability according to the DP specifications from VESA. These types of dongles are less expensive and readily available. However, they also tend to cause more problems due to quality issues and capabilities. We see many issues reported that are caused by low quality passive dongles that we have no way to correct with the driver. We also see a lot of confusion around passive dongle capabilities where a slower/cheaper dongle is tried at higher speeds/resolutions that they simply do not support and again cannot be fixed by the driver. Intel® IOTG does support dongles, but we do not recommend their use.

**Figure 46: Passive Adapter Overview**



**Figure 47: Example Passive Adapter**

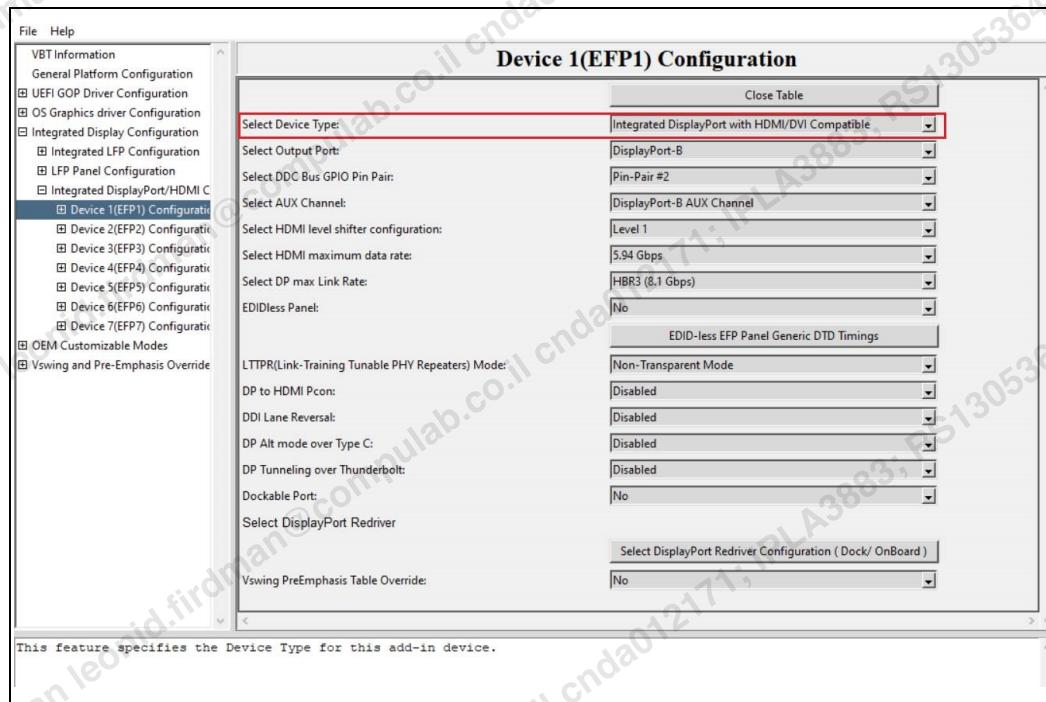


### 5.1.7

### Dual-Mode DisplayPort (DP with HDMI/DVI Compatible)

Dual-Mode DisplayPort is a standard which enables a DisplayPort to use simple passive adapter (dongle) to connect to HDMI or DVI displays. Dual-mode is an optional setting (in VBT) by the OEM so not all DisplayPorts are necessarily set to support DVI/HDMI passive adapters. Officially, the "DP++" logo is used to indicate a DP that supports dual-mode, but some system manufacturers (especially Embedded platform manufacturers) do not do the certification to get the rights to use the logo. The DP ports provided on the Intel® chipsets have dual-mode capability but, again, that option may or may not be configured (in firmware).

**Figure 48: VBT Option for Enabling Dual-mode**



Devices that implement dual-mode will detect that a DVI or HDMI passive adapter is attached, and send DVI/HDMI TMDS signals instead of DisplayPort signals. The original DisplayPort Dual-Mode standard (version 1.0), used in DisplayPort 1.1 devices, only supported TMDS clock speeds of up to 165 MHz (4.95 Gbit/s bandwidth). This is equivalent to HDMI 1.2, and supports up to 1920×1200 at 60 Hz.

With Dual-Mode 1.1 standard in 2013, which added support for up to a 300 MHz TMDS clock (9.00 Gbit/s bandwidth) and is used in newer DisplayPort 1.2 devices. This is slightly less than the 340 MHz maximum of HDMI 1.4, and supports up to 1920×1080 at 120 Hz, 2560×1440 at 60 Hz, or 3840×2160 at 30 Hz.

In 2014, the DisplayPort 1.3 standard was released, and it supports up to a 600 MHz TMDS clock (18.00 Gbit/s bandwidth), the full bandwidth of HDMI 2.0 was added. This theoretically can handle 1920×1080 at 240 Hz, 2560×1440 at 144 Hz, or 3840×2160 at 60 Hz.

With the 1.4/1.4a standard in 2018, the standard was updated to support 8K via HDMI 2.1 standards. However, it is unclear if a passive dongle/adapter exists at the time of publication of this document.

You should check the capabilities of the specific Intel® chipset you are using to verify the max resolution that is supported.

### 5.1.8 DP to VGA (CRT) Adapter

A DP to VGA adapter is an active adapter as it accepts the DP signals and actively converts them to the analog RGB signals a VGA display requires. The DP port stays in DP mode thinking it is communicating to a DP display, but in reality, it is the VGA dongle input that appears to be a DP device.

**Figure 49: Example DP to VGA Adapter**



### 5.1.9 DP Alt Mode on USB Type-C

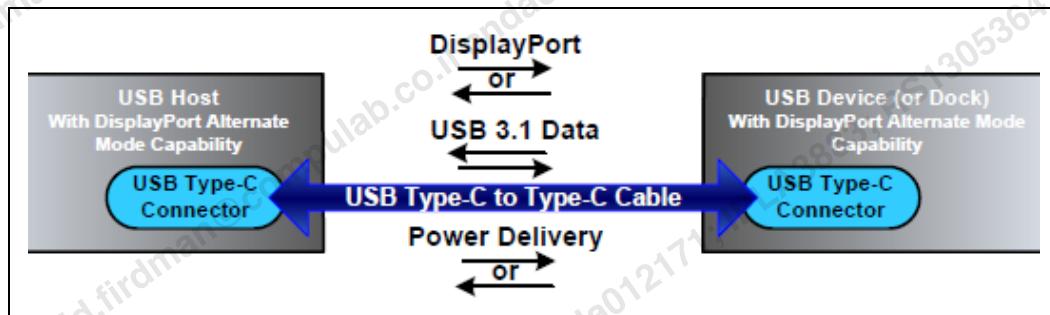
The USB Type C receptacle, plug and cable provide a smaller, thinner and more robust alternative to the existing USB 3.1 interconnect.

It allows for more functionality than standard USB, including more power, and, more important, the ability to support DP display output on the connector. Check your specific Intel® processor capabilities to check if this is supported by the product you are using.

**Figure 50: Type-C System Connector (Female)**



The VESA DisplayPort Alt Mode Standard, Version 1.0a, was released on Aug 10, 2015. It enables the use of the USB-C interface for DisplayPort. Thereby, a USB Type C port can become capable of transferring Display data (DP Alt mode over USB-C), USB data, and power at the same time. DP Alternate Mode is a functional extension of the USB-C interface.

**Figure 51: Type-C Connector Overview**

### 5.1.9.1 **USB Type-C signals**

The above figure illustrates the functional signal plan for the USB Type-C receptacle, not all signals shown are required in all platforms or devices. As shown, the receptacle signal list functionally delivers both USB 2.0 (D+ and D-) and USB 3.1 (TX and RX pairs) data buses, USB power (VBUS) and ground (GND), Configuration Channel signals (CC1 and CC2), and two Sideband Use (SBU) signal pins.

Multiple sets of USB data bus signal locations in this layout facilitate being able to functionally map the USB signals independent of plug orientation in the receptacle. For reference, the signal pins are labeled (in the following figure).

**Figure 52: USB Type-C Signals**

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	TX1+	TX1-	VBUS	CC1	D+	D-	SBU1	VBUS	RX2-	RX2+	GND
GND	RX1+	RX1-	VBUS	SBU2	D-	D+	CC2	VBUS	TX2-	TX2+	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

Only one CC pin is connected through the cable to establish signal orientation and the other CC pin is repurposed as VCONN for powering electronics in the USB Type C plug. Also, only one set of USB 2.0 D+/D- wires are implemented in a USB Type C cable.

### 5.1.9.2 **DisCon VBT Settings**

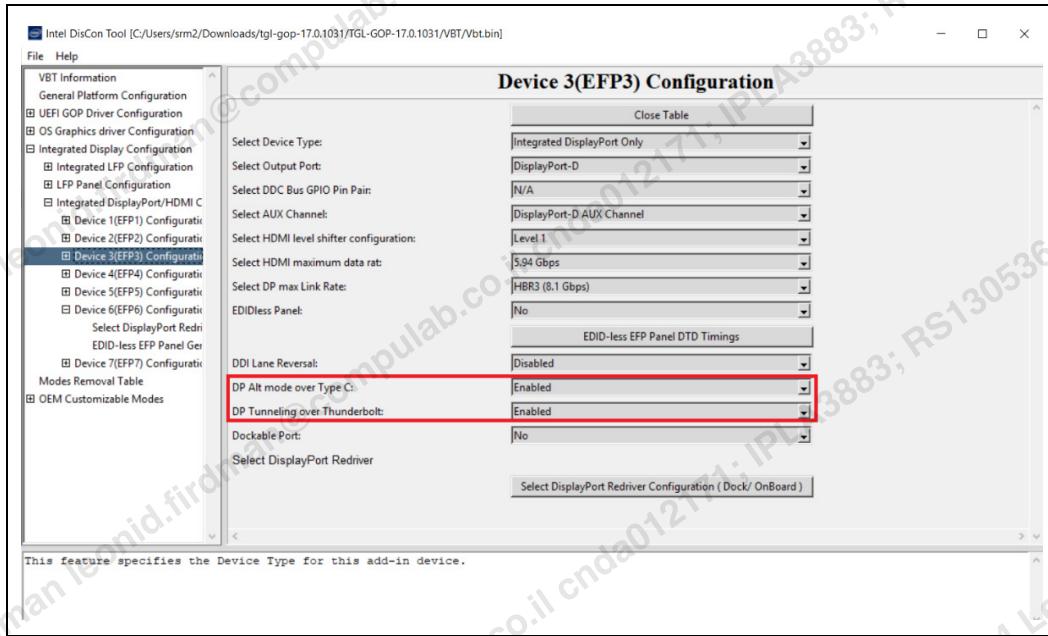
In general, once you design your hardware appropriately, you will need to configure your pre-boot firmware (GOP) and driver by configuring the VBT that is stored in your BIOS image using the DisCon tool.

You will need to know how your board designer connected the Type-C ports on your hardware in order to be able to properly configure the VBT.

As shown, select the appropriate Output Port (e.g. DisplayPort-D) and then enable the "DP Alt mode over Type C". If your board implements the Type-C port as a

Thunderbolt® port, you will also need to enable the "DP Tunneling over Thunderbolt" option.

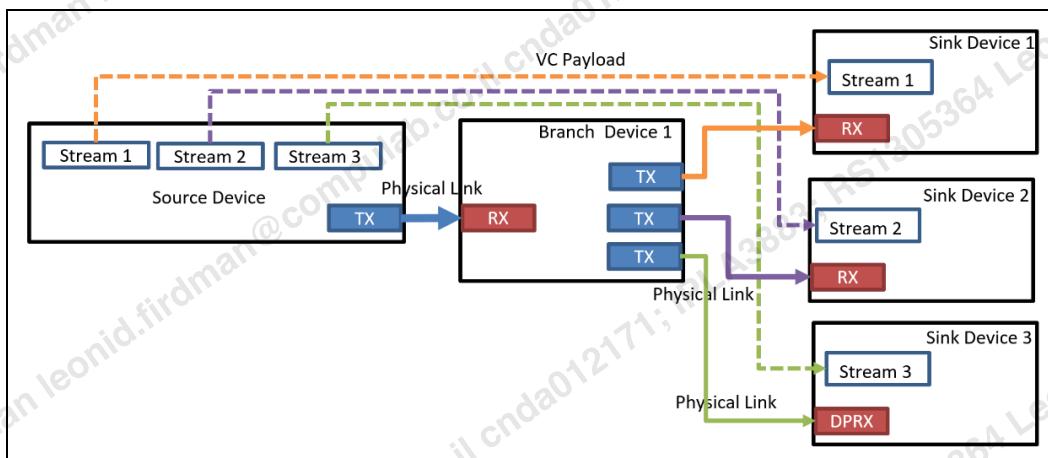
**Figure 53: VBT Settings for Enabling DP over Type-C**

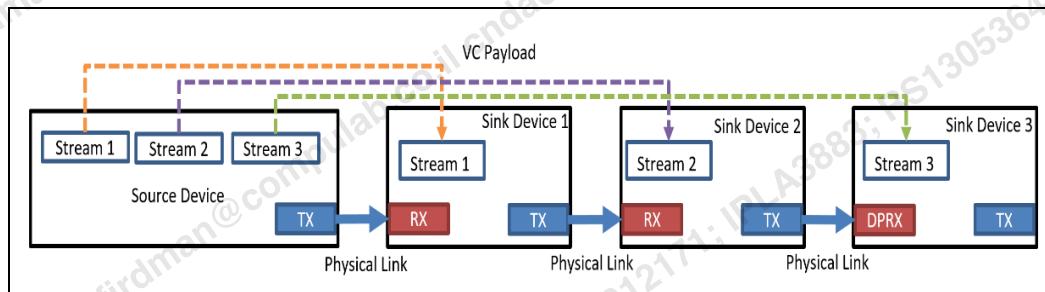


## 5.1.10 DP MST

Display Port Multi Stream Transport (DP MST) allows you to connect multiple monitors to a single output port using an MST hub or by daisy-chaining MST capable monitors. The following figures show the data flows for the two configurations.

**Figure 54: MST Hub**



**Figure 55: MST Daisy-Chaining**

The key to successfully configuring MST is to make sure you do not exceed the maximum capability of the display port.

As display technologies improve, maximum display resolutions have risen. Where a 1920x1080p display may have been common a few years back, the norm has improved to 4K displays (i.e. 3840x2160) and is moving to 8K. As the displays have improved, so has the capabilities of the Intel® integrated graphics progressed. This has included faster port link speeds, as well as increased memory bandwidth required to supply the data for the higher resolutions. Earlier products may have had a lower maximum resolution and therefore will not support the newer display technologies. The key to understanding the maximum supported resolution will be a combination of the maximum link speed as well as the maximum supported resolution of the graphics engine. These specs cannot be exceeded, or you may experience unintended operation. Refer to your specific product specs and design guides.

Maximum resolution supported for various display configurations is calculated as shown in the following example. Monitor Refresh rate is assumed to be 60 Hz and port interface is DP.

Refer to platform EDS to get:

- "Display Resolutions and Link Bandwidth for Multi-Stream Transport calculations" table for pixel clock and link bandwidth.
- "Maximum Display Resolutions" table.

**Table 6: Example MST Calculations**

Number of Displays on the System	Monitor Layout (COLS x ROWS)	Orientation per Port (COLS x ROWS)	Monitor Resolution	Resolution/Port	Total Resolution	Number of Splitters Used (One Splitter per Port)	Displays/Splitter	Link bw/ port [Gbps]	Pixel Clock/Port [MHz]	Supported
6	6x1	2x1	3840x2160	7680x2160	23040x2160	3	2	16*2=32	533.25*2=1066.5	No, Exceeds port maximum link bandwidth, Pixel clock and X resolution.
					7680x6480					No, Exceeds port maximum link bandwidth, Pixel clock and X and Y resolution.
6	6x1	2x1	1920x1080	3840x1080	11520x1080	3	2	4.46*2 = 8.92	148.5*2= 297	Yes, within link bandwidth and maximum port resolution
					3840x3240					

**NOTES:**

1. Total link bandwidth < 21.60 Gbps
2. Total Pixel clock < 720 MHz
3. Resolution per port < 4096x2304
4. HBR: 2.7 Gbps; HBR2: 5.4 Gbps; HBR3: 8.1 Gbps

Figure 56: Horizontal 6x1 MST Configuration

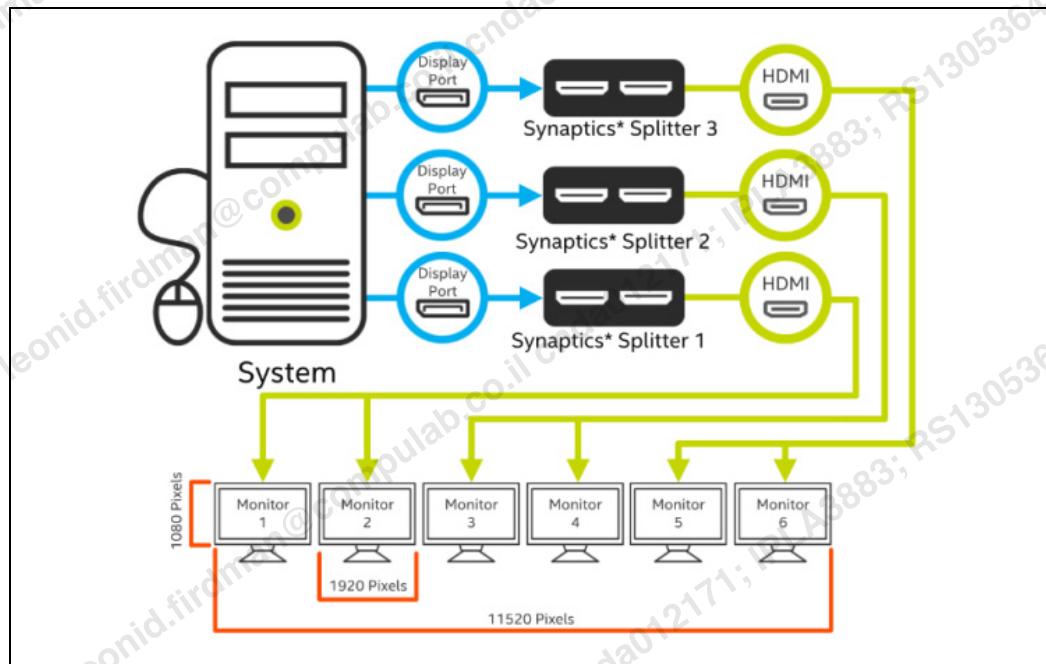
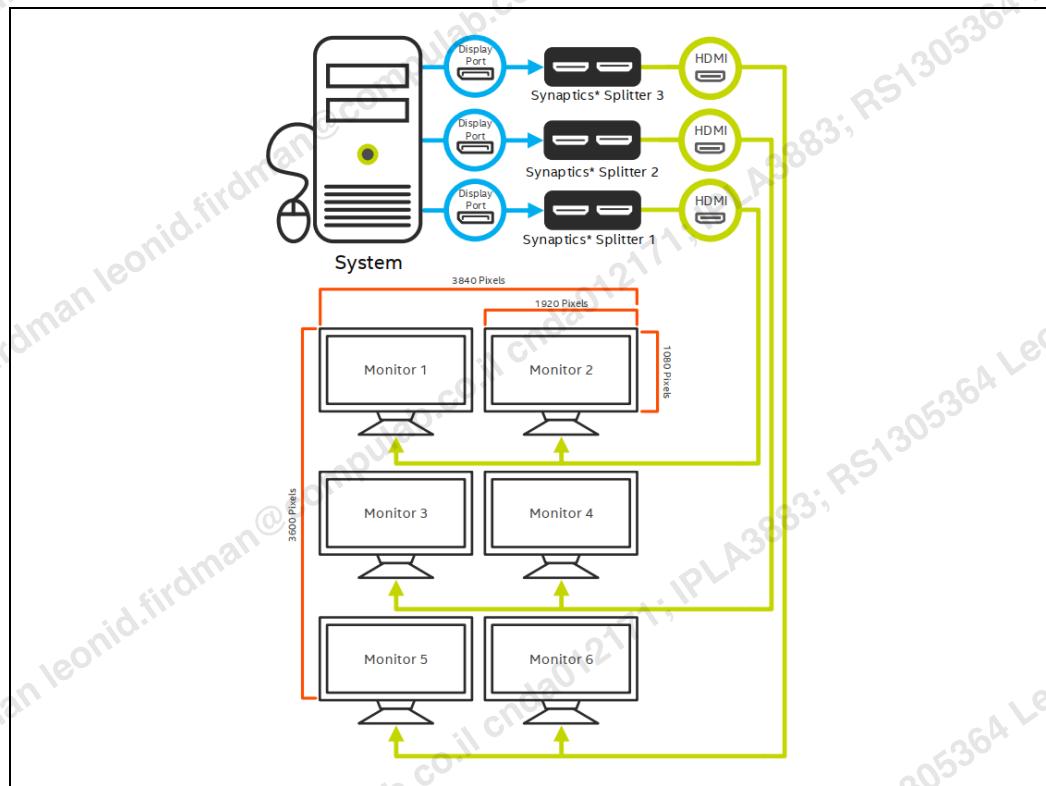


Figure 57: Vertical 2x3 MST Configuration



### 5.1.11 DP-IN

DP-IN interface is a feature required to support platforms with external graphics (dGPU) that want to access the integrated Type-C ports. Each stream transmitted from the dGPU to the DP-IN receiver interface can be internally routed to each of the USB-C subsystem ports when a Type-C solution has been implemented:

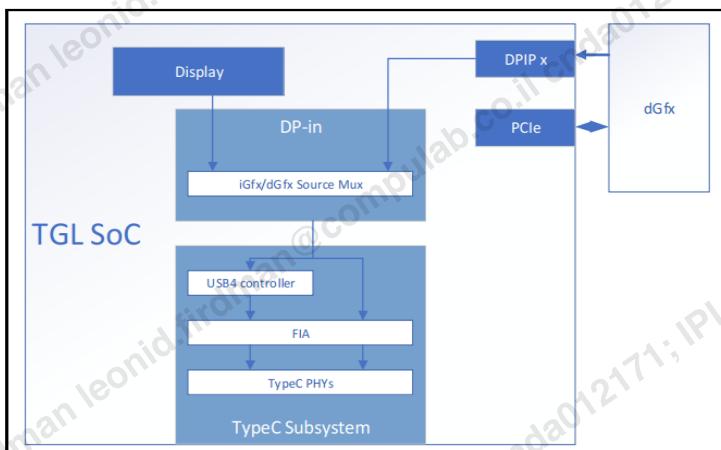
- DPoC port - DisplayPort Over Type-C.
- USB4 port - DisplayPort tunneled over USB4.

The DP-IN interface supports VESA\* LTTPR (Link Training Tunable PHY Repeater).

**Note:**

1. DP-IN is not supported on all processors. For example, on Tiger Lake, it is supported only on H processor lines.
2. Use of DP-IN requires an external display source that supports VESA\* LTTPR (Link Training Tunable PHY Repeater) non-transparent mode.

**Figure 58: Example of DP-IN on Tiger Lake-H**



Each DP-IN port supports:

- Hot Plug Detect
- AUX channel
- Main Link supporting up to 4 lanes each with up to HBR3 link rate.
- VESA\* Link Training Tunable PHY Repeater
- Only on Type-C (DPoC) or DisplayPort\* tunneling via Thunderbolt on USB-C port.

**Note:** Not supported for Fixed DP, HDMI, or eDP interfaces.

For platform specific information, please refer to DP-IN enabling user guide on Intel® RDC. For example, document #634165 is for enabling DP-IN on Tiger Lake.

## 5.2

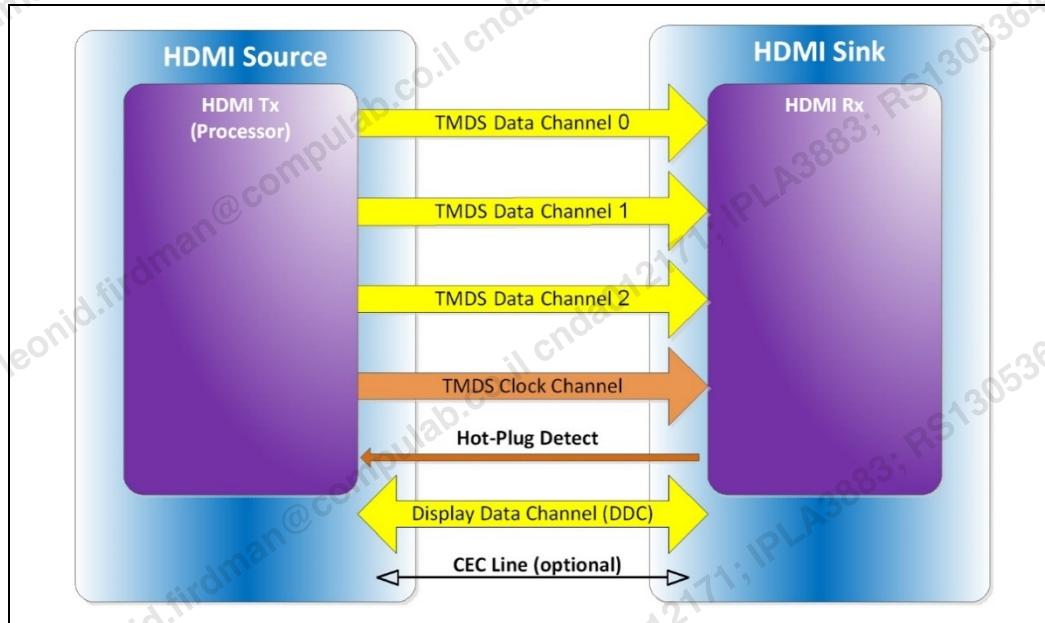
## High Definition Multimedia Interface (HDMI)

The High-Definition Multimedia Interface (HDMI\*) is provided for transmitting uncompressed digital audio and video signals from DVD players, set-top boxes, and other audio-visual sources to television sets, projectors, and other video displays. It can carry high-quality multi-channel audio data and all standard and high-definition consumer electronics video formats. The HDMI\* display interface connecting the processor and display devices uses transition minimized differential signaling (TMDS) to carry audiovisual information through the same HDMI\* cable.

HDMI\* includes three separate communications channels: TMDS, DDC, and the optional CEC (consumer electronics control). CEC is not supported on the processor. As shown in the following figure, the HDMI\* cable carries four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In addition, HDMI\* carries a VESA DDC. The DDC is used by an HDMI\* Source to determine the capabilities and characteristics of the Sink.

Audio, video, and auxiliary (control/status) data is transmitted across the three TMDS data channels. The video pixel clock is transmitted on the TMDS clock channel and is used by the receiver for data recovery on the three data channels. The digital display data signals driven natively through the PCH are 3v AC coupled and need level shifting to convert the AC coupled signals to the 5V HDMI\* compliant digital signals. The processor HDMI\* interface is designed in accordance with the High-Definition Multimedia Interface.

**Figure 59: HDMI Overview**



### 5.2.1 Advantages – HDMI

The following are the advantages of using HDMI:

- Very common on TVs and other consumer devices.
- Supports high resolution displays.

### 5.2.2 Disadvantages – HDMI

The following are the disadvantages of using HDMI:

- There is a royalty associated for using HDMI interface with which system manufacturers will have to account.
- Tends to lag behind DP for adoption of higher resolution capability.
- Many HDMI cables are low quality and cannot support capabilities or resolutions that the system and monitor can support.

### 5.2.3 Cable/Connector – HDMI

HDMI connectors come in standard, mini or micro form factors.

**Figure 60: Standard HDMI System Connector****Figure 61: Standard HDMI Cable Connector****Figure 62: Mini HDMI Cable Connector****Figure 63: Micro HDMI Cable Connector****Figure 64: HDMI Cables/Connectors Size Comparison**

## 5.2.4 Troubleshooting – HDMI

To troubleshoot issues while using HDMI:

- Blank screen, image corruption and/or display running at a lower resolution than it supports is likely due to poor quality cable. Try a higher quality certified HDMI cable. This may also be a system design issue where the HDMI port layout does not conform to Intel® design guidelines.
- Check the HDMI spec level of the specific Intel® processor family you are using before trying higher resolution (i.e. 4K or higher) to make sure the resolution is supported.
- Using a LSPCON\* interface may allow you to implement a higher HDMI spec level output. These are third party components and usually you will need to work with the providers support organization if you are having display issues with a LSPCON\* enabled design.

## 5.2.5 Max Resolution Support on HDMI Interface

The following are the known maximum resolutions for the HDMI interface for the latest announced Intel® processor families at the time of publication of this document.

**NOTES:**

1. Refer to the EDS for the latest platform specific maximum resolution and which LSPCON\* version that can be supported.
2. Refer to the EDS and Spec Updates for any changes to the maximum resolutions and which LSPCON\* that can be supported.

**Table 7: HDMI Maximum Resolution by Platform**

Platform	Spec Version	Maximum Resolution
Skylake-U Skylake – H/S	1.4 (native)	4096x2160 @ 24 Hz, 24bpp
	2.0 (via LS-PCon)	4096x2160 @ 60 Hz, 24bpp*
Kaby Lake-S	1.4 (native)	4096x2160 @ 30 Hz, 24bpp
	2.0/2.0a (via LS-PCon)	4096x2160 @ 60 Hz, 24bpp*
Coffee Lake – S/H	1.4 (native)	4096x2160 @ 30 Hz, 24bpp
	2.0/2.0a (via LS-PCon)	4096x2160 @ 60 Hz, 24bpp*
Whiskey Lake-U	1.4 (native)	4096x2160 @ 24 Hz, 24bpp
	2.0/2.0a (via LS-PCon)	4096x2160 @ 60 Hz, 24bpp*
Tiger Lake	1.4 (native)	4096x2160 @ 30 Hz, 24bpp
	2.0b (native)	4096x2160 @ 60 Hz, 24bpp
Apollo Lake	HDMI 1.4 (native)	3840 x 2160 @ 30 Hz

Platform	Spec Version	Maximum Resolution
Elkhart Lake	HDMI 2.0b (native)	4096 x 2160 @ 60 Hz

\* For Gen9 iGPU platforms, there is a special ultra-wide 5K mode that can be supported with limitations. These platforms can support 5120x1440 60Hz, however, some graphics composition features are not supported in hardware and will need to be handled using slower DWM composition in Windows\*. 5K resolution support if not required can be disabled using the inf key "DisableSinglePipe5kModeSupport".

## 5.2.6

### Level Shifter and Protocol Converter (LSPCON\*)

A DP-HDMI LSPCON\* is a device that can operate both as a DP-to-HDMI Protocol Converter Branch Device (PCON) and as a DisplayPort Dual-mode Type 2 Adaptor (level shifter or LS).

A LSPCON\* is used when you want to support a higher-level HDMI specification (I.e. HDMI 2.0, 2.1, etc.) than the Intel® product family supports natively. For example, if the Intel® product family supports HDMI 1.4 but you need HDMI 2.0, you would implement a LSPCON\*. Refer to the Intel® product family EDS for details on the native HDMI level supported on the product you are using.

The DP-HDMI 2.0 LSPCON\* needs a 27 MHz clock that can be provided using an external crystal oscillator. The DP-HDMI 2.0 LSPCON\* also needs an external SPI Flash chip to store firmware/instructions.

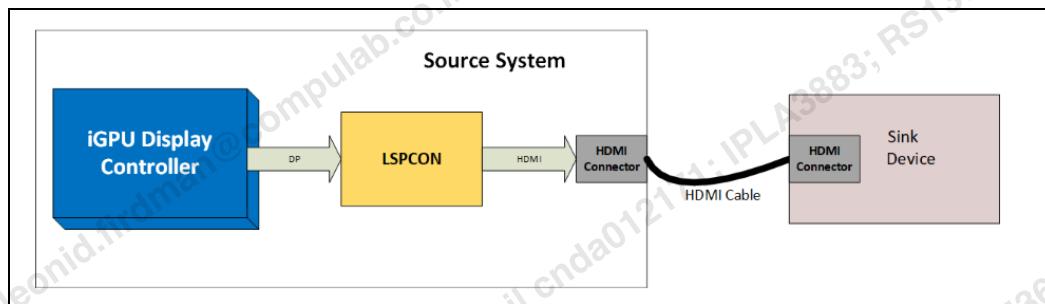
The Type 2 Adaptor function can provide a low power means of allowing an AC coupled TMDS signal from a DP Dual mode display controller to drive a HDMI\* Source receptacle.

The DP Branch Device with DP-to-HDMI protocol converter function allows a DP display controller to drive an HDMI 2.0 Source receptacle.

The LSPCON\* may be placed on the same PCB as the display controller. This spec provides information specific to a motherboard down configuration as shown in the following figure. The GPU can switch LSPCON\* operating modes using sideband signaling.

Diagram showing DP-HDMI LSPCON\* Down on the Motherboard:

**Figure 65: LSPCON\***



#### DP\* capabilities:

The LSPCON\* supports DP HBR2 and DP Dual-mode (AC coupled TMDS signaling) on its upstream-facing main-link input port. "4k" modes defined in HDMI 2.0 require the bandwidth of DP HBR2 for Protocol Converter mode.

The LSPCON\* must accept DP SST stream. The upstream-facing port accepts Manchester-mode AUX signaling, which can be used for both native AUX transactions and for I2C-over-AUX protocol. The upstream-facing port also accepts native I2C\* signaling, consistent with DP Dual-mode Type 2 adaptor requirements.

#### HDMI\* capabilities:

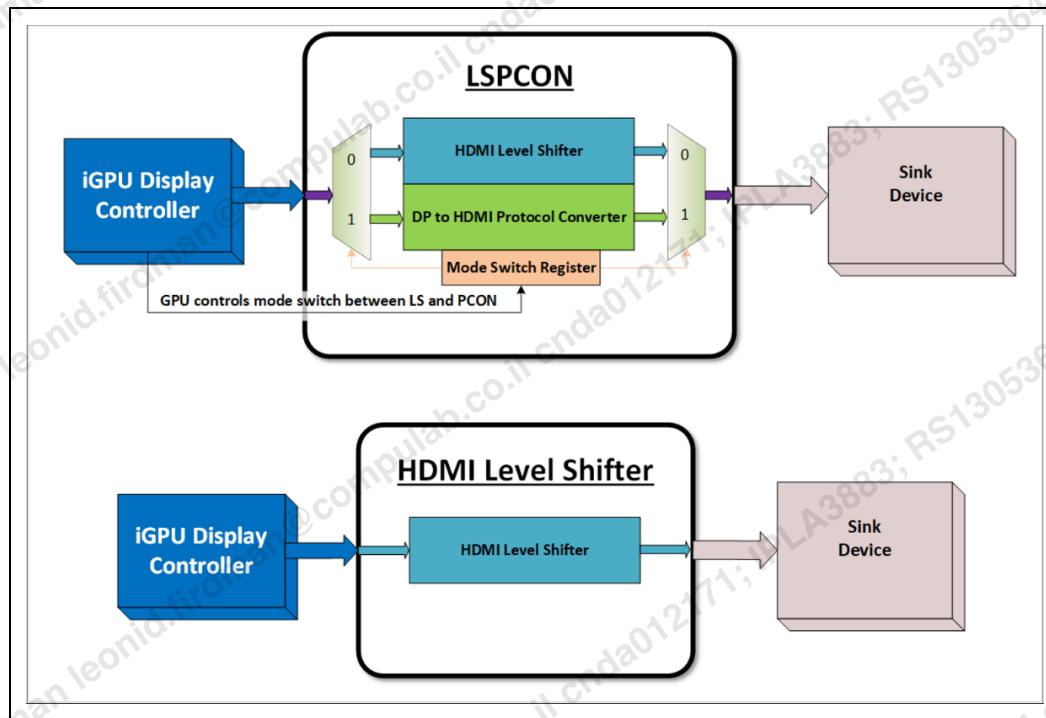
The LSPCON's\* downstream-facing HDMI Tx port allows the HDMI 2.0 Source to meet HDMI 1.4b and HDMI 2.0 compliance requirements at the HDMI Source's receptacle. The LSPCON's\* upstream-facing main-link port accepts AC coupled HDMI 1.4b streams up to 340MHz when in LS mode.

When in PCON mode, the LSPCON\* accepts a DP HBR2 stream on the upstream facing main link – and does protocol conversion from DP to HDMI 2.0 to support HDMI 2.0 streams from 340MHz to 594MHz on the downstream HDMI Tx port. The upstream-facing sideband interface accepts both native I2C signaling and I2Cover-AUX signaling, as described in the DP Dual-mode standard for Type 2 adapters.

The LSPCON\* device contains two functions:

1. A DisplayPort Dual-mode Type 2 Adaptor Function that supports HDMI 1.4b up to 3.4 Gbps.
2. A DP Branch Device with a DP-to-HDMI 2.0 Protocol Converter Function that supports HDMI 2.0 up to 5.94 Gbps through protocol conversion from DP (HBR2) to HDMI 2.0.

**Figure 66: Difference in Device Organization between DP-HDMI and HDMI LS**



**Note:** The LSPCON\* devices are provided by a third-party provider and therefore most support for these devices should be directed to the provider.

## 5.3

### Digital Visual Interface (DVI)

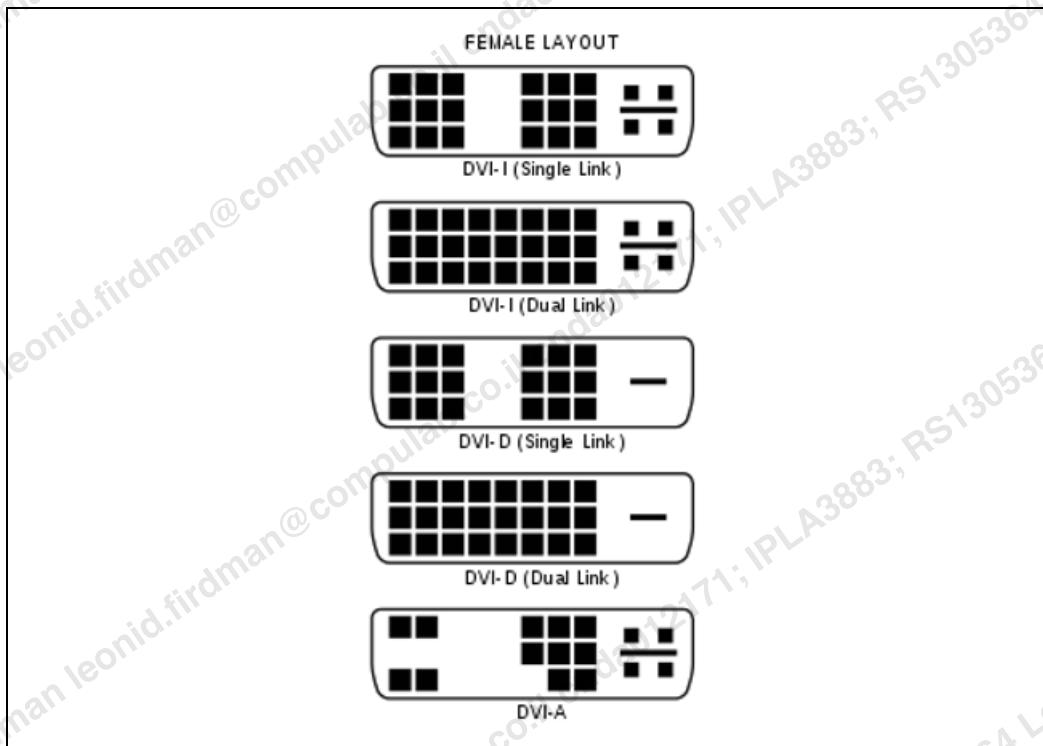
The processor Digital Ports can be configured to drive DVI-D. DVI uses TMDS for transmitting data from the transmitter to the receiver, which is similar to the HDMI\* protocol except for the lack of audio and CEC. The digital display data signals driven natively through the processor are AC coupled and need level shifting to convert the AC coupled signals to the HDMI\* compliant digital signals.

#### 5.3.1

##### DVI-A / DVI-D / DVI-I

DVI connectors are available in a few variations.

- DVI-I (integrated, combines digital and analog in the same connector; digital can be single or dual link).
- DVI-D (digital only, single link or dual link).
- DVI-A (analog only).

**Figure 67: DVI Variations**

### 5.3.2 Advantages – DVI

The following are the advantages of using DVI:

- First digital display interface.
- Supports EDID as standard.
- Has ability to support an optional VGA output as an alternative connection on the connector.
- Easy conversion to HDMI (uses the same signals) via a simple connector only converter.

### 5.3.3 Disadvantages -DVI

The following are the disadvantages of using DVI:

- No audio capability.
- Large connector not well suited for small form factor or mobile designs.

### 5.3.4 Cable/Connector – DVI

The following picture shows standard DVI cable/connector.

**Figure 68: DVI System Connector**



**Figure 69: DVI Cable Connector**



### 5.3.5 Troubleshooting - DVI

DVI is quite a stable interface with very few problems. It just tends to work. The biggest issues that happen with DVI is when one attempts to use simultaneously the VGA capability and DVI outputs with a splitter cable.

**Figure 70: DVI to DVI and VGA Splitter**



The issue is that the DVI connector only has one DDC signal to get an EDID so there is no reliable way to read an EDID for the two monitors. The DVI monitor can get incorrectly be identified (via EDID read) as being attached to both the VGA and DVI connector. Intel<sup>®</sup> recommends against attempting to use the VGA connection and DVI simultaneously.

## 5.4 Video Graphics Array (VGA)

The legacy analog interface of Video Graphics Array (VGA) was introduced by IBM\* in 1987 and is typically used in driving CRTs, flat panels, HD-TVs and projectors. The standard VGA connector is a 15-pin d-sub mini, composed of red, green, blue, horizontal, and vertical sync video signals, a VESA Display Data Channel (DDC) for communication of display configuration data and a clock signal. There are no defined standards related to image quality or maximum resolution supported by the interface, but Intel® typically provides a Digital to Analog Convertor (DAC) that will increase the frequency up to 340 MHz depending on the desired resolution.

### 5.4.1 Advantages – VGA

The following are the advantages of using VGA:

- No licensing associated for utilizing this interface.
- Was very common PC display interface.

### 5.4.2 Disadvantages – VGA

The following are the disadvantages of using VGA:

- Fast becoming obsolete.
- Analog signals provide inconsistent color rendition and low color depth.
- No native interface support in current Intel® products.
- No content protection.
- No audio support.
- Limited maximum resolution.

### 5.4.3 Cable/Connector – VGA

Figure 71: VGA System Connector



**Figure 72: VGA Cable Connector**

#### 5.4.4

#### Troubleshooting - VGA

When VGA was first introduced for the original IBM\* PC, the monitors were CRT and did not have a way to identify their capability via a DDC and EDID. This caused (and continues to cause) difficulty for the system firmware and driver to identify the display capability. It is highly recommended to use VGA monitors that have EDID capability or the display may not be able to be used properly. Most VGA issues can be traced to a lack of an EDID, either caused by the display used, the cable (some VGA cables do not provide the connections for the DDC), or an improperly implemented VGA port. Always follow design guidelines when implementing a VGA port, or better, use a modern interface like DisplayPort\*. If you are using an EDID-less VGA display, it may be required to modify the VBT to operate the port as EDID-less and define a DTD for the display. This process is explained in the "EDID-Less Displays" section of this document.

§

## 6.0 EDID-less Displays

An EDID, or Extended Display Identification Data is a way for a display to inform the graphics driver about itself. It provides details such as manufacturer, model, supported resolutions / timings and other details about the display. This allows the graphics drivers to dynamically adapt when different displays are attached. EDID is HIGHLY recommended for the best operation of external monitors as well as built in LCD panels typically found on notebook system. EDID format is defined by VESA and you can find details on EDID online at sources such as Wikipedia\*.

A display that does not have the capability to send identification and timing information to the driver requires that DTD information to be defined. When a display does not provide a way to identify it automatically via an EDID, the configuration will need to provide it somehow. Intel® graphics firmware (VBIOS and/or GOP) and the drivers do provide a way for you to configure the minimum details needed to drive the display. The particular timing information will need to come from the display provider (not Intel®). Panel datasheets are a good source for this information especially for eDP panels which can vary widely in their requirements.

### 6.1 LFP EDID-less Settings

There are two ways of setting LFP EDID-less displays based on the generation of platform that is being used.

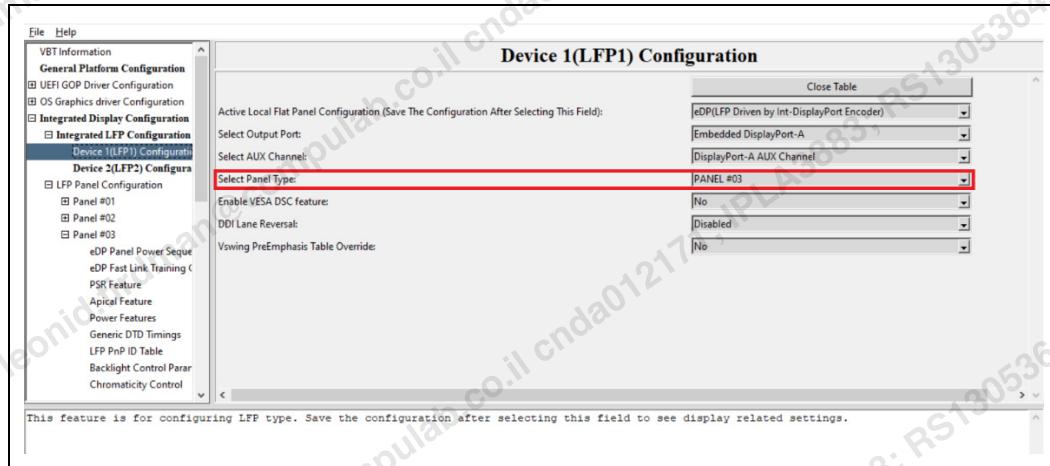
#### 6.1.1 EDID-less settings for Gen11 and newer platforms

The following will show you the typical settings needed to be done to VBT (with the DisCon tool) and BIOS. VBT changes are required for the GOP and the BIOS OpRegion capability is required for the graphics driver. More details below.

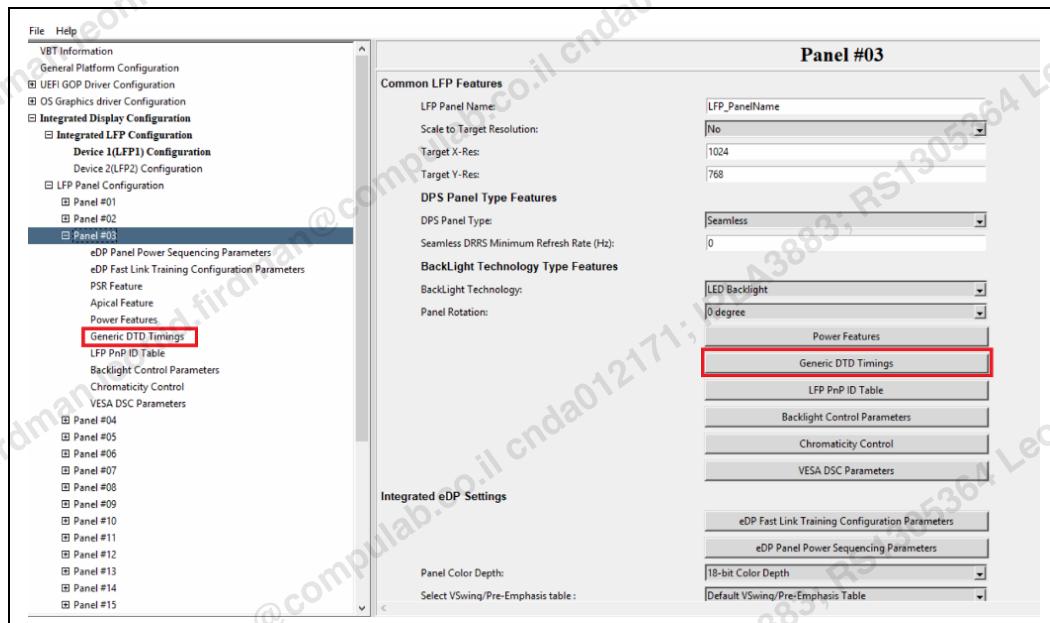
##### 6.1.1.1 VBT changes for GOP

If LFP port is enabled and GOP detects that the panel does not have an EDID, GOP will automatically use the DTD timings for the selected panel type. The selected panel type is either set in the VBT or by BIOS (user settings in BIOS).

For VBT changes for GOP, please refer to the screenshots below.

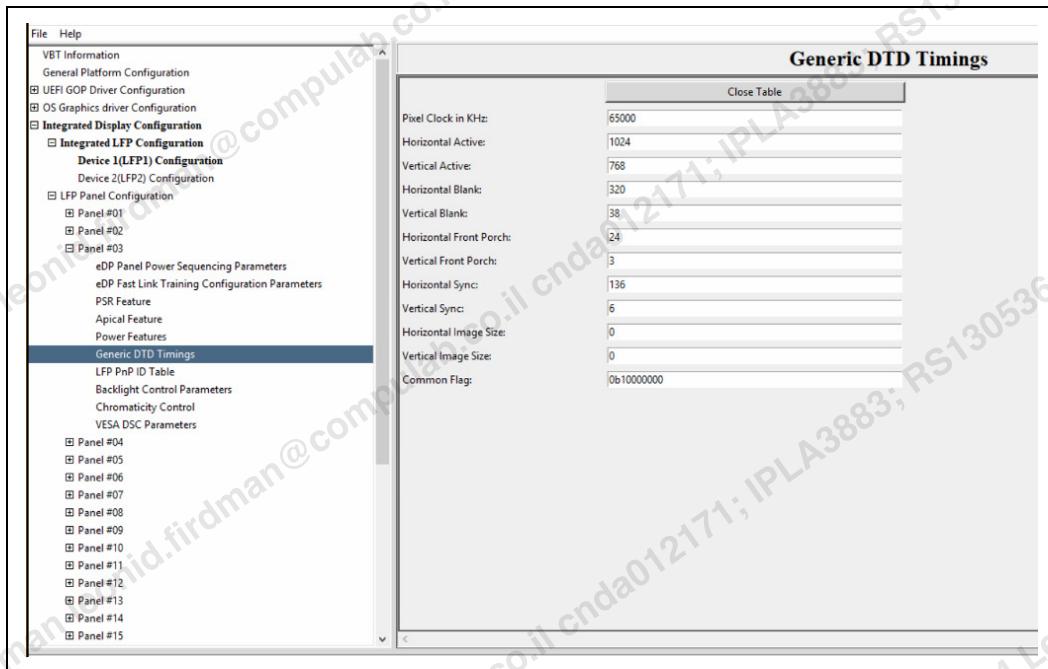
**Figure 73: Set "Panel Type #"**


The DTD information is placed in the "Generic DTD" field for the enabled panel (example Panel #03):

**Figure 74: Generic DTD Timings Selection**


Example default DTD data for Panel #03 which is a 1024 x 768 panel:

**Figure 75: DTD Entries**



If the panel has an EDID, the GOP will use the EDID information. If the GOP detects that the panel does not have an EDID, then it uses the above Generic DTD timings of the specified Panel#.

**Note:** Some BIOS have an option to allow the user to set the Panel# and if that is done, please make sure to modify the “Generic DTD timings” of that Panel# in VBT rather the default Panel#. For example, if panel #1 is chosen in BIOS, then modify “Generic DTD timings” of Panel #1 in VBT.

### 6.1.1.2 BIOS OpRegion Capability for graphics driver

When an EDID-less panel is connected, EDID for this panel needs to be defined in BIOS OpRegion MailBox 5 for graphics driver to use. Your BIOS vendor (IBV) should have details on how this works. There is also a document, “Integrated Graphics Device OpRegion BIOS Specification” (#621530), on RDC for more details.

### 6.1.1.3 Special considerations for Elkhart Lake

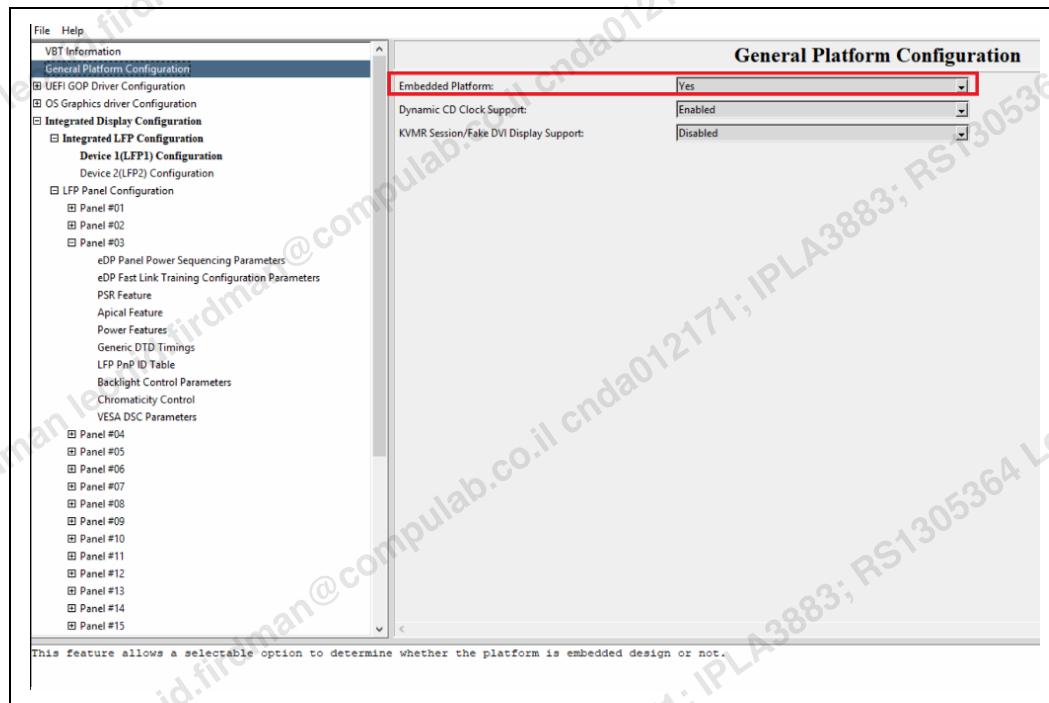
Although Elkhart Lake is a Gen11 platform, the VBT option for GOP is similar to Gen9 platform such that it includes the “Local Flat Panel (LFP) EDID support” option. However, the graphics driver uses EDID defined in BIOS OpRegion MailBox 5 as above.

### 6.1.2 EDID-less settings for Gen9 and older legacy platforms

The following will show you the typical settings needed to be done to a VBT (with the DisCon / BMP tool) to enable EDID-less operation.

**Note:** In the VBT, it is sometimes necessary to define the platform as embedded to enable the EDID-less operation:

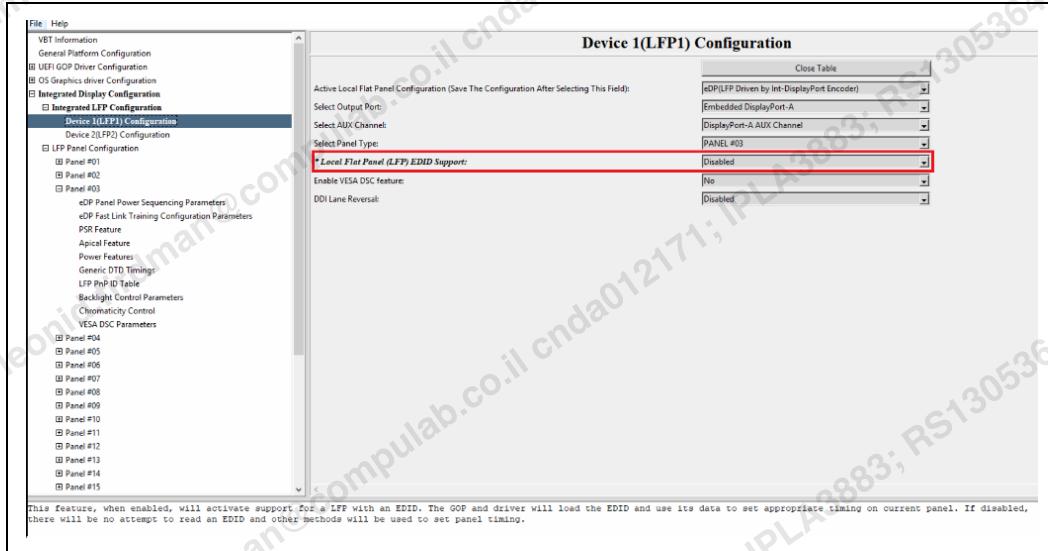
**Figure 76: Setting "Embedded Platform" Option**



For the LFP (i.e. eDP) to operate EDID-less, set the EDID Support to disabled:

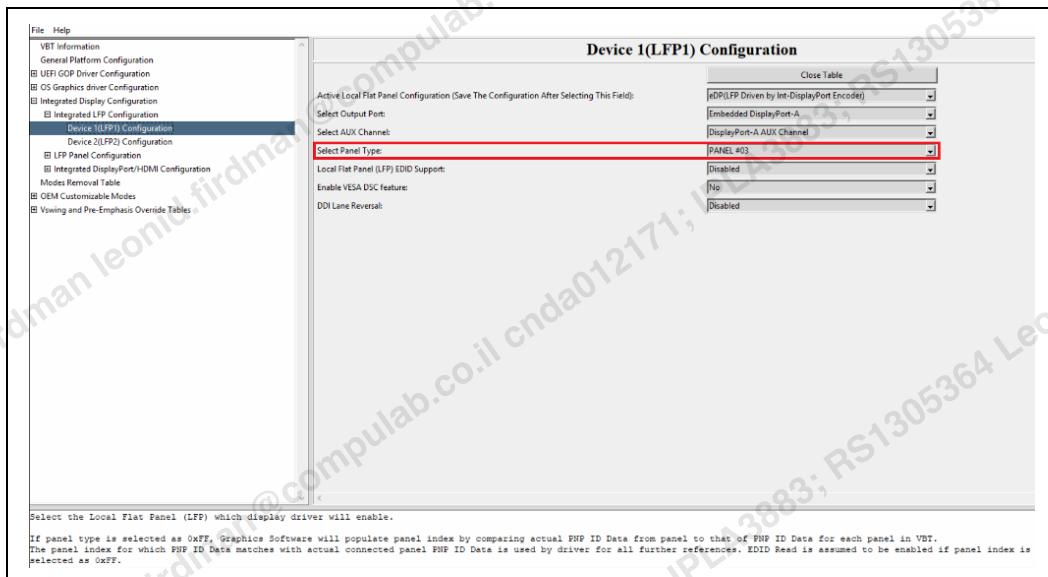
## EDID-less Displays

**Figure 77: Disabling EDID Read**

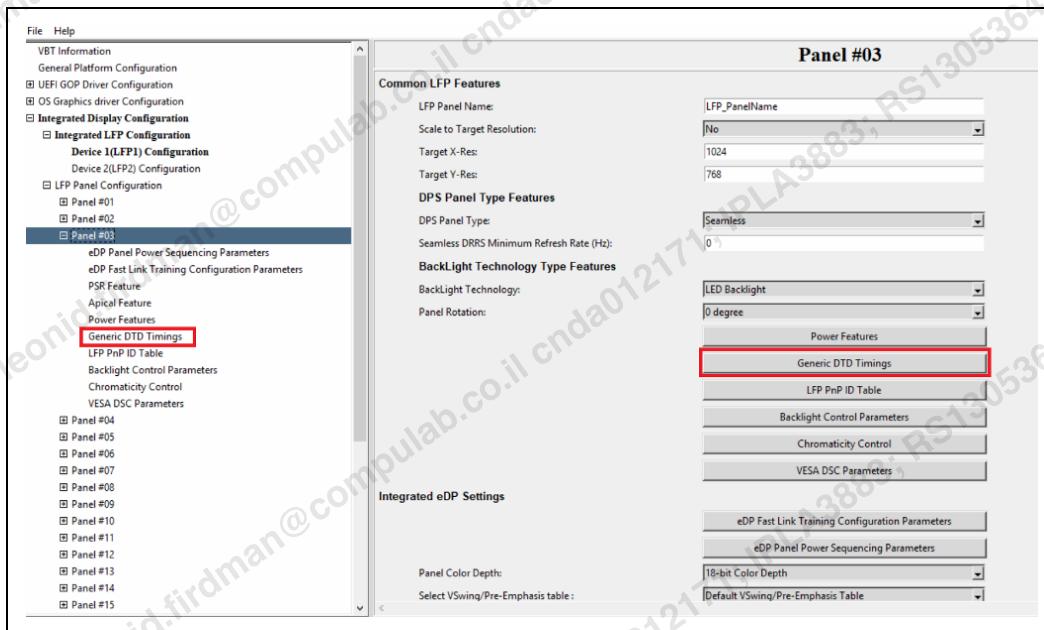


The Device Configuration sets the panel type that provides the DTD information:

**Figure 78: Set "Panel Type #"**



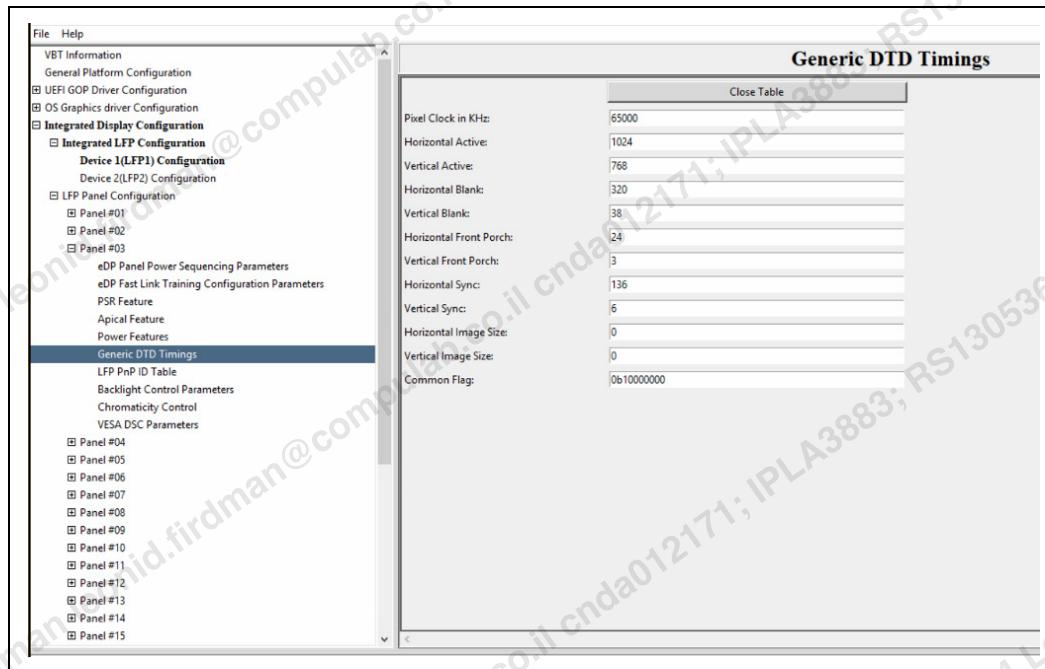
The DTD information is placed in the "Generic DTD" field for the enabled panel (example Panel #03):

**Figure 79: Generic DTD Timings Selection**

## EDID-less Displays

Example default DTD data for Panel #03 which is a 1024 x 768 panel:

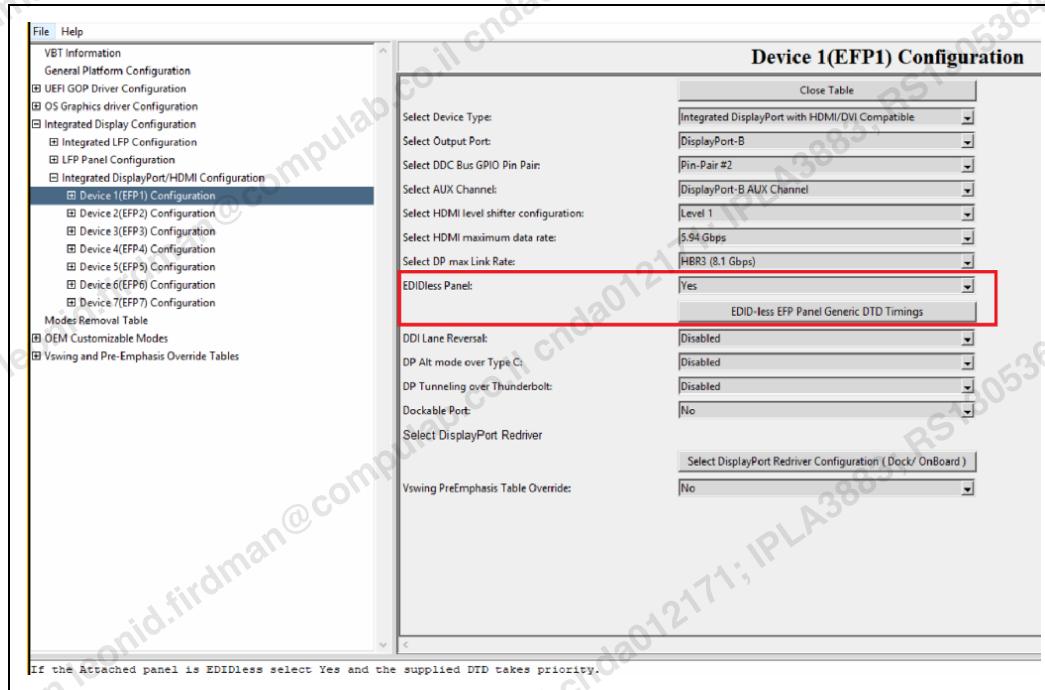
**Figure 80: DTD Entries**



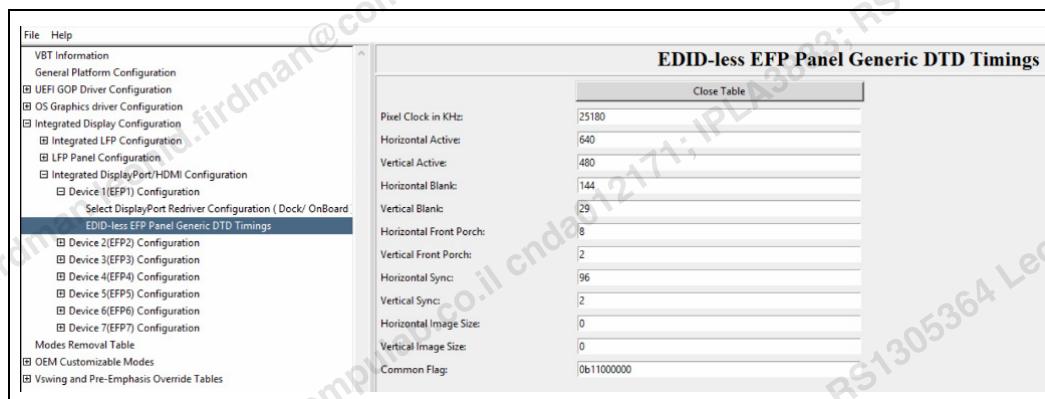
## 6.2

### EFP EDID-less Settings

For an EFP (example DVI, DP, HDMI, etc.), EDID-less operation is set by setting "EDID-less Panel" to Yes as follows:

**Figure 81: EFP EDID-less Option**


The DTD configuration is set in the "EDID-less EFP Panel Generic DTD Timings".

**Figure 82: EFP DTD Entries**


EDID-less operation should only be enabled for very specific configurations where it is absolutely necessary. It is better if you use LFPs and EFPs that have an EDID that are correct, so the automatic EDID configuration operates properly. EDID-less operation imposes limitations on your flexibility and makes the configuration more complex. Getting the correct DTD values can be difficult as not all display providers provide the information in a usable format. Intel® is not able to provide you the information for third party hardware nor do we have the ability to do custom DTDs for you.

## §

## 7.0 Collage Mode aka Combined Desktop

Collage display or "Combined Desktop" as it is now known, is a feature that allows users to combine multiple monitors to create a single unified desktop user interface to achieve a bigger viewing area and resolution. This capability allows user to enjoy higher quality pictures and videos with multiple lower resolution supported panels. It is very cost effective in comparison to purchase of one single high-resolution panel.

Multi-monitor display is a current market trending segment and finds commercial viability in the fields of digital signage, financial, stock market, medical, transportation, call center & surveillance among others. With Intel® Processor Families that have the Collage/Combined Desktop feature (starting from 3<sup>rd</sup> Generation Intel® Core™ processor family), it opens the potential to expand display support beyond clone/extended displays. The Bezel pattern scheme is used to view image flawlessly across all the monitors despite the physical limitations of the monitors.

If multiple displays (minimum two displays) are placed horizontally, it is known as Horizontal Collage/Combine Desktop (i.e., positioning displays one next to another where one image stretches across panels, as shown in the following:

**Figure 83: Typical Horizontal Collage Display**



If the displays are placed vertically, it is known as Vertical Collage/Combined Desktop (i.e., where the upper part of the image is shared across the vertical direction, as shown in the following figure:

**Figure 84: Typical Horizontal Collage Display**

## 7.1

### What is the Maximum Resolution of Collage Displays without External Splitter?

Maximum resolution for Collage Display is always dependent on type of Display connected to a system and also the system's Platform capability (Processor/Chipset).

Following are generalized formulas for a system:

#### 7.1.1

##### Horizontal Display

Assuming all the connected display devices are of same model, maximum collage resolution will be  $(N * W) \times H$ . Here N is the number of displays; W and H are width and height of the maximum resolution supported by the individual display port and/or displays. The maximum resolution the port can support can be found in the EDS for the Intel<sup>®</sup> processor you are using.

If the connected display devices are not the same, the maximum collage resolution will be:

$(\text{Sum of widths of the maximum resolutions supported by the displays}) \times \text{Minimum of Heights of the highest resolutions supported by the displays}$ .

### 7.1.2 Vertical Display

Assuming all the connected display devices are of same model, maximum collage resolution will be  $W \times (N * H)$ . Here N is the number of displays; W and H are width and height of the maximum resolution supported by the port and/or displays.

If the connected display devices are not the same, the maximum collage resolution will be:

Minimum of Widths of the widths of highest resolutions supported by the displays x

(Sum of heights of the maximum resolutions supported by the displays).

## 7.2 Things to Keep in Mind

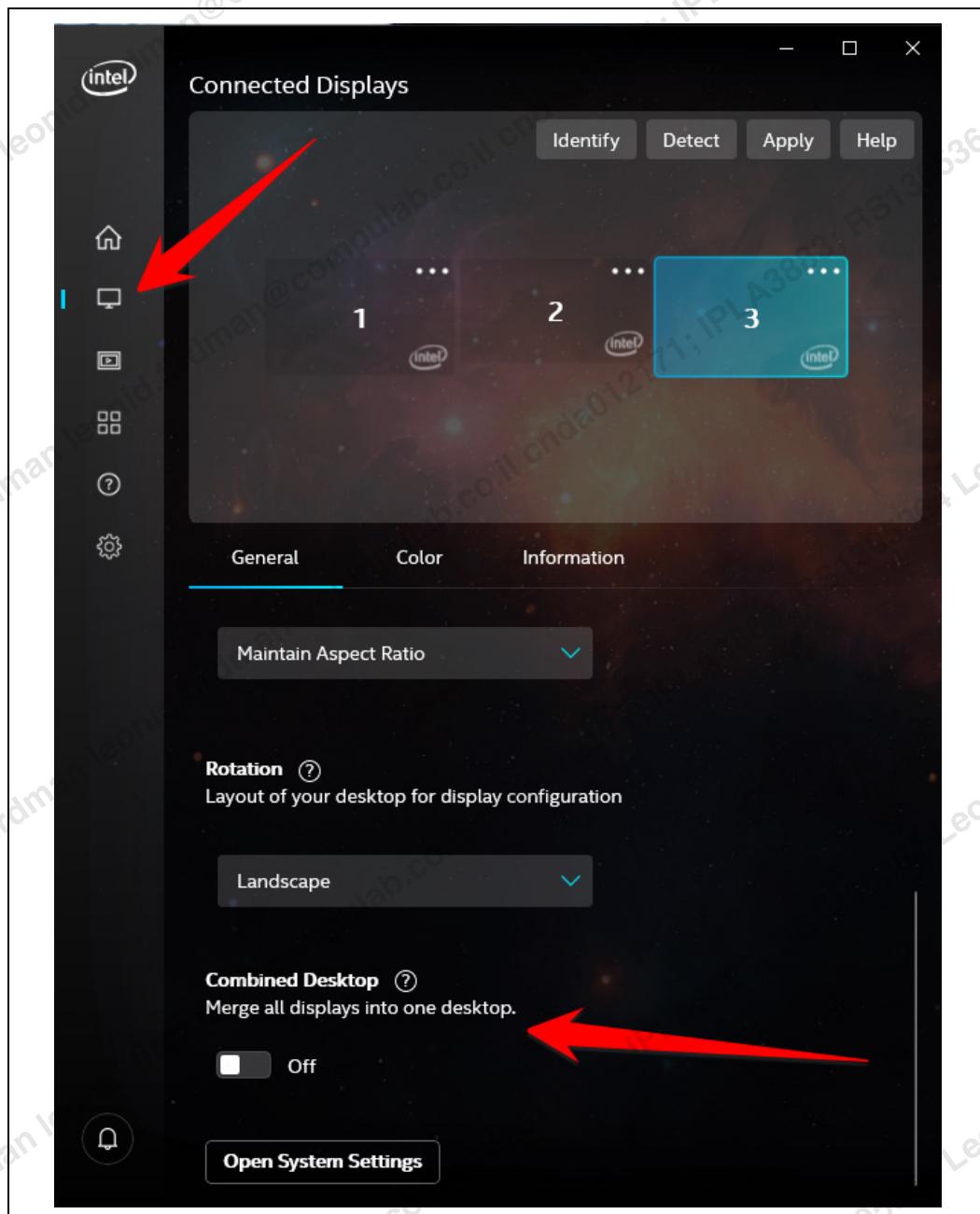
The number of displays supported depends on the number of active displays a given processor supports and the number of MST Splitters used. Without a splitter, the following combinations may be possible: 1x2, 2X1, 1X3, 3X1, 1x4 and 4x1. With MST Splitters it increases the number of displays that can be supported however, most MST Splitters will only split a landscape display horizontally.

- Collage mode does not support Content protected playback from Blu-ray\* disk or DRM content because of limitations to the content protections mechanisms.
- Collage mode does not support Touch enabled Feature across all monitors in Microsoft\* Windows\* 8 operating system.
- The displays being combined must be in landscape mode only. Displays that are rotated are not supported.
- Performance of video/game playback in collage mode will vary in accordance with workload of the platform.
- Audio splitting across all monitors is not possible.
- Hot plug and Scaling are enabled to support this feature but to a Maximum resolution of the capabilities of the specific processor (see EDS) across each monitor.
- Intel® WiDi/Miracast\* or InTru™ 3D is not supported.
- Media playback (Non premium content) is supported but may be limited to certain refresh rates.
- Is supported in Switchable graphics and Hybrid graphics designs.
- Bezel Correction is supported.
- MST is only supported on a DP port, but the splitter may convert the split DP signal to another display interface such as HDMI or DVI.

### 7.3 How to Enable Collage/Combined Desktop

Combined desktop can be enabled through the Integrated Graphics Control Center (IGCC) or the Graphics Control Panel (GCP) as follows. Shown is Horizontal display configuration:

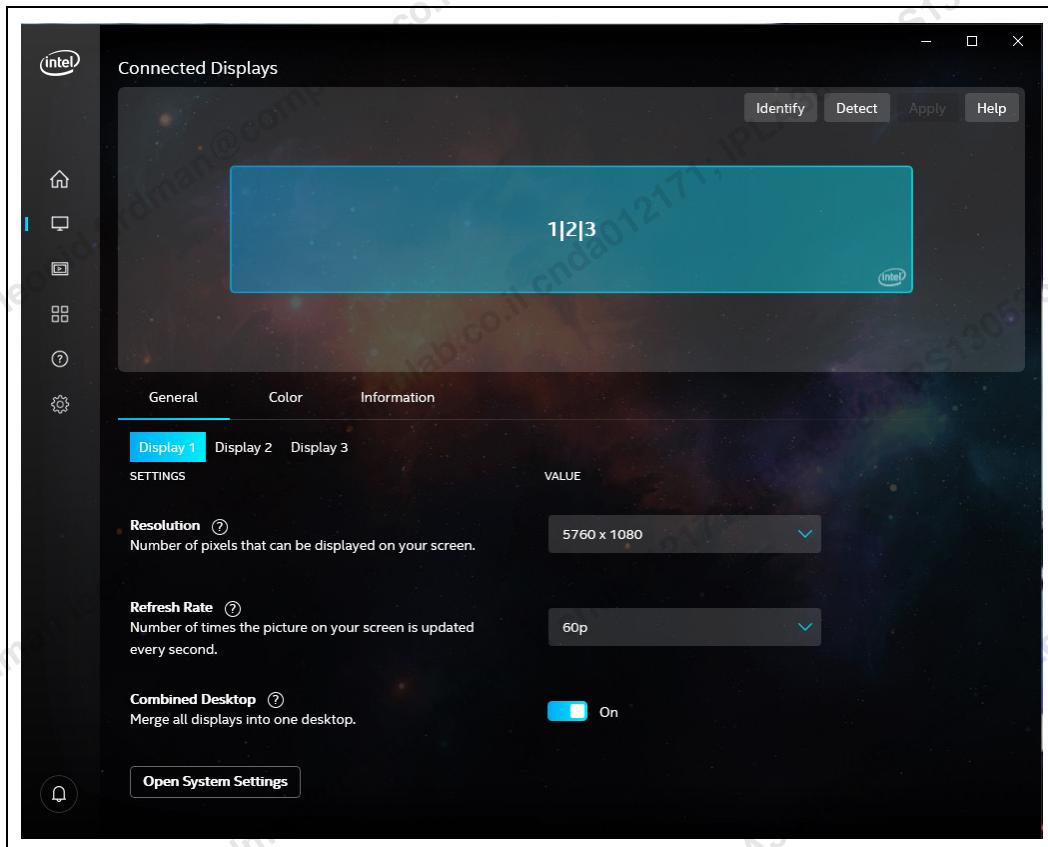
Figure 85: Arrange Displays and Enable "Combined Desktop" or "Collage"



### Collage Mode aka Combined Desktop

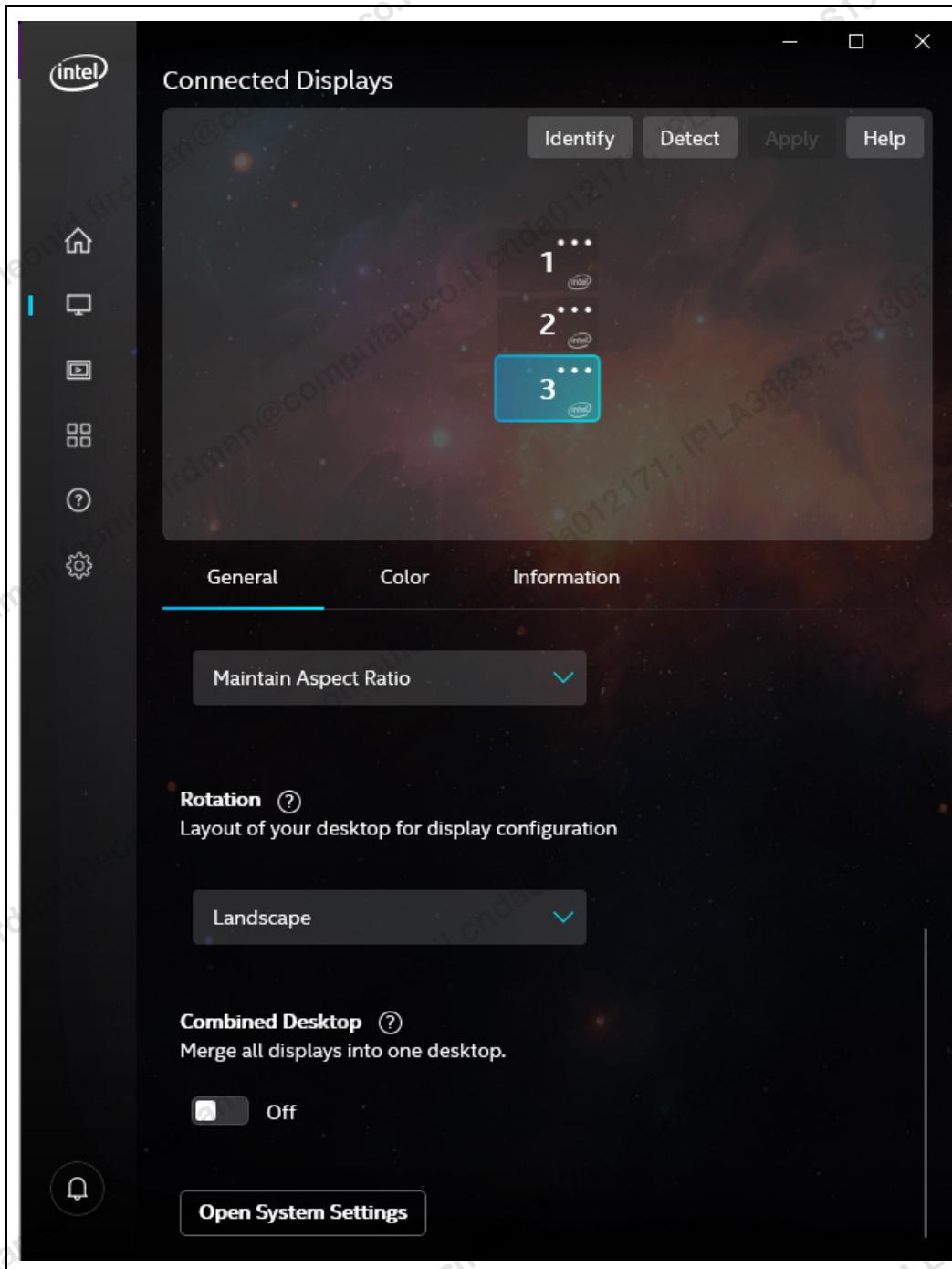
At this point, the displays will be grouped as shown:

**Figure 86: Combined Desktop Enabled in Horizontal**



The following figure shows a vertical display configuration.

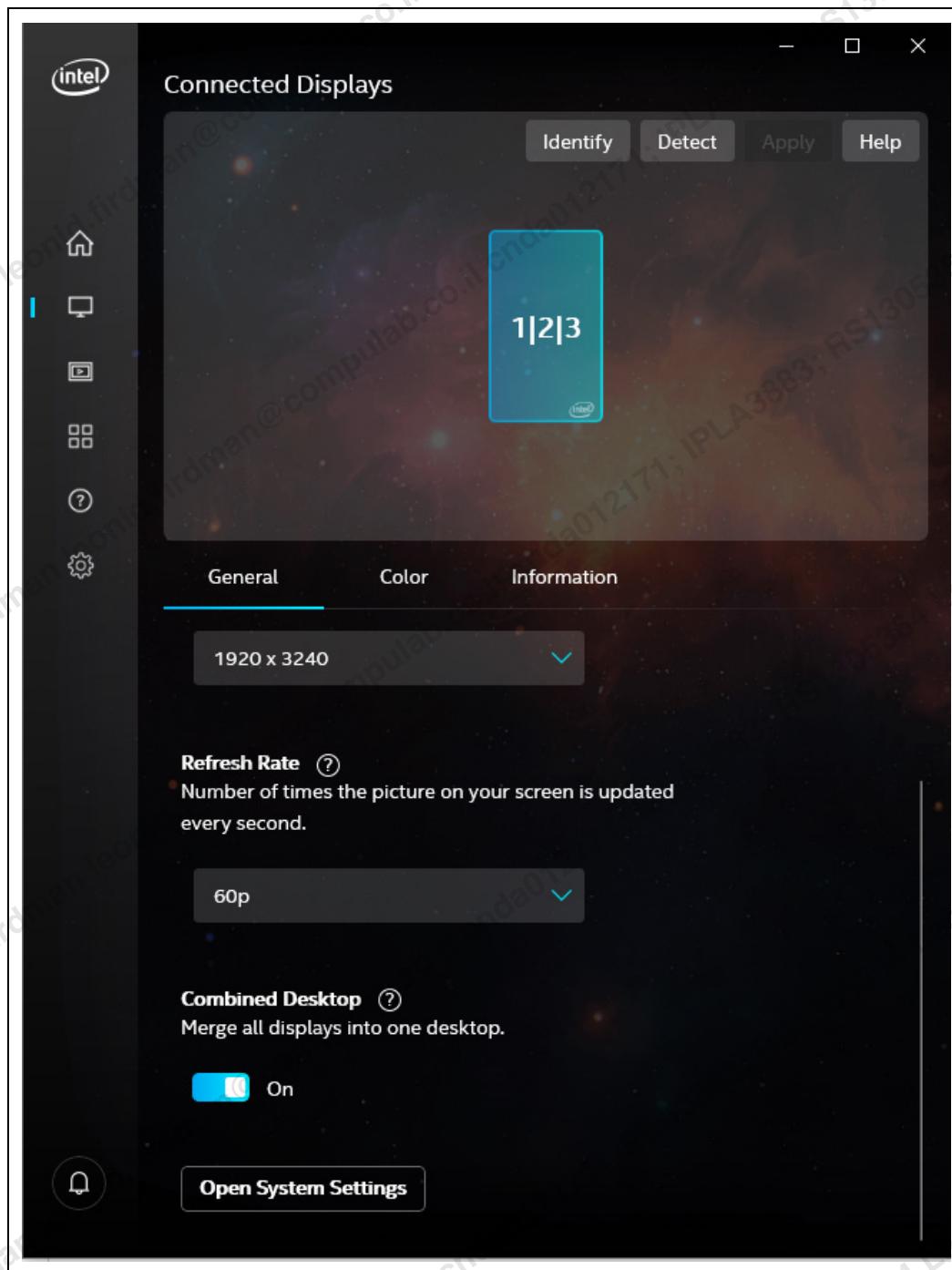
Figure 87: Combined Desktop Enabled in Vertical



**Collage Mode aka Combined Desktop**

And after Combined Desktop is enabled for vertical configuration:

**Figure 88: Combined Desktop Enabled in Vertical**



## 7.4 Combined Desktop Limitations

The following are the limitations of the Combined Desktop:

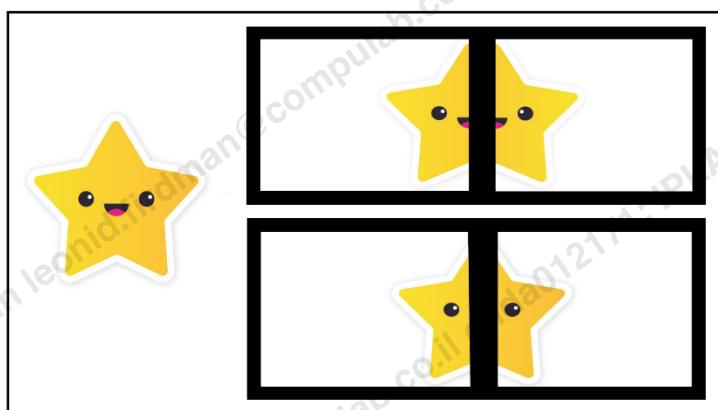
- This feature will use the lowest common settings for all displays - for example, if you have a 4K monitor and a 1080p monitor, it will use the 1080p settings for both.
- Intel® Graphics Command Center currently supports Combined Desktop in only horizontal and vertical arrangements.
- Disabled monitors cannot be enabled when Combined Desktop is active.
- Monitors connected to non-Intel® Adapters are not supported by this feature, so displays plugged into external graphics adapters must be disabled before Combined Desktop can be turned on.
- Monitors are aligned with the top left corner, so monitors of a different size should be aligned along the top edge (horizontal layout) or left edge (vertical layout) to be considered a valid configuration.
- Starting a game in full-screen mode with Combined Desktop may cause the displays to enter Mirror mode instead.

## 7.5 Bezel Correction

This feature allows you to adjust the image to correct for the physical monitor bezel so images that span multiple monitors can be visually adjusted. Current versions of the Intel® Graphics Command Center (IGCC) now support this capability.

The following figure shows an illustration of how bezel correction can be used. It shows an example of an image you might want to display on the left. The top right illustration shows how the image would look without enabling bezel correction. The bottom right illustration shows how the image might appear after enabling bezel correction and setting the value to a desired number.

**Figure 89: Bezel correction example**

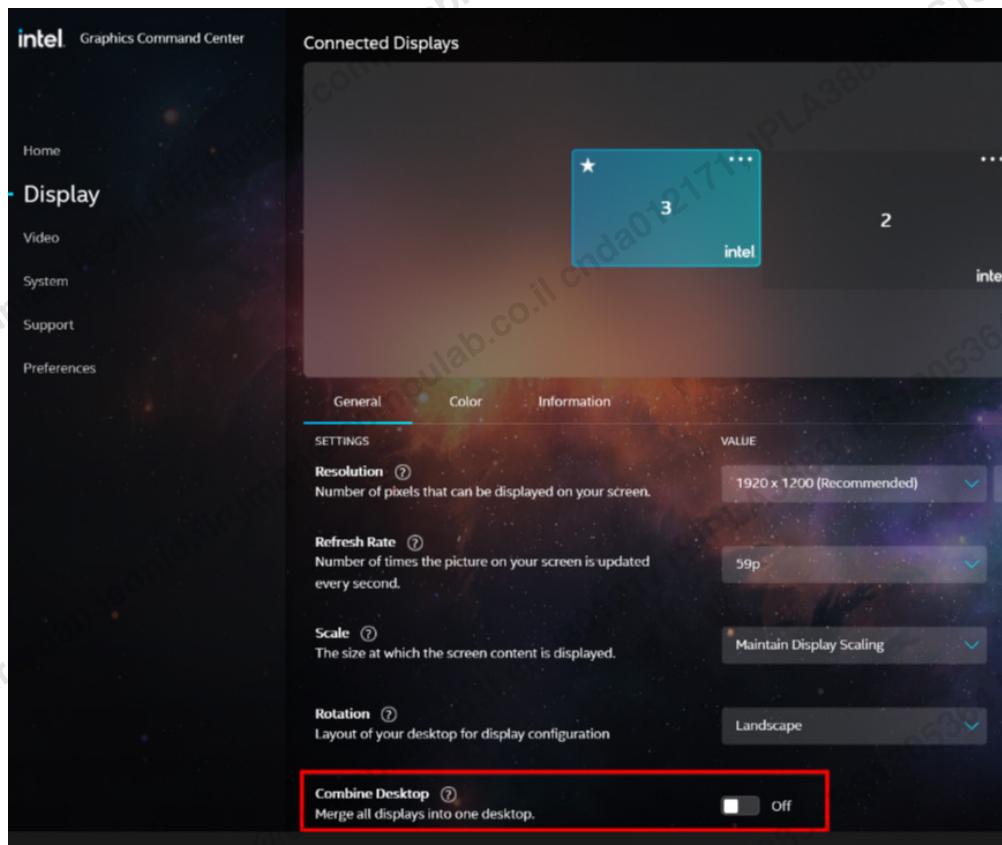


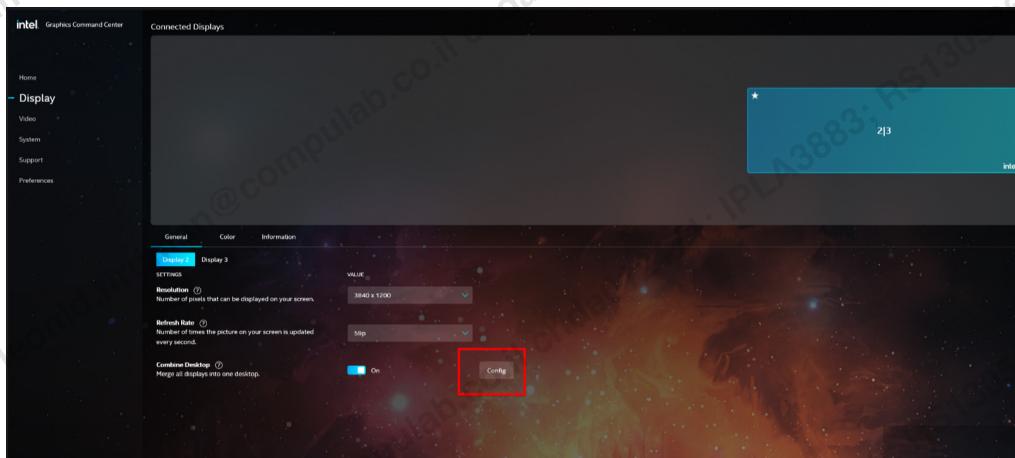
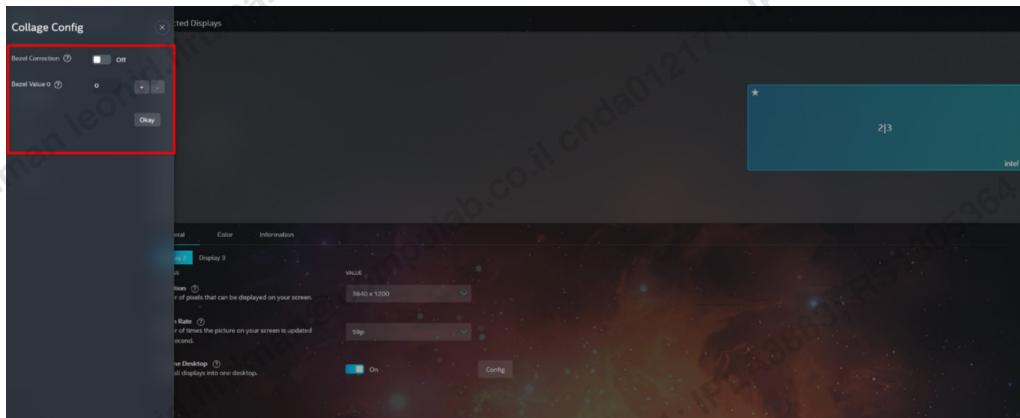
### Collage Mode aka Combined Desktop

To enable bezel correction in IGCC follow the below steps:

1. Open IGCC application.
2. Access "Display" tab.
3. Turn on "Combined Desktop" option as seen in Figure 90 below.
4. Click on Config option as seen in Figure 91 below.
5. Turn on Bezel correction and set a value to the desired number that corresponds to your display requirements as seen in Figure 92 below.

**Figure 90: Combined Desktop Option in IGCC**



**Figure 91: Config option for Combined Desktop****Figure 92: Collage/Combined Desktop Config Options**

§

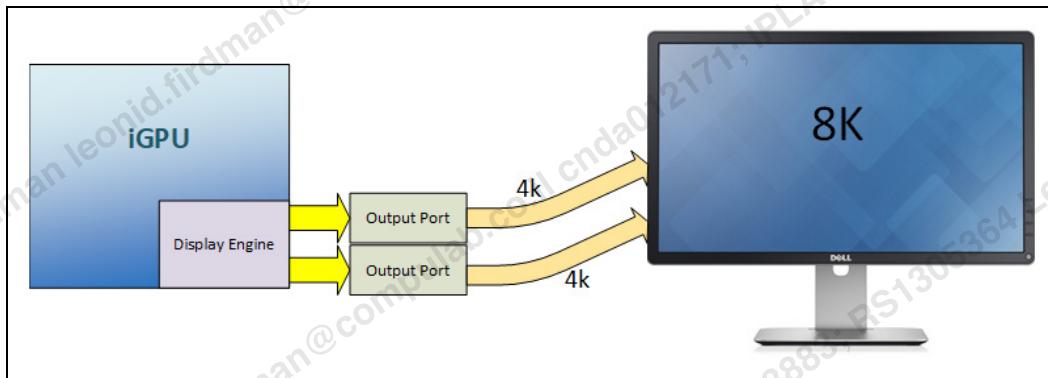
## 8.0 8K Resolution Support

As of publication of this document, the topic of 8K resolution is a bit tricky. Currently there are two ways to display 8K on a single monitor.

The most prevalent as of 2020, is for the monitor to plug into two monitor ports which both run at 4K and the monitor combines them into a single display. This is considered a "Two Pipe, Two Port" solution and can be accomplished with any Intel® iGPU that supports 4K output.  $4\text{K} + 4\text{K} = 8\text{K}$ . The 8K monitor must do the combining and have the TWO display inputs to support this.

### 8.1 Two Pipe Two Port

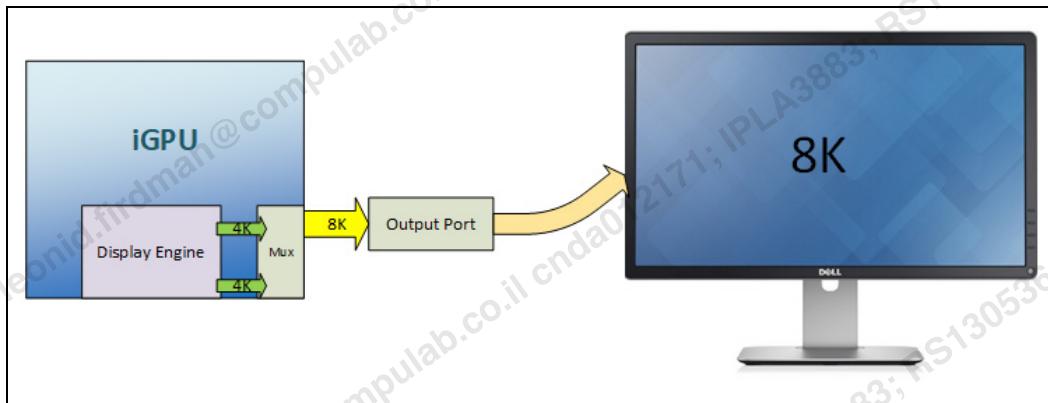
Figure 93: Two Pipe Two Port 8K



Some newer Intel® products like Tiger Lake, have the ability to internally MUX together the two 4K pipes and output a single 8K output. The 8K monitor used must support a single 8K input. This is considered a "Two Pipe, One Port" solution. This still uses two of the iGPU pipes therefore the one display is counted as being two of the total number of displays supported. For example, with Tiger Lake, which can support 4 simultaneous displays, an 8K monitor uses up 2 outputs internally therefore only three displays total are possible when one is 8K. If two 8K displays are used, then only those two displays can be enabled simultaneously because each monitor effectively uses 2 of the 4 simultaneous displays.

## 8.2 Two Pipe One Port

Figure 94: Two Pipe One Port 8K



## 8.3 Future

It is expected that future products will support 8K in a One Pipe One Port configuration but again, that is in the future. Higher resolutions beyond 8K are also likely to be supported using similar types of architectures. Refer to the future EDS for the specific Intel® processor family you are interested in for details on how this might be supported.

§

## 9.0 High-bandwidth Digital Content Protection (HDCP)

HDCP is an Intel® initiative that protects uncompressed digital content being transmitted from a PC or other source device for display on TVs and monitors using DVI, HDMI, or VESA DisplayPort display protocol. In a HDCP system, two or more HDCP devices within the system are required to have their own set of private HDCP keys which will be validated during an authenticated process before the protected content can be displayed. The graphics driver supports HDCP via COPP\*, OPM\*, PAVP and GPU-CP interfaces.

HDCP relies on three major factors to protect content:

1. Authentication – assures only licensed devices receive protected content.
2. Encryption - data sent over the display interfaces prevents the data from being intercepted between the system and display.
3. Key Revocation – stops devices that have been compromised and/or cloned from receiving protected content.

For HDCP to work, both the system and the monitor must be able to negotiate the best level of HDCP that can be achieved during Authentication. In other words, to be able to display HDCP protected content, the display must also support HDCP or the protected content will NOT display.

### 9.1 HDCP Levels

HDCP has been improved over time for both security and capability. As of Level 2.2, backwards compatibility requirements with older HDCP versions has been removed for security reasons. HDCP 2.2 and up will only work with devices (like monitors) that also support 2.2 or higher. Refer to specific platform EDS document to determine the HDCP level supported on your specific Intel® product family, and with your sink device (i.e. monitor) to determine what version of HDCP it supports.

### 9.2 O/S HDCP Support

It may be obvious, but for HDCP to work to display protected content, the operating system must also support HDCP content protection. Operating systems like Microsoft® Windows\* have HDCP built in. Other O/S like Linux\* may not natively support HDCP and therefore are unable to display protected content. HDCP may be able to be added to those O/S that do not currently natively support it as there are efforts to create the capability as an “add-on” capability. Check with your O/S provider and/or community group that support the O/S for details.



Contact the BIOS vendor for any specific BIOS changes required to enable HDCP.

## **9.3 Troubleshooting**

To troubleshoot issues in HDCP:

- Blank display or blank window that should be displaying protected content means either authentication failed, or a compromised or cloned display is being attempted to be used. If the Intel® product supports HDCP 2.2 or higher, the display must be 2.2 or higher for protected content to display.
- Using multiple displays attached to a "display splitter" may cause HDCP failure and inability to display protected content.
- HDCP may fail on a laptop if an external display is attached and the native built-in display (i.e. eDP panel) is disabled or the lid of the laptop is left closed.

§

## 10.0 KVM Remote (KVMR)

KVMR (Keyboard Video Mouse Remote) provides an ability to gain remote access of a system that is not directly physically accessible or that may be designed to be "headless" (no display attached).

Example application remote headless system on factory floor:

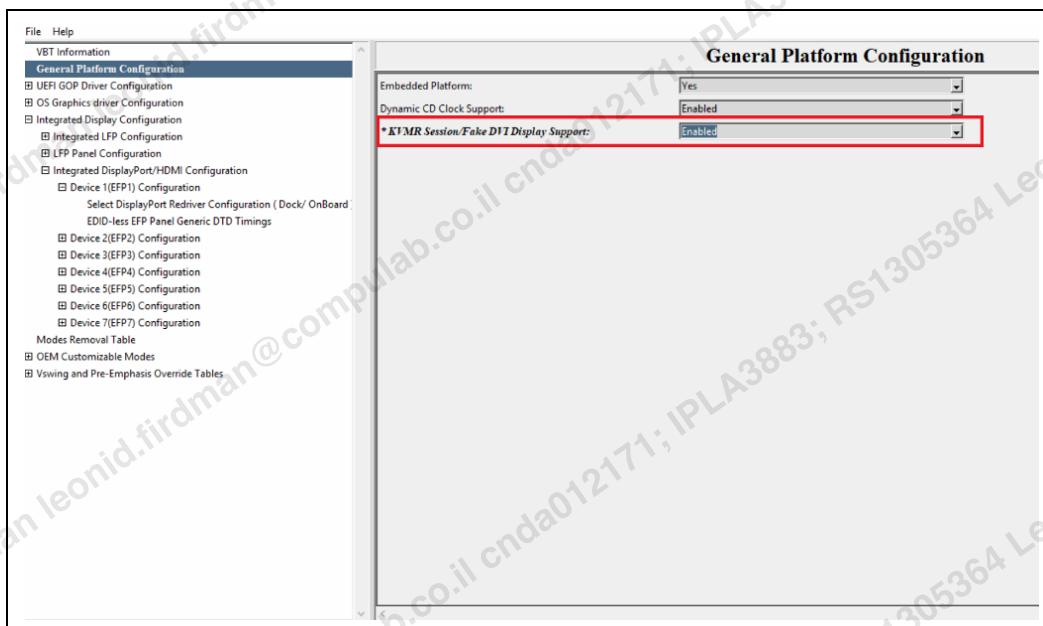
**Figure 95: Example Remote Headless Application**

In the headless situation, there are special settings required in the VBT so that the system will create a "display" capability without a monitor, that can be viewed remotely. If these settings are not used, the Intel® firmware (GOP) and driver will not initialize as that normally means there is nothing to do because there is no display. This prevents a KVMR capability from being able to remote in. Fortunately, modern firmware and drivers can be configured to create the necessary display that can be remote accessed.

When enabled, GOP and graphics driver will keep a display pipe enable even if no displays are attached. When no displays are attached, GOP or Gfx driver will check VBT settings for EFP1/2/3/4 for DVI support. If any EFP setting supports DVI display type, GOP/driver will enable that port. If none of the EFP settings support DVI display type, GOP/driver will enable DVI on port-B by default.

The following screenshot shows how to set KVMR with the DisCon tool:

**Figure 96: KVMR Setting in DisCon**



## 10.1 Enabling KVMR with Platforms that do not have the KVMR Option

Some firmware or drivers for earlier Intel® products may not have an explicit option for KVMR operation. There is a workaround that will allow headless KVMR on those platforms:

In the VBT, set an LFP for EDID-less operation (even though you do not have an eDP attached). This causes the firmware and driver to initialize the non-existent panel which gives the KVMR routines a 'display' to share remote. Refer to the "EDID-less Displays section of this guide for details.

This "phantom" eDP will still be present even if you attach any actual displays.

You may also need to tweak your "Child Device List" to make sure the LFP is not the primary display when a real display is attached. It can be primary when it is the only 'display' active.

§

## **11.0 High Dynamic Range (HDR)**

---

HDR is supported starting with 7th Gen Intel® Core™ processor platforms (Kaby Lake) which are the first Intel® integrated graphics platforms to support HDR. Laptops, desktops and 2-in-1s manufactured by OEMs using this product are available in the market today. Intel® first launched support for HDR in the form of Ultra HD Blu-ray\* playback in 2016 on Kaby Lake platforms on Windows\* 10. Since that time, Intel® and Microsoft\* have worked together to support OS-native HDR on Intel® graphics starting with the Windows\* 10 Fall Creator's Update (RS3).

Part of achieving an HDR experience requires HDR source content. Some of the key HDR content sources today are:

- Streaming HDR (e.g., YouTube\*)
- Streaming premium HDR (e.g., Netflix\*)
- Local HDR Video Files
- ULTRA HD Blu-ray\*
- HDR games
- HDR content creation apps

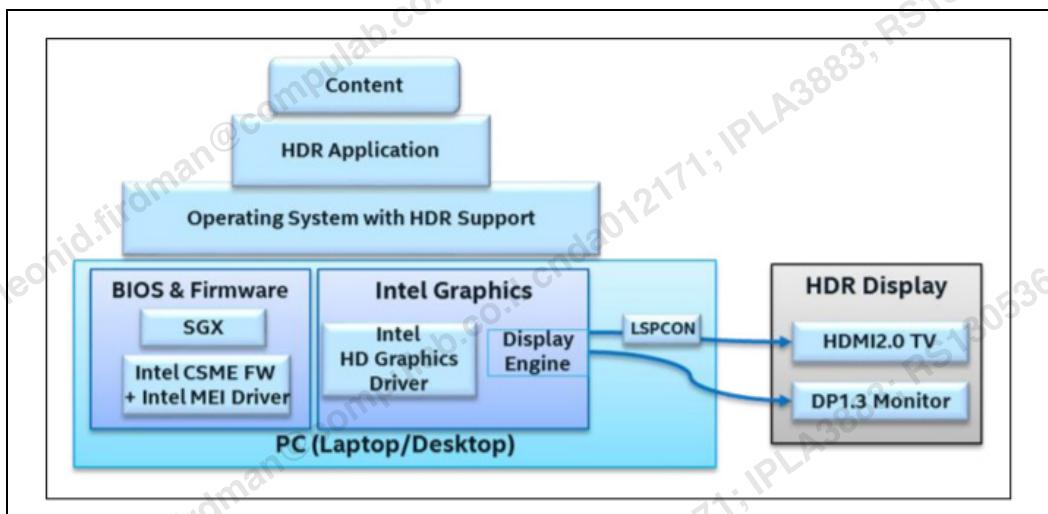
Each of these content types requires a slightly different set of ingredients in order for HDR playback to be supported.

### **System Ingredients:**

The ingredients required for a fully functional stack on Intel® graphics platform are shown in the following figure, along with a brief note on each of the ingredients.

## 11.1 HDR Functional Stack on Intel® Graphics Platforms

Figure 97: HDR Stack



- Intel® HD/UHD Graphics: This is the graphics engine hardware integrated in Intel® Core™ Processor, used to decode and render HDR content. Its display engine transmits HDR signals over HDMI and DisplayPort cables to an HDR display.
- Intel® HD/UHD Graphics driver: In addition to the appropriate HW above, a particular driver version is required. It is always recommended to get the latest graphics driver released on intel.com, or PC OEM website or via Windows\* Update path,
- Operating System: An appropriate version of the Windows\* 10 operating system. Windows\* 10 Fall Creators Update (or newer) is required.
- LSPCON\*: In order to achieve HDR signaling over HDMI starting from 7th Gen Intel® Core™ Processor, an additional hardware component known as the LSPCON\* (Level Shifter and Protocol Converter) must be on the motherboard. This is an ingredient that must be installed by the PC manufacturer and cannot be added by end-users. Intel® graphics HW starting from Tiger Lake will not require LSPCON\* for HDR on HDMI since it supports native HDMI 2.0 output. Note that the LSPCON\* is required only for HDMI, and not for DisplayPort.
  - LSPCON\* FW: The correct version of FW is also required on the LSPCON\*.
- System BIOS: Specifically for Ultra-HD Blu-ray\* playback, the system BIOS must be properly configured today to support Intel® Software Guard Extensions (SGX or the equivalent). It is possible that the system provider turned off SGX by default, in which case the user would have to manually turn it on in the BIOS settings.
- Intel® CSME FW: The Intel® Management Engine (ME) Firmware version is required in order to achieve necessary HW-DRM support and HDCP2.2 (or higher) link protection required for premium HDR video content. It is typically included in the System BIOS owned by the system manufacturer.

- Intel® MEI Driver: This driver must be installed so that software can communicate with the ME FW.
- Application: Specific application and internet browsers (like: Microsoft\* Edge\* or Google\* Chrome\*) are required to play HDR content.
- Content: HDR video files come from different sources. In order to receive HDR content from certain streaming providers such as Netflix, you must have an appropriate plan/account type.
- HDR display: True HDR playback is not yet available on the built-in (eDP) displays of laptops and tablets, although a partial HDR experience known as Extended Dynamic Range (EDR) is available on the built-in displays of a small number of devices today. TVs are of course available across various makes and models.
- Display connector: The physical display interface (connector) on the PC that connects it to the HDR display could be HDMI, DisplayPort, mini-DisplayPort or USB Type-C. For HDMI, version 2.0a is required. It is important to look for HDCP2.2 support, which is needed for premium content to be transmitted to the display.
- Cable/dongle: For PCs with native HDMI or DP connector support, it is straightforward to use an appropriate high quality cable to connect to the display. In case of PCs with USB Type-C port (with Thunderbolt® 3 support or DP Alt mode), an adapter or dongle is required to convert USB Type-C to HDMI 2.0 or DisplayPort, as well as to support HDCP2.2. Similar adapter is required for miniDP connector too. Such adapters are available from 3rd party vendors.

## 11.2

### Activating Native HDR Mode on Windows\*

OS-native HDR support is an opt-in feature on Windows\* 10 Fall Creator's Update (RS3). To enable HDR, go to 'Display Settings' and then slide on "HDR and advanced color" setting.

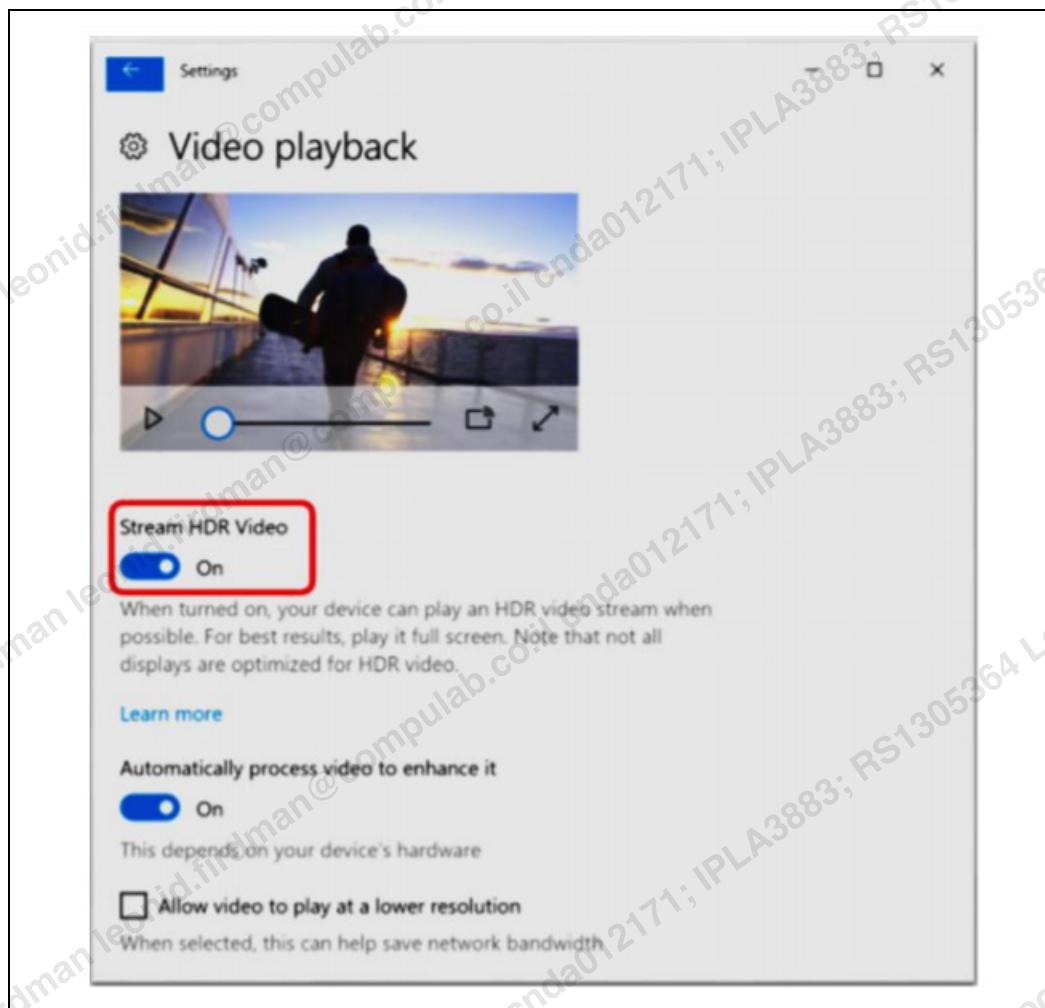
Figure 98: Where to Enable HDR in Windows\*



As soon as you activate this switch, everything on screen will immediately appear darker, and by comparison desaturated or washed out. This is because the vast majority of applications and content are SDR while your system is now in HDR mode. In order for SDR and HDR content to appear at the same time, the SDR content must be shown darker than the HDR. You will also find that if the color gamut of the display is wider than sRGB, transition to HDR mode will make the desktop appear washed out, compared to SDR mode where the sRGB colors looked oversaturated due to color mapping (stretching sRGB to panel's wide color gamut). This approach allows HDR content to be properly viewed, with highlights far brighter than the SDR content and also with greater detail in darker scenes. Also, if you need to stream HDR contents with applications like Netflix and YouTube, make sure "Stream HDR Video" setting is on in the 'Video playback' settings page.

## 11.3 Screenshot Showing 'Stream HDR Video' Enabled

Figure 99: Example HDR Enabled



Additional details can be found online at

<https://www.intel.com/content/www/us/en/support/articles/000025998/graphics.html> and

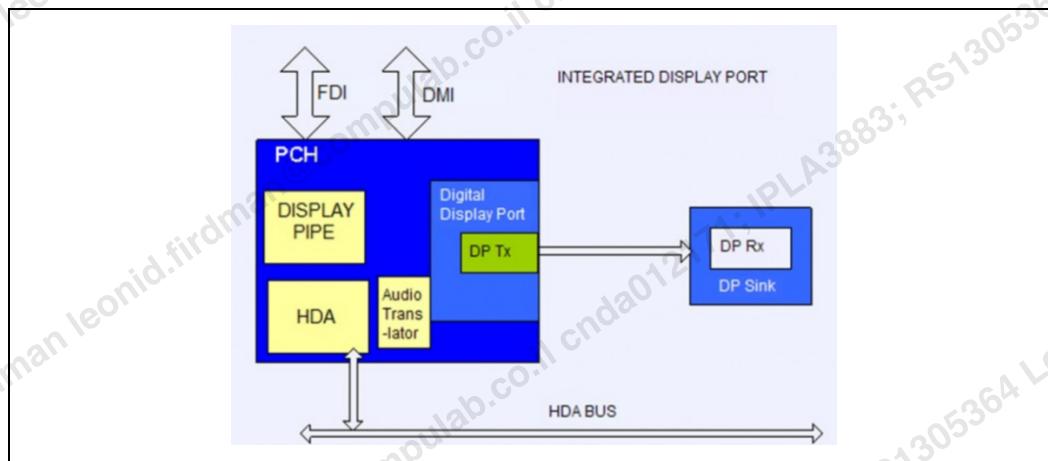
[https://www.intel.com/content/dam/support/us/en/documents/graphics/HDR\\_Intel\\_Graphics\\_TechWhitePaper.pdf](https://www.intel.com/content/dam/support/us/en/documents/graphics/HDR_Intel_Graphics_TechWhitePaper.pdf)

## 12.0 Display Audio

Both DisplayPort (including over Type-C) and HDMI have the ability to output system sound to speakers in the monitor (or attached to the monitor). To accomplish this, Intel<sup>®</sup> typically provides a "Display Audio" driver along with the graphics driver that should be installed during the graphics driver installation process.

This driver synchronizes the audio with the display data on the display interface.

**Figure 100: Example DP Data Flow (from Display Audio SPS)**



This driver is designed to support basic audio capability out to the monitor. It is designed to work with all supported display link rates. It does support Lip Sync (for movie playback) on HDMI. The specifics for support are detailed in the "Display Audio SPS" document for the particular Intel<sup>®</sup> processor family you are using. Contact your Intel<sup>®</sup> representative for availability to you.

The Display Audio driver can coexist in a system along with the HD Audio driver (or equivalent) but should not be confused with the HD Audio or other expanded audio capability drivers.

Display Audio does not support the following:

- Sound output to audio jacks (headphone or speaker jack).
- Microphone input.
- Other advanced audio capabilities.

Currently, the Display Audio driver will be installed with the graphics driver from the Intel<sup>®</sup> distribution using the setup utility for installing the graphics driver.

## 12.1

### Intel DAC vs ACX DAC

Intel and Microsoft\* have jointly decided to replace Intel Display Audio Codec (DAC) with Microsoft\* Inbox Audio Class Extension (ACX) Display Audio Codec Driver (DAC) and SGPC for better interoperability. As part of this change, display audio functionality will be supported through ACX Inbox DAC driver from Microsoft\* (released with 19H1 OS).

**Target Platform:** Tiger Lake using Windows\* 19H1 OS onwards

#### Prerequisites:

- Shared Graphics Power Component (SGPC) is a prerequisite (enabled by default in OS and GFX) for ACX DAC driver support.
- Minimum OS requirement: 19H1
- Bus driver support: Both MSFT\* and iSST.

#### Impact:

DAC load behavior change

- ACX DAC driver will not load on LFP only configurations.
- ACX DAC driver will be loaded only when audio capable EFPs are connected and active.
- MSFT\* ACX DAC driver is visible under "Sound, video and game controllers" in Device Manager as "HD Audio Driver for Display Audio".

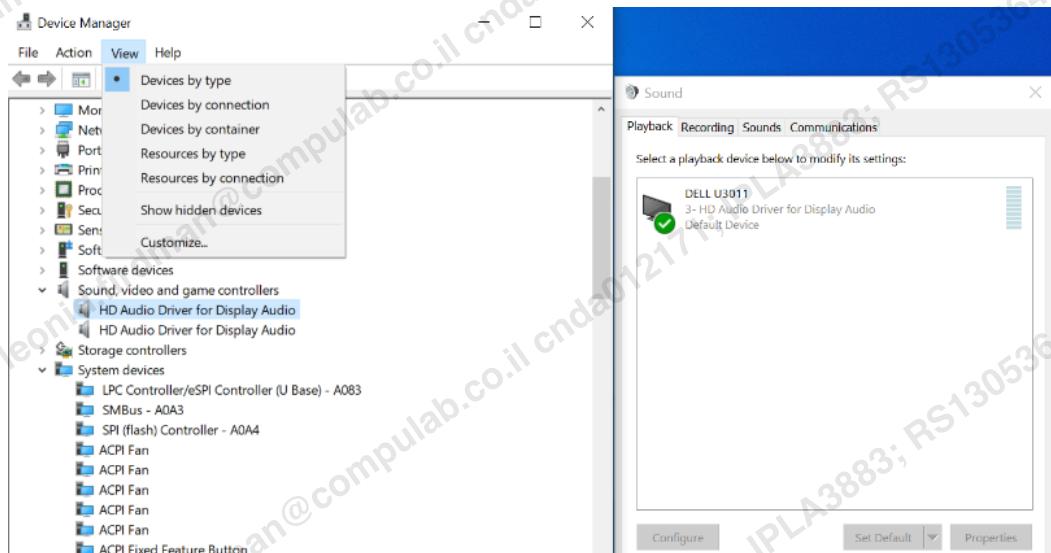
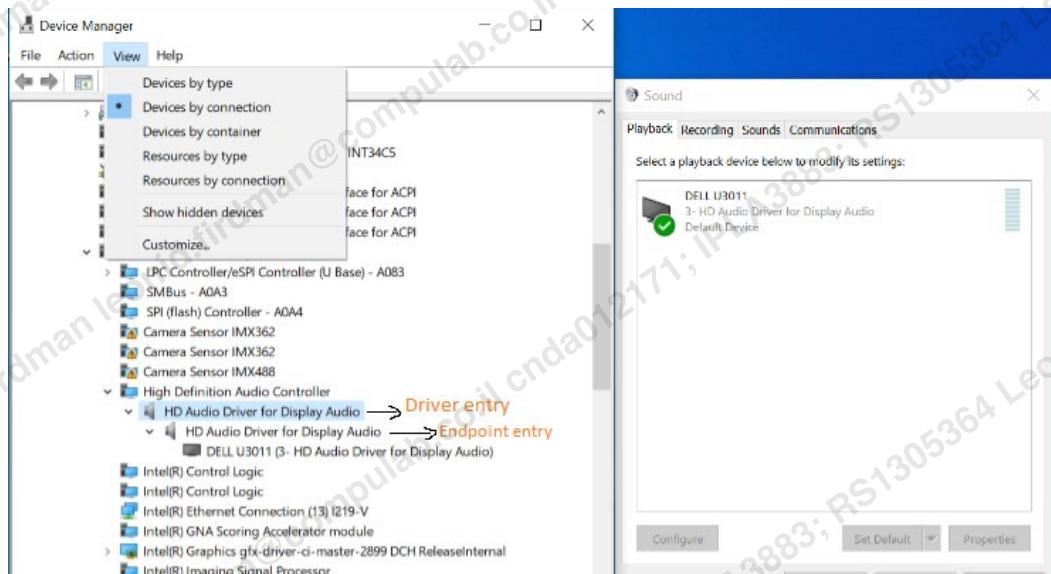
Name is subject to change by MSFT\*.

**Note:** MSFT\* ACX DAC driver is supported on Tiger Lake platform onwards with OS version 19H1 onwards. Intel DAC driver will not be supported for systems with those configurations unless RS1 LTSC OS or RS5 LTSC OS is POR for that platform on the roadmap.

#### 12.1.1

### MSFT\* ACX DAC Loading Behavior in Device Manager (DM)

Below is the Device Manager window with One Display connected. Under "Sound, video and game controllers" in DM, there are two entries for "HD Audio Driver for Display Audio". First is the Driver entry and Second is the Endpoint entry

**Figure 101:** One display (Device by type)

**Figure 102:** One display (Device by connection)


### 12.1.2

### MSFT\* ACX DAC Version Details

As per the current design, ACX codec version will be shown same as GFX driver version under Device Manager -> HD Audio Driver for Display Audio Properties -> Driver Tab.

But, to get the exact ACX Codec version, go to ACX codec properties → Details Tab → Driver File Details and see the ACX codec version, which should be same as OS version.

Another way to know ACX Codec version:

Go to C:\Windows\system32\Drivers\AcxHdAudio.sys → Properties – Details tab and see the ACX codec version, which matches with OS build version.

## 12.2

## Troubleshooting

To troubleshooting issues with the Intel Display Audio:

- If you used the "Have Disk" installation method for the graphics driver, the Display Audio driver may not be automatically installed. In this case you will either need to use the Intel® Graphics Driver Setup or do a "Have Disk" install of the audio driver from the audio folder of the graphics installation files.
- Install an HD Audio driver (or the equivalent) if you need any audio jacks or other more advanced audio capabilities.
- If the platform is Tiger Lake or later and RS5 OS is POR, please make sure Intel Display audio driver is installed. If not, there will no audio output even if an audio capable monitor is connected since RS5 OS only supports Intel DAC that is not POR on all platforms newer than Tiger Lake. Please refer to IOTG roadmap for platform and OS support.

§

## 13.0 Discrete Graphics (dGPUs)

On most Intel® processor family products, Intel® provides an integrated graphics (iGPU) capability. The iGPU provides a highly integrated solution that is power efficient and very capable. However, there are some applications that might need a higher power/performance graphics capability and that is where a discrete graphics (dGPU) might provide the needed capabilities. This usually comes at a higher cost, higher power and higher system thermal requirements than the integrated solution.

**Figure 103:** Example Discrete Graphics Cards



High-end gaming is one application that can benefit from a dGPU.

The following are the advantages of dGPU:

- May provide higher performance graphics capability.
- Can add additional simultaneous monitor handling (i.e. additional extended mode displays).
- Usually includes the RAM used by the GPU.
- Can be an add-in card or may be able to be included on the motherboard (motherboard down).

The following are the disadvantages of dGPU:

- Cost
- Power
- Heat
- Support provided by third party provider, not Intel®.
- Motherboard down solutions can be complex and take up board space.

External cards can be used as the exclusive graphics source or can be used cooperatively with the integrated graphics in a concurrent or hybrid/switchable mode. Details on concurrent and hybrid modes can be found in the following sections of this document.

## 13.1 Troubleshooting

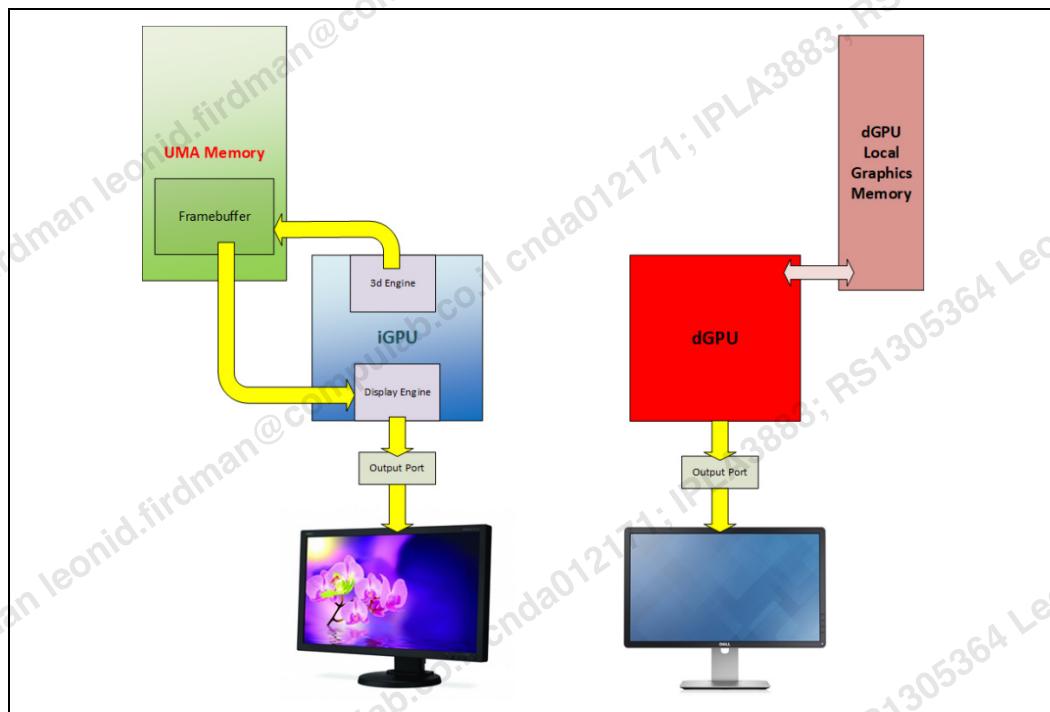
To troubleshoot issues using dGPUs:

- If you are trying to operate both the Intel® iGPU and a dGPU concurrently, then the dGPU firmware and driver must support the simultaneous operation. Intel® firmware (GOP) and drivers are designed for concurrent operation but not all third-party cards support this capability.
- The BIOS setup has options for determining if the dGPU or iGPU are the primary display. This setting is found in different places in different vendors BIOS. The default is to make an added dGPU as primary.
- Additional information provided in the "Concurrent Mode" and "Hybrid (Switchable)" sections.

## 13.2 Concurrent Mode

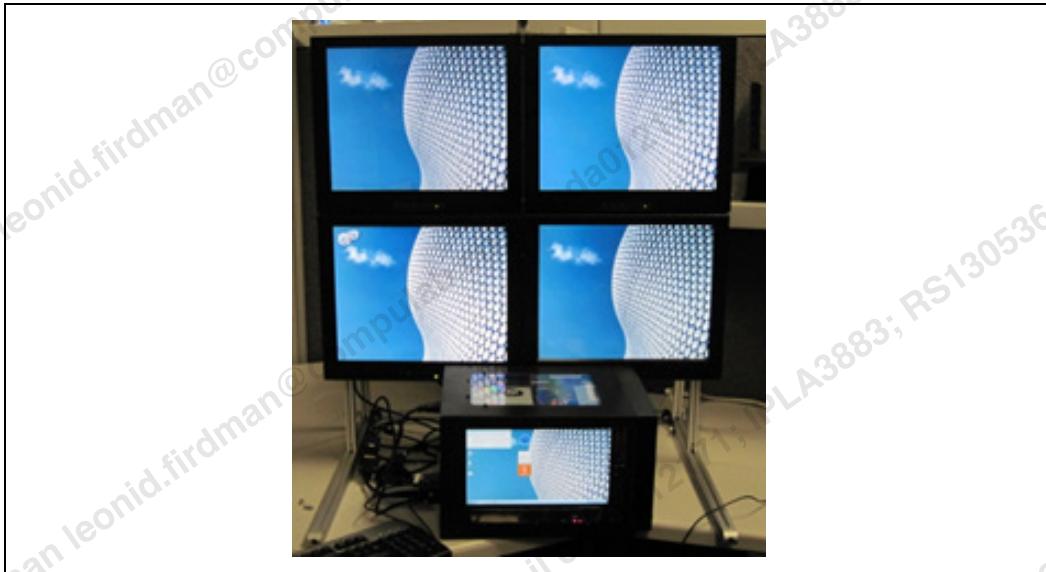
dGPU (i.e. add-in “external” graphics card) drives the displays connected to it at the same time (concurrently) as the iGPU (integrated graphics) drives the displays connected to the iGPU outputs. This requires the driver for the external card to support concurrent operation. There can be up to 4 displays connected to iGPU (with 4 display capable products) and 1 or more displays connected to dGPU depending on its capability.

**Figure 104: Concurrent iGPU/dGPU Diagram**



The following picture shows an example of concurrent operation with an eDP panel (bottom center) and two EFP displays running from the iGPU concurrent with dGPU running two displays.

**Figure 105:** Example Concurrent iGPU/dGPU



### 13.2.1 Concurrent Limitations

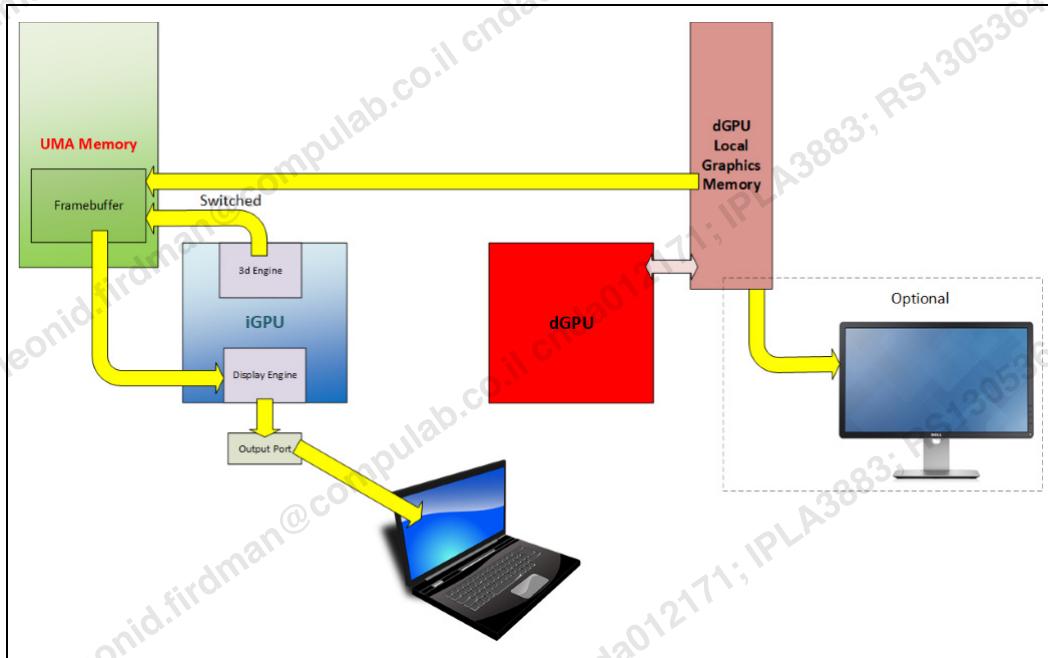
- The third-party card (firmware and driver) must be designed to support concurrent capability. There are dGPU products on the market that will not work in concurrent mode. Intel® is not able to fix that issue and you must get support from the third-party graphics vendor if you have issues.
- This is normally used in independent "extended" desktop mode for all the displays.
- You can use clone display only with displays all driven by the iGPU, or clone between displays driven by the dGPU. You cannot clone between iGPU and dGPU.

## 13.3

### Hybrid (Switchable) Mode

Hybrid (or Switchable-Hybrid) Technology was developed and added to platforms to allow switching between integrated (on-board/on-chip) graphics and a discrete graphics processor. This allows the user to dynamically switch between the enhanced power conservation of Intel® integrated graphics and the potentially higher performance capabilities of discrete graphics when needed.

Figure 106: Switchable/Hybrid iGPU/dGPU Diagram



### 13.3.1 Hybrid Requirements

Microsoft\* Hybrid Graphics has these requirements:

- The system contains a single integrated GPU and a single discrete GPU.
- It is a design assumption that the discrete GPU has a significantly higher performance than the integrated GPU.
- Both GPUs shall be physically enclosed as part of the system and available at boot.
  - MS Hybrid does not support hot-plugging of GPUs.
  - OEMS should seek further guidance from MS before designing systems with the concept of hot-plugging.
- With Microsoft's\* WDDM 2.0 or later in Windows\* 10, the discrete GPU can be utilized as a render-only device, with no displays connected to it, or can additionally drive a display while in hybrid mode. Render-only operation is NOT required. It must be noted that systems that have outputs available off of the discrete GPU will NOT be supported by previous versions of the OS (Windows\* 8.1 and Older).

The Microsoft\* Windows\* 10 operating system enables the Windows\* 10 Hybrid graphics framework so the GPUs and their drivers can be utilized simultaneously to provide users with the benefits of both performance capability of discrete GPU (dGPU) and low-power display capability of the processor integrated GPU (iGPU). For instance, when there is a high-end 3D gaming workload in progress, the dGPU will process and

render the game frames using its graphics performance, while the iGPU continues to perform the display operations by compositing the frames rendered by dGPU.

Each vendor [i.e. Intel®, 3rd Party Graphics Vendor, and Microsoft\*] is responsible for support of their own component, and Microsoft\* owns the overall implementation including all Hybrid Graphics related documentation and guidance.

Hybrid Graphics has requirements which determine support on different platforms and these requirements may be different between versions of OS. Refer to Microsoft\* documentation for detailed requirements per OS version.

### 13.3.2 Utilizing Hybrid

For Hybrid operation, the iGPU device is configured as the primary display adapter driving the embedded or local flat panel of the device. Note that if the dGPU is set to drive the internal display, it gets configured as the primary display adapter. In this scenario, and when there is no external display driven by iGPU, the system ceases to qualify as a hybrid graphics, and it rather behaves like any discrete graphics system where the capabilities of iGPU are not utilized.

Having a system configured where both iGPU and dGPU are driving displays needs to be carefully evaluated especially if there is difference in display capabilities of the two graphics adapters that would lead to user confusion or a bad user experience.

In Hybrid mode, rendering can be done either by dGPU or iGPU.

If the dGPU is used:

1. dGPU renders to its local memory which then is copied to the system UMA memory.
  - a. If a display is attached to the dGPU, the image is displayed there as well.
2. The frame buffer, in UMA memory, is displayed by the iGPU's Display Engine.

If the iGPU is used:

1. iGPU renders to the frame buffer (UMA memory) directly.
2. The frame buffer (UMA memory) is displayed by the iGPU Display Engine.

§

## 14.0 Hot-Plug Detect (HPD)

Hot-plug Detect is a hardware signal specific to the display pipe and port. It is used to indicate when a display is present (at boot), is disconnected after startup, or connected after startup. It tells the driver (and GOP firmware with support) that it needs to act upon the HPD event:

1. HPD is not used for LFPs. LFPs like eDP panels are assumed to always be connected by the firmware and driver.
2. At boot, if a monitor (EFP) is connected, HPD tells the firmware/driver that the port should be initialized and active, or left inactive if the HPD pin indicates there is no display.
3. Upon a display connect, it tells the driver (and some newer GOP firmware) that the port should go from inactive to active (assuming there are free pipes available to assign to the display).
4. Upon a display disconnect, it tells the driver/firmware to deactivate the port, free up the pipe, and shut off the display hardware to the port to save power. "Lock Display" may change this behavior.

Things to keep in mind for HPD:

- A display should not be rapidly disconnected and reconnected or the display and port can get out of sequence. Allow a few seconds after a disconnect before reconnecting to allow the port and display to settle. You can experience a black screen on the display if you reconnect the display too quickly.
- VBIOS generally do not support HPD during pre-boot.
- Newer GOP firmware (from Skylake onwards) should support HPD display connect and was supported on some earlier GOP. To test, set up the system to always boot into the UEFI Shell, reboot with the display disconnected, attach the display (it will be blank initially as anything that would have been displayed is lost because there was no display to send it to), then press the enter key on the keyboard to see if you see the UEFI Shell prompt appear. You may need to change the VBT setting as the default for GOP HPD is "disabled".

The following figure shows the DisCon page where HPD for GOP is normally enabled or disabled in the VBT.

**Figure 107: HPD in VBT**

- If you use the driver "Lock Display" feature available on most versions of the Intel® graphics driver, it ignores HPD and leaves the port active when a display is disconnected.
- HPD pin is specific to the DDI port and should not be interchanged on the hardware design.
- HPD on eDP is not supported even though the hardware has a pin assigned for it.

§

## 15.0 Redriver / Retimer

The frequency that a digital display interface runs at increases as the resolution of the display goes up. This in turn means the signal integrity of the display interface becomes increasingly difficult to maintain. Higher resolution drives higher display interface frequencies which mean keeping signal integrity becomes more important but also more difficult to maintain.

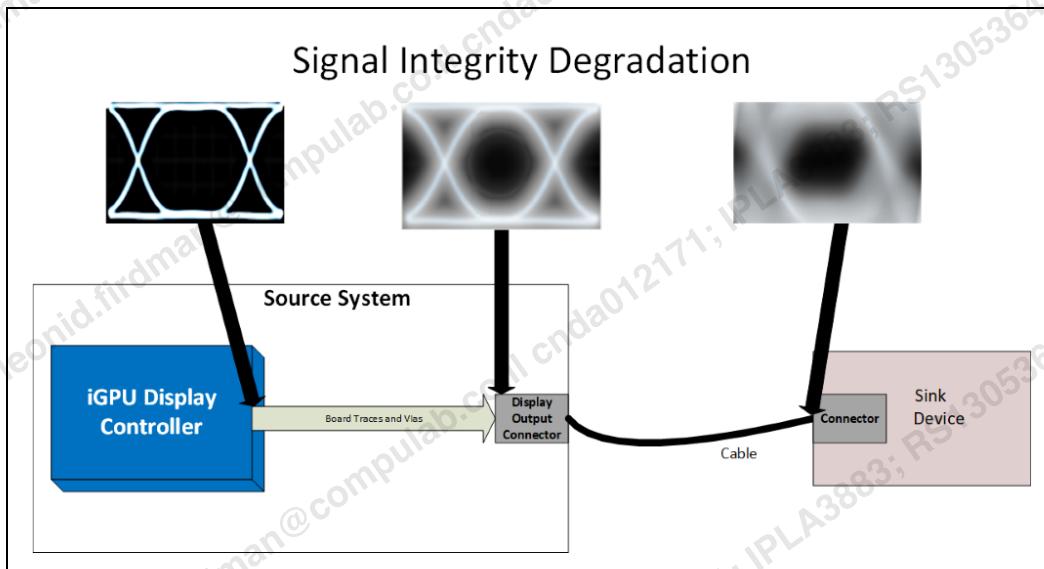
Many factors impact the display signal integrity including:

- Board design clock stability.
- Board design trace lengths of the display interface.
- Board signal routing (e.g. # of vias and adjacent signals).
- Additional connectors between the GPU and the display interface (e.g. in a docking station, or CPU daughter card to motherboard connector).
- The EMI environment the system is used in.
- Monitor cable lengths and cable quality.
- Monitor quality.

Decreased signal integrity can cause many display issues including, but not limited to:

- Display flicker
- Link training errors
- "Black screen" (no display)
- Decreased resolution capability

The following diagram illustrates the issue. The CPU outputs a clean display signal at the display transmitter that is degraded as traces carry it to the display output connector which then is degraded further by the monitor cable/connector and then further internally in the monitor as the signal is carried to the receiver.

**Figure 108: Video Signal Degradation**

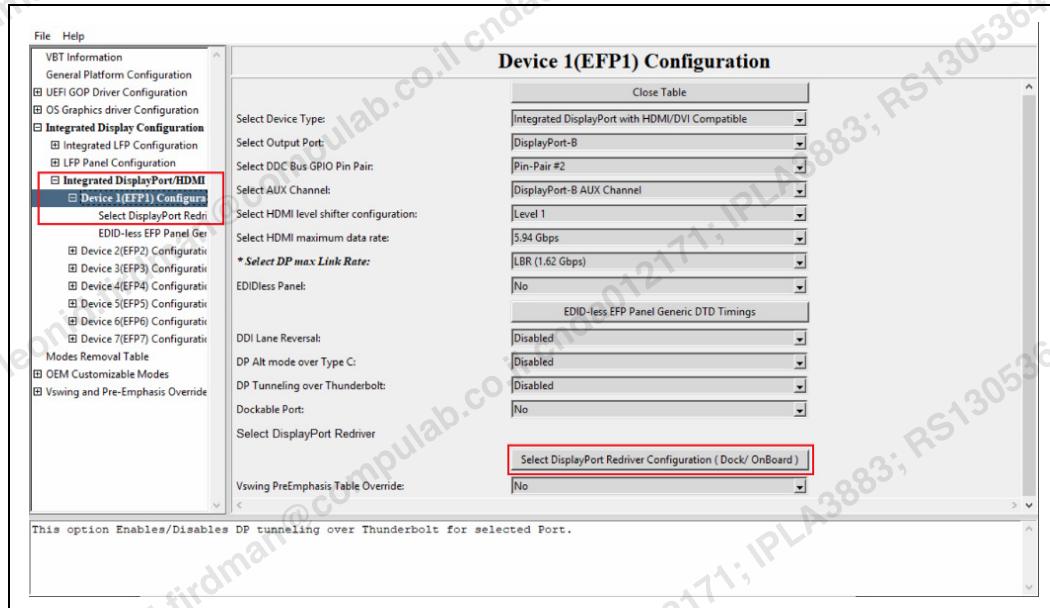
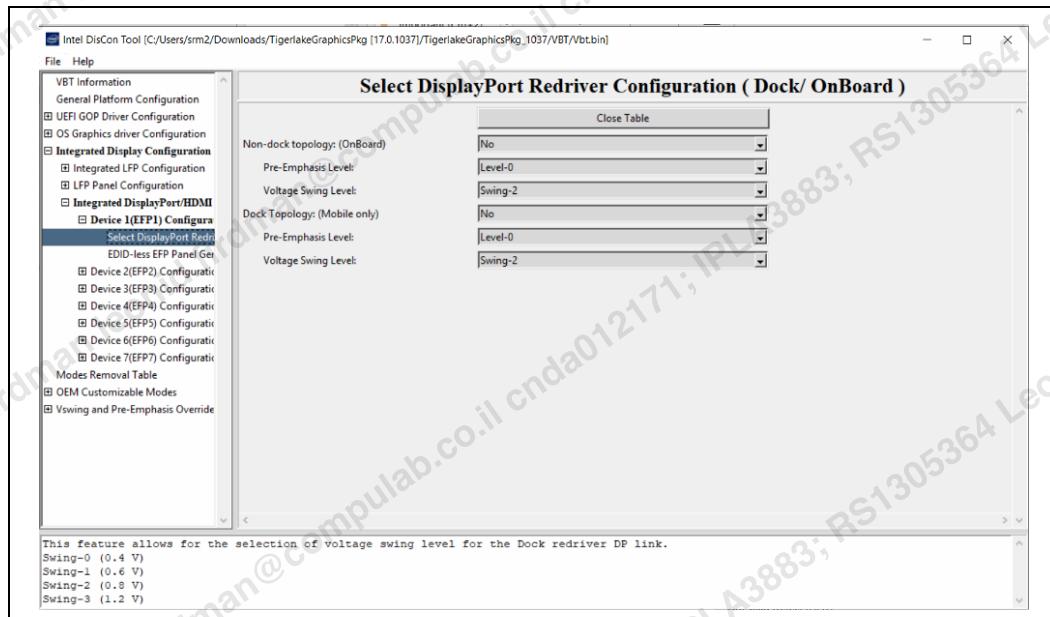
In these cases, the design may require either a “redriver” or “retimer” between the iGPU and connector(s) to solve display interface signal integrity issues within the system to the system display output. Keep in mind that this cannot always compensate for board layout issues, a bad/long display cables, or bad monitor design.

## 15.1 Redriver

A redriver is a specially designed amplifier that can be tuned to compensate for signal integrity issues. It is designed to give a specific boost to the display signals so they can be properly transmitted to the display. The specific type of redriver for display interfaces is known as a "linear redriver" and may be specific to the display interface type (e.g., DP or HDMI).

The tuning of the redriver settings is specific to the board design. Some redriver settings should be available in the Intel® VBT as set with the DisCon or BMP tools. The availability of the settings can vary between Intel® processor families, so refer to your specific firmware (GOP) and driver for details.

Following is an example of where to find redriver settings for DP in DisCon:

**Figure 109:** Redriver in DisCon

**Figure 110:** Redriver Settings


The settings are board specific that the board designer will need to determine for your board design and monitors used.

**Caution:** Too much amplification can cause distortion as bad as too little.

## 15.2 Retimer

A retimer is another solution that can improve display signal integrity. It works by taking in the display signals and internally resynthesizing and resyncing them. This strengthens and retimes the signals in order to restore the signal integrity. Keep in mind that a retimer has the tendency to skew the display clocking from the system clock which can impact situations that require synchronization to the system clock such as multi-CPU display genlock.

§

## Appendix A

### A.1

### Windows\* Registry for Graphics

In Windows\*, it is sometimes necessary to make manual tweaks of the graphics driver Registry settings. This might be necessary if your application has unusual requirements or if you are directed by Intel® Customer Support to make a custom entry or other adjustment. To facilitate this, we have some tips/guidelines to assist you:

**Caution:** Editing the Registry is a fairly advanced task, so modify only if you are an expert.

- Arbitrary changes to the Registry can end up with an unbootable system so backups of your data and of the Registry itself are good ideas before you attempt any changes.
- If you make a system unbootable, Intel® will not be able to help you recover it- you may end up needing to reinstall Windows\* to make the system usable again.
- Windows\* provides a tool called Registry Editor or "RegEdit" for editing the Registry.
- In some cases, there may be multiple graphics registry entries. This may be the case if you have installed the graphics driver more than once or have had Windows\* do a graphics driver update. You need to determine the active registry entry "Driver Key" for the graphics driver in order to make changes to the correct one. To determine the driver key, use the following procedure:
  1. Go to **Device Manager->Display Adapters**.
  2. Right click **Intel® graphics driver**-> Choose **properties**.
  3. Click **Details** tab.
  4. Choose **Driver Key** from the drop-down.

In this example case, the active graphics driver registry is:

"{4d36e968-e325-11ce-bfc1-08002be10318}\0001\"

which can be searched for in the Registry with RegEdit.

You may need to modify existing entries in the graphics driver Registry or add new ones to enable specific driver features.

#### A.1.1

#### Registry Batch

In the event you would like a batch file (.bat) to edit the registry in a programmatic way, the following script may give you some ideas on how to accomplish this task. This is not tested code, it is just a concept that you will need to adapt and debug. You assume the responsibility for deploying this capability.

The following example shows how to set a new value for the existing regkey DelayedDetectionForHDMI.

```
REM --- Start of MODIFY batch
@echo off

REM --- Example batch file to change a value that resides on the
Display Node in the Windows registry ---

echo.
echo IMPORTANT NOTE: This batch file MUST be run with
Administrator privilege!
echo.

:AdminTest
echo.
echo Testing Privileges...
echo.

NET SESSION >nul 2>&1
IF %ERRORLEVEL% EQU 0 (
    @echo Administrator Privileges Detected!
) ELSE (
    @echo This Batch File Session does NOT have Administrator
Privilege!
    echo.
    @echo Exiting program...
    GOTO end
)

REM --- Example batch file to change a value that resides on the
Display Node in the Windows registry ---

echo.

@setlocal enableextensions enabledelayedexpansion

REM --- This will set our key path right up to the point of where
it can diverge.
set
DispDriverRegKey=HKLM\SYSTEM\CurrentControlSet\Control\Class\{4D3
6E968-E325-11CE-BFC1-08002BE10318}

REM --- Note that "pci\ven_8086&dev_" below represents any kind
of Intel Display hardware. Intel = "8086" Nvidia = "10de" ---
set Operation=Reg QUERY %DispDriverRegKey%/s /f
"pci\ven_8086&dev_"

REM --- Create a string variable to hold a return value from the
command operation
set newRegKey=
set DetectDelayValue=1

if "%~1" == "" GOTO noParam
```

```

@echo Setting Delay detection value to %1
set DetectDelayValue=%1

echo.

:noParam

if "%~1" == "" (
    echo No command parameter, defaulting to use 100 for
Delay Detect Value.
)
echo.

REM --- Run the command "Reg QUERY" with display driver key
parameters to locate a Display Node matching Intel's hardware ---
REM --- Driver settings such as "CUINotRequired" reside on that
node.
REM --- Display Nodes or Instances appear as four-digit numbers
such as 0000, 0001, 0002, 0003 and so on ---
REM --- For Example, Nvidia HW could be on node 0000, ATI HW on
node 0001, and Intel HW on node 0002 and so on ---
REM --- After this operation below we are expecting to see \000?
appended to DispDriverRegKey declared above ---

FOR /F "usebackq delims==&" %%A IN (`%Operation%`) DO ( set
newRegKey=!newRegKey!%%A )

REM --- If we find "0 match" in the newRegKey string variable
then that means failure to find Intel HW ---

if "%newRegKey%"=="%newRegKey:0 match=%" (
    REM --- Intel HW match found ---
    For /f "tokens=1,2 delims= " %%a in ("%newRegKey%") do (
        set BEFORE_UNDERSCORE=%%a
        set AFTER_UNDERSCORE=%%b
    )
) else (
    REM --- Intel HW match found ---
    echo No Intel display hardware installed!
    goto end
)

REM --- The "BEFORE_UNDERSCORE" variable now has the correct
matching \0000 or \0001 or \0002 -etc appended to
DispDriverRegKey ---

REM --- This command sets the value "DelayedDetectionForHDMI " to
100 which means to reduce the delay to work around certain
monitor issues ---
reg ADD %BEFORE_UNDERSCORE% /v DelayedDetectionForHDMI /t
REG_DWORD /d %DetectDelayValue% /f

echo.

```

```
@echo %BEFORE_UNDERSCORE%
echo.

:end
endlocal

echo.

The following example shows how to add a new key to the active Intel® graphics
registry:

REM --- Start of ADD batch
@echo off

REM --- Example batch file to change a value that resides on the
Display Node in the Windows registry ---

cls

echo.

echo IMPORTANT NOTE: This batch file must be run with
Administrator privilege!
echo.

@setlocal enableextensions enabledelayedexpansion

REM --- This will set our key path right up to the point of where
it can diverge.
set
DispDriverRegKey=HKLM\SYSTEM\CurrentControlSet\Control\Class\{4D3
6E968-E325-11CE-BFC1-08002BE10318}

REM --- Note that "pci\ven_8086&dev_" below represents any kind
of Intel Display hardware. Intel = "8086" Nvidia = "10de" ---
set Operation=Reg QUERY %DispDriverRegKey%/s /f
"pci\ven_8086&dev_"

REM --- Create a string variable to hold a return value from the
command operation
set newRegKey=
set EnableFromSpecificBlock=1

if "%~1" == "" GOTO noParam
    set /a block=%1-1
    @echo Setting bitBashing to begin at EDID block %block%
    set EnableFromSpecificBlock=%1

:noParam

if "%~1" == "" @echo No command parameter was given, defaulting
to 1( Use bitBashing for ALL blocks)
echo.
```

```

REM --- Run the command "Reg QUERY" with display driver key
parameters to locate a Display Node matching Intel's hardware ---
REM --- Driver settings such as "CUINotRequired" reside on that
node.
REM --- Display Nodes or Instances appear as four-digit numbers
such as 0000, 0001, 0002, 0003 and so on ---
REM --- For Example, Nvidia HW could be on node 0000, ATI HW on
node 0001, and Intel HW on node 0002 and so on ---
REM --- After this operation below we are expecting to see \000?
appended to DispDriverRegKey declared above ---

FOR /F "usebackq delims==&" %%A IN (`%Operation%`) DO ( set
newRegKey=!newRegKey!%%A )

REM --- If we find "0 match" in the newRegKey string variable
then that means failure to find Intel HW ---

if "%newRegKey%"=="%newRegKey:0 match=%" (
    REM --- Intel HW match found ---
    For /f "tokens=1,2 delims= " %%a in ("%newRegKey%") do (
        set BEFORE_UNDERSCORE=%%a
        set AFTER_UNDERSCORE=%%b
    )
) else (
    REM --- Intel HW match not found ---
    echo No Intel display hardware installed!
    goto end
)

REM --- The "BEFORE_UNDERSCORE" variable now has the correct
matching \0000 or \0001 or \0002 -etc appended to
DispDriverRegKey ---

REM --- This command sets the value
"BitBashingEnableFromSpecificBlock_WA" to EnableFromSpecificBlock
which means Enable bitBashing at block 0-?... ---
reg ADD %BEFORE_UNDERSCORE% /v
BitBashingEnableFromSpecificBlock_WA /t REG_DWORD /d
%EnableFromSpecificBlock% /f

echo.
@echo %BEFORE_UNDERSCORE%
echo.

:end
Endlocal

```

§