Intel® Embedded Media and Graphics Driver v36.16.4 for Windows Embedded Compact 7* and v36.18.4 for Windows Embedded Compact 2013* for Intel® Atom™ Processor E3800 Product Family / Intel® Celeron® Processor N2807 / N2930 / J1900

User Guide

January 2016

Contents

Figures

Tables

Revision History

This document may have been updated since the release shown below. See http://edc.intel.com/Software/Downloads/ for the most recent version.

Date	Revision	Description
January 2016	006	Intel® Embedded Media and Graphics Driver v36.16.4 for Windows Embedded Compact 7* and v36.18.4 for Windows Embedded Compact 2013* for Intel® Atom™ Processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900
July 2015	005	Intel® Embedded Media and Graphics Driver v36.16.3 for Windows Embedded Compact 7* and v36.18.3 for Windows Embedded Compact 2013* for Intel® Atom™ Processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900
November 2014	004	Intel® Embedded Media and Graphics Driver v36.16.2 for Windows Embedded Compact 7* and v36.18.2 for Windows Embedded Compact 2013* for Intel® Atom™ Processor E3800 Product Family/ Intel® Celeron® Processor N2807/N2930/J1900 Release
August 2014	003	Intel® Embedded Media and Graphics Driver v36.16.1 for Intel® Atom™ Processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900 Windows Embedded Compact 7* Release
June 2014	002	Intel® Embedded Media and Graphics Driver v36.16.0 for Intel® Atom™ Processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900 Windows* Embedded Compact 7 Windows Embedded Compact 7*WEC7* Release
May 2014	001	Initial Release

§ §

1.0 Introduction

The Intel® Embedded Media and Graphics Driver (Intel® EMGD) comprises a suite of multi-platform graphics drivers designed to meet the requirements of embedded applications. Featuring Intel® Dynamic Display Configuration Technology (DDCT), the drivers run on the following Embedded Intel® Architecture (eIA) chipsets:

- Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900 (Windows Embedded Compact 7* and Windows Embedded Compact 2013*).

Note: If you need support for a chipset that is not listed above but is in the same family as those listed, please contact your Intel representative.

The Intel® EMGD supports the following types of display devices:

- Analog CRT
- HDMI
- DisplayPort
- Embedded DisplayPort (eDP)

Intel® EMGD is designed to work with fixed-function systems, such as Point-of-Sale (POS) devices, ATMs, gaming devices, In-vehicle Information/Entertainment systems, etc. It can be configured to work with various hardware and software systems and supports both Microsoft Windows Embedded Compact 7* and Windows embedded Compact 2013*operating system.

Note: Refer to product release notes for more details on Intel® EMGD supported features.

1.1 Purpose

This manual provides information on both firmware and software, providing hardware design considerations, installation requirements, and static configuration options.

1.2 Intended Audience

This document is targeted at all platform and system developers who need to interface with the graphics subsystem. This includes, but is not limited to: platform designers, system BIOS developers, system integrators, original equipment manufacturers, system control application developers, as well as end users.

1.3 Related Documents

The following documents provide additional information that may be useful when using the Intel® Embedded Media and Graphics Driver. Additional resources are available at http://edc.intel.com/Software/Downloads/EMGD/.

- *Intel® Graphics Media Accelerator - Binary Modification Program (BMP) User Guide* (Document #368119) Revision 3.0 or above.

This document provides information on configuration of Video BIOS.

- *Intel® Embedded Media and Graphics Driver – Binary Modification Program (BMP) User Guide Addendum for Windows* Embedded Compact* (Document #548367)

This document provides information on configuration of Video BIOS for Windows* Embedded Compact

- *Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API-Reference Manual*

This document Provides Information on configuration of Intel® EMGD DirectShow Codecs for Windows* Embedded Compact 7* and Windows embedded Compact 2013*

- *Intel® EMGD for Windows Embedded Compact 7** and Windows embedded Compact 2013**DirectShow Codecs with Intel® Atom™ Processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900 Video Acceleration Users Guide*

1.4 Conventions

The following conventions are used throughout this document.

Boldface	Represents text that you type and text that appears on a screen.
Italics	Introduces new terms and titles of documents.
`Courier New`	Identifies the names of files, executable program names, and text that appears in a file.
Angle Brackets (<>)	Encloses variable values in syntax or value ranges that you must replace with actual values.
Vertical Bar (|)	Used to separate choices (for example, TRUE | FALSE)

1.5 Acronyms and Terminology

The table below lists the acronyms and terminology used throughout this document.

Table 1. Acronyms and Terminology (Sheet 1 of 4)

Term	Description
ADD Card	AGP Digital Display. An adapter card that can be inserted into the PCIe* x16 port of Intel chipset family-based systems. ADD cards allow configurations for TV-out, LVDS, and TMDS output (i.e., televisions, digital displays, and flat panel displays).
AIM	Add In Module.
API	Application Programming Interface.
BDA	BIOS Data Area. A storage area that contains information about the current state of a display, including mode number, number of columns, cursor position, and so on.

Table 1. Acronyms and Terminology (Sheet 2 of 4)

Term	Description
BIOS	Basic Input/Output System. The Intel® Embedded Media and Graphics Driver interacts with two BIOS systems: system BIOS and Video BIOS (VBIOS). VBIOS is a component of the system BIOS.
BLDK	Boot Loader Development Kit.
BMP	BIOS Modification Program; allows customizing the data in VBT.
Clone Display Configuration	A type of display configuration that drives two display devices, each displaying the same content, but can have different resolutions and (independent) timings. Compare DIH Display Configuration.
Contrast Ratio	Contrast ratio is the measure of the difference between light and dark on a display. If the contrast is increased, the difference between light and dark is increased. So something white will be very bright and something black will be very dark. Brightness and Contrast Controls differ in function between CRTs and LCDs.
COPP	Certified Output Protection Protocol* is a Microsoft-defined API to provide application with information about what output protection options are available on a system.
D3D	Microsoft Direct3D*. A3D graphics API as a component of DirectX* technology.
DC	Display Configuration.
DDCT	Intel® Dynamic Display Configuration Technology.
DirectDraw*	A component of the DirectX* Graphics API in Microsoft Windows* OS.
DIH Display Configuration	Dual Independent Head. A type of display configuration that supports two displays with different content on each display device. The Intel® Embedded Media and Graphics Driver supports Extended mode for Microsoft Windows systems and Xinerama for Linux* systems.
DP	Display Port
DTD	Detailed Timing Descriptor. A set of timing values used for EDID-less devices.
DVI	Digital Video Interface.
DVO	Digital Video Output.
EBDA	Extended BIOS Data Area. An interface that allows the system BIOS and Option ROMs to request access to additional memory.
EDID	Extended Display Identification Data. A VESA* standard that allows the display device to send identification and capabilities information to the Intel® Embedded Media and Graphics Driver. Intel® EMGD reads all EDID data, including resolution and timing data, from the display, thus negating the need for configuring DTD data for the device.
EDID-less	A display that does not have the capability to send identification and timing information to the driver and requires DTD information to be defined in the driver.
eDP	Embedded Display Port
EFI	Extensible Firmware Interface.
eIA	Embedded Intel® Architecture.
EMI	Electromagnetic Interference.
EPOG	Embedded Pre-OS Graphics feature.
Extended Clone Mode	A feature that allows you to have different sized displays in Clone mode.
Framebuffer	A region of physical memory used to store and render graphics to a display.
GDI	Graphics Device Interface. A low-level API used with Microsoft Windows operating systems.

Table 1. Acronyms and Terminology (Sheet 3 of 4)

Term	Description
GMA	Intel Graphics Media Accelerator. Refers to both the graphic hardware in Intel chipsets as well as the desktop/mobile driver. The GMA driver is not intended for use in embedded applications.
GMS	Graphics Mode Select (stolen memory).
HAL	Hardware Abstraction Layer. An API that allows access to the Intel® chipsets.
HDCP	High-bandwidth Digital-Content Protection. A specification that uses the DVI interface. HDCP encrypts the transmission of digital content between the video source (transmitter) and the digital display (receiver).
HDMI	High-Definition Multimedia Interface, an uncompressed, all-digital, audio/video interface.
IAL	Interface Abstraction Layer. An API that allows access to graphics interfaces including the GDI, and DirectDraw*.
iDCT	Inverse Discrete Cosine Transformation (hardware feature).
LPCM	Linear Pulse Code Modulation (LPCM). A method of encoding audio information digitally. The term also refers collectively to formats using this method of encoding.
LVDS	Low Voltage Differential Signaling. Used with flat panel displays, such as a laptop computer display.
NTSC	National Television Standards Committee. An analog TV standard used primarily in North and Central America, Japan, the Philippines, South Korea, and Taiwan. Its resolutions are based on 525-line systems. Compare PAL.
OAL	Operating system Abstraction Layer. An API that provides access to operating systems, including Microsoft Windows* and Linux*.
Option ROM (OROM)	Code that is integrated with the system BIOS and resides on a flash chip on the motherboard. The Intel Embedded Video BIOS is an example of an option ROM.
OS	Operating System.
PAL	Phase Alternating Lines. An analog TV standard used in Europe, South America, Africa, and Australia. Its resolutions are based on 625-line systems. Compare NTSC.
PCF	Parameters Configuration File.
PCI	Peripheral Component Interface.
Port Driver	A driver used with the sDVO interfaces of the System Controller Hub (SCH).
POST	Power On Self Test.
PWM	Pulse Width Modulation.
Reserved Memory	A region of physical memory in a Windows Embedded Compact 7 system set aside for BIOS, VBIOS, and graphics driver operations. Reserved memory can be configured for use by the operating system and other applications when not in use by the BIOS.
Saturation	Monitors and scanners are based on the “additive” color system using RGB, starting with black and then adding Red, Green, and Blue to achieve color. Saturation is the colorfulness of an area judged in proportion to its brightness. Full saturation of RGB gives the perception of white, and images are created that radiate varying amounts of RGB, or varying saturation of RGB.
SCART	French Acronym - Syndicat des Constructeurs d'Appareils Radiorecepterus et Televiseurs. A video interface possessing up to four analog signals (Red/Green/Blue/Composite PAL). S-Video (Luma/Chroma) is possible over the SCART interface as well.
SCH	System Controller Hub.

Table 1. Acronyms and Terminology (Sheet 4 of 4)

Term	Description
SCS	Software Compliance Statement.
sDVO	Serial Digital Video Output.
Single Display Configuration	A type of display configuration that supports one and only one display device.
SSC	Spread Spectrum Clock.
Stolen Memory	A region of physical memory (RAM) set aside by the system BIOS for input and output operations. The amount of stolen memory is configurable. Stolen memory is not accessible to the operating system or applications.
System BIOS	The standard BIOS used for basic input and output operations on PCs.
TMDS	Transitioned Minimized Differential Signaling. Used with DVI displays, such as plasma TVs.
TOM	Top Of Memory.
TSR	Terminate and Stay Resident. A program that is loaded and executes in RAM, but when it terminates, the program stays resident in memory and can be executed again immediately without being reloaded into memory.
VBIOS	Video Basic Input Output System. A component of system BIOS that drives graphics input and output.
VESA*	Video Electronics Standards Organization.
VEXT Display Configuration	Vertical Extended. A type of display configuration that enables both Primary and secondary displays. Primary and secondary displays can be configured with separate timings. The resolution for the secondary display must be the same as the primary. Content comes from a single framebuffer that spans both displays oriented vertically.
VGA	Video Graphics Array. A graphics display standard developed by IBM* that uses analog signals rather than digital signals.
VLD	Variable Length Decoding.
VMR	Video Mixing Render.
WHQL	Windows* Hardware Quality Labs. WHQL is a testing organization responsible for certifying the quality of Windows drivers and hardware that runs on Windows operating systems.
YUV	Informal, but imprecise reference to the video image format, Y'CbCr. The Y' component is luma, a nonlinear video quality derived from RGB data denoted without color. The chroma components, Cb and Cr, correspond nonlinearly with U and V as differences between the blue and luma, and between the red and luma, respectively.

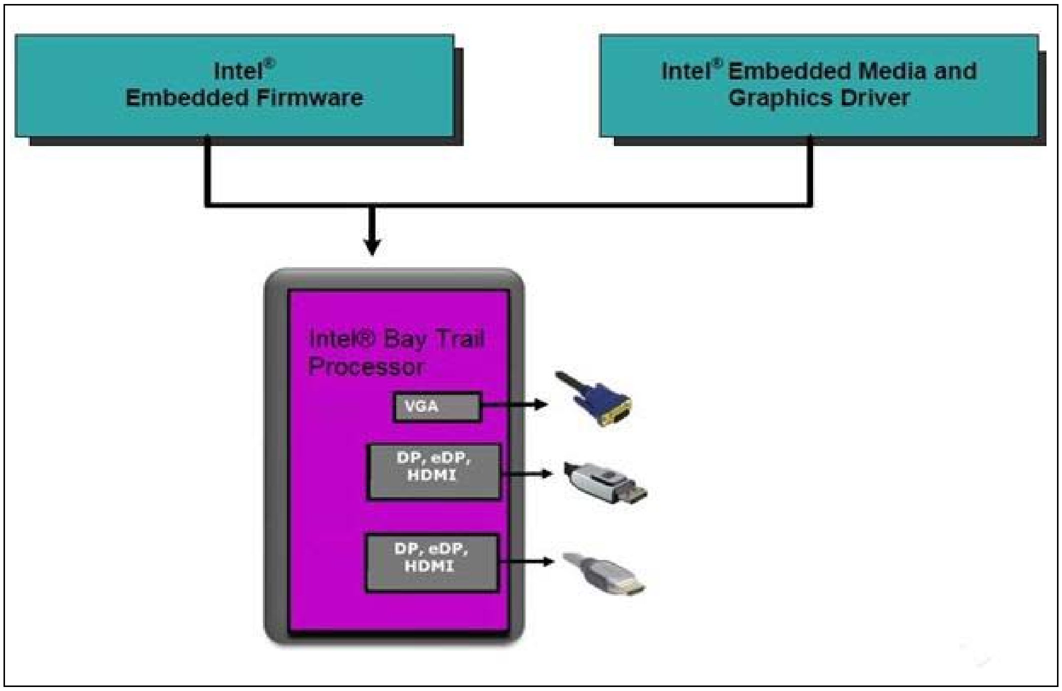
§ §

2.0 Architectural Overview

2.1 Introduction

The Intel® Embedded Media and Graphics Driver is composed of a runtime graphics driver and a video firmware component. (See the illustrations below.) Both the driver and video firmware (for instance, VBIOS, GOP, and EPOG) control the processor to perform display and render operations. The video firmware is predominantly leveraged by System BIOS during system boot but is also used at runtime by the driver to handle full-screen text mode on the operating systems.

Figure 1. Intel® Embedded Media and Graphics Driver


Intel®
Embedded Firmware
Intel® Embedded Media and Graphics Driver
Intel® Bay Trail Processor
VGA
DP, eDP, HDMI
DP, eDP, HDMI

2.1.1 Display Options

The following section describes the types of displays and configurations supported by the Intel® Embedded Media and Graphics Driver.

2.1.1.1 Types of Displays

The table below lists the types of displays supported by the Intel® Embedded Media and Graphics Driver.

Table 2. Types of Displays Supported

Display	Description
CRT	Analog CRT, supported with an external transmitter via an sDVO port.
HDMI	High-Definition Multimedia Interface
DP	Display Port
eDP	embedded Display Port

2.1.1.2 Display Configuration

Intel® EMGD supports driving two displays simultaneously. Several configurations are supported, dependent on operating system and chipset. The various display configurations are described in the table below.

Table 3. Display Configuration Definitions

Display Configuration Mode	Description
Single	Normal desktop configuration, single monitor
Clone‡	Two displays, same content, different resolutions, independent timings
Extended*	Two displays, continuous content (available in Windows* only)
Vertical Extended	Two displays, continuous content vertically
‡ Supported display depends on driver and hardware availability. See the RelNotes.txt for more information.	

Table 4 below summarizes which display configurations are supported by Intel chipsets.

Table 4. Supported Display Configurations

Chipset	Operating System
	Windows Embedded Compact 7* / Windows Embedded Compact 2013*
Intel® Atom™ Processor E3800 Product Family/ Intel® Celeron® Processor N2807/N2930/J1900	Single, Clone, Extended, Vertical Extended

2.2 Features

The following sections describe major features Intel® EMGD supports.

2.2.1 Processors Supported

The table below lists Intel® EMGD-supported processors.

Table 5. Processors Supported by the Intel® EMGD v36.16.4 for Windows Embedded Compact 7* and v36.18.4 for Windows Embedded Compact 2013*

Chipset	Intel® EMGD VBIOS Support	Intel® EMGD Support
Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900	Yes	Yes

All supported processors provide output on digital monitors. CRTs and TVs are supported through integrated display ports such as Display Port (DP), embedded Display Port (eDP), HDMI, etc. interfaces, depending on hardware availability.

2.2.2 OS and API Support

Intel® EMGD supports the following operating systems and driver variants:

- Microsoft Windows Embedded Compact 7*
- Microsoft Windows Embedded Compact 2013*

2.2.3 DisplayID Support

The Intel® Embedded Media and Graphics Driver supports the DisplayID specification. DisplayID is a VESA* specification (www.vesa.org) that describes the data format for the display configuration parameters and provides the capability to unify the display data structure thereby decreasing the need to rely on proprietary extensions. For more information on DisplayID, its uses and parameters please reference the VESA* specification (www.vesa.org).

2.2.4 EDID-Less Configuration

EDID-less support is the ability to run a display panel that does not have display timing information within the panel. Therefore, the user has to provide the display timing information to the graphics drivers during configuration using BMP.

This document describes only the necessary edits to the configuration files that are required to implement the graphics driver and VBIOS, and not specific settings for EDID-less panel configuration. Please refer to the manufacturer's specifications for the DTD settings to use for your EDID-less panels.

2.2.4.1 EDID-Less Panel Type Detection

The Intel® Embedded Media and Graphics Driver supports EDID-less displays that do not export timing modes. This is accomplished by allowing configuration of a Detailed Timing Descriptor (DTD), and associating that DTD with a specific display port.

2.2.5 Rotation

Rotation is the ability to rotate the display for the Intel® Embedded Media and Graphics Driver. Rotation support includes 0°, 90°, 180°, 270°. Rotation is supported only on the following chipsets using Windows* Embedded Compact operating system:

- Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900

Note: Rotation is not supported with the VBIOS. Rotation is supported with both Windows Embedded Compact 7* and Windows Embedded Compact 2013* but only in static mode.

§ §

3.0 Platform Configuration Using Binary Modification Program (BMP)

Intel® Binary Modification Program (BMP) is an application that allows you to modify the data in Video BIOS Table (VBT). The VBT is a block of customizable platform-specific data. It holds platform-specific information used by the Video BIOS and device drivers such as Flat Panel Timings, OEM definable Mode Timing, GPIO pins, Clock, and more. The data in the VBT is customized using the BMP utility or simply editing the source code and rebuilding.

3.1 Configuring VBT Using BMP

1. Install the latest version of Intel® BMP on your host to be able to launch BMP. Download the latest version Intel® BMP and the latest version of Video BIOS files (.bsf and .dat) from https://platformsw.intel.com. If you do not have access, contact your Intel Representative.
2. Intel® BMP is designed for ease of use and configuration of Intel® EMGD VBIOS. Each configuration page has online help available and each data field is validated. Refer to *Intel® HD Graphics – Binary Modification Program (BMP) User Guide* (Document #368119) *and Intel® Embedded Media and Graphics Driver – Binary Modification Program (BMP) User Guide Addendum for Windows* Embedded Compact* (Document #548367) for more details on how to use BMP for VBT customization.

§ §

4.0 Video Firmware

4.1 Overview

The Intel® Embedded Video firmware (VBIOS) incorporates many of the features and capabilities of the Intel® Embedded Media and Graphics Driver. Intel® EMGD supports three variants of video firmware for different working environments, where one pairs GOP with SBIOS (UEFI), EPOG with OTM and VBIOS with Legacy SBIOS. Windows Embedded Compact 7* and Windows Embedded Compact 2013* requires a VBIOS to boot properly so you must be sure your firmware on your platform has a VBIOS. The video firmware includes support for the following processor:

- Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900

Intel VBIOS uses the BMP tool for configuration. For more detail about VBIOS configuration, refer to the BMP document (for GOP and VBIOS as detailed in Section 1.3, “Related Documents” on page 7).

4.2 System Requirements

The target system must contain one of the following Intel processors:

- Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900

•The target system must contain a minimum of 64 MB of RAM.

Note: If you will be making VBIOS configuration changes, you will need access to your BIOS vendor's merge tool and flash utility as well as have the support files to make “VBT” (Video Bios Table) changes. Changes are merged into your BIOS and VBIOS images then flashed back onto the platform. Contact your BIOS vendor for details.

§ §

5.0 Configuring Intel® EMGD for WEC 7*and WEC 2013*

5.1 Microsoft Windows Embedded Compact 7* Installation

The following sections describe how to install Intel® EMGD on the Microsoft Windows Embedded Compact 7* operating system.

5.1.1 Prerequisites

The development system should have the following software installed:

- Visual Studio* 2008 and Visual Studio Professional Service Pack 1
- Windows Embedded Compact 7* Platform Builder
- Latest Board Support Package (BSP) (see Section 5.1.2.1, "Installation and Setup" on page 17 for download location) or contact your Intel representative for the latest BSP.

The target system must contain one of the following Intel processors:

- Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900

Notes: For proper driver operation you must:

1. Replace the default VBIOS with the latest EMGD VBIOS.
2. Install the latest Intel® EMGD Windows Embedded Compact 7* software package.

5.1.2 Platform Builder Requirements

You must configure your Platform Builder parameters specific to the options that the system and image requires, for example, options for the operating system. A Board Support Package (BSP) is also required.

Note: Configuration steps for the BSP are beyond the scope of this procedure. An Intel® BSP must be used with Platform Builder to support platform based on the Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900.

5.1.2.1 Installation and Setup

Note: The installation sequences are crucial for compilation success.

1. Install Visual Studio* 2008 Professional.
2. Install Visual Studio* 2008 Professional Service Pack 1.
3. Install Windows Embedded Compact 7* platform builder.

   In the SETUP dialog during installation, select **x86** in the processor architecture section.
4. Install Board Support Package (BSP). Download the BSP from Adeneo* at:

   http://www.adeneo-embedded.com/Products/Board-Support-Packages/Intel

   or Bsquare* at:
   http://www.bsquare.com/software-downloads.aspx

Note: You need to register before you are able to download.

5.1.2.2 Configuring INTEL_CS.bat, platform.reg and platform.bib to include Intel® EMGD to the WEC7* Image

1. From the C:\WinCE700\platform\INTEL_CS folder, open file **INTEL_CS.bat**. Find the “`@REM set IMGRAM256=`” and change it to “`set IMGRAM256=0`”.
2. From the C:\WinCE700\platform\INTEL_CS\FILES folder, open the file **platform.reg.** Find the line **ENDIF BSP_DISPLAY_RAGEXL**, Paste the following codes between **ENDIF BSP_DISPLAY_RAGEXL and ENDIF BSP_NODISPLAY!**

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\PCI\Template\EMGD]
       "Dll"="isr_emgd.dll"
              "Class"=dword:03
       "SubClass"=dword:00
       "ProgIF"=dword:00
       "VendorID"=multi_sz: "8086","8086","8086","8086","8086"
       "DeviceID"=multi_sz: "0f31","0412","0406","0416","0426"
       "Prefix"="IGD"
       "IsrDll"="isr_emgd.dll"
       "IsrHandler"="isr_handler"

[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
       "Display"="ddi_emgd.dll"
[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
       "D3DMOverride"="ddi_emgd.dll"
[HKEY_LOCAL_MACHINE\System\D3DM\Drivers]
       "RemoteHook"="ddi_emgd.dll"

[HKEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]
"Candidate3"="Drivers\\Display\\Intel"
;#include "C:\WINCE700\OSDesigns\OSDesign1\OSDesign1\RelDir\
INTEL_CS_x86_Release\emgd_filters.reg"

;[HKEY_LOCAL_MACHINE\System\GDI\Monitors]
     ;"Total Monitors"=dword:2
```

Note: In order to configure Display in Extended mode, you need to uncomment the above two lines (Remove ";").

To include Intel® EMGD DirectShow codecs are provided in the Windows Embedded Compact 7* OS image, you need to comment out the below line (Remove ";").

```
;#include “C:\WINCE700\OSDesigns\OSDesign1\OSDesign1\
RelDir\INTEL_CS_x86_Release\emgd_filters.reg"
```

3. From the C:\WinCE700\platform\INTEL_CS\FILES folder, open the file platform.bib.
4. Add the following codes at the end of the file.

```
ddi_emgd.dll        $(_FLATRELEASEDIR)\ddi_emgd.dll         NK SHK
dp.dll              $(_FLATRELEASEDIR)\dp.dll               NK SHK
analog.dll          $(_FLATRELEASEDIR)\analog.dll           NK SHK
hdmi.dll            $(_FLATRELEASEDIR)\hdmi.dll             NK SHK
isr_emgd.dll        $(_FLATRELEASEDIR)\isr_emgd.dll         NK SHK
libGLESv2.dll       $(_FLATRELEASEDIR)\libGLESv2.dll        NK SH
libegl.dll          $(_FLATRELEASEDIR)\libegl.dll           NK SH
```

Here `$(_FLATRELEASEDIR)` Refer to the path: C:\WINCE700\OSDesigns\OSDesign1\OSDesign1\RelDir\INTEL_CS_x86_Release

Download the latest Intel® EMGD Driver Package from the Intel EDC Website or Contact your Intel representative. Unzip and copy the contents of **[Driver] folder** from Intel® EMGD driver package to **`[INTEL_CS_x86_Release]`**folder.

Intel® EMGD supports hardware accelerated video decode on Windows Embedded Compact 7*, the Intel® EMGD Windows Embedded Compact 7* Microsoft DirectShow* filters are provided in the form of middleware codec libraries (DLLs) that will interface with the Intel® EMGD Windows Embedded Compact 7* driver to operate. For installation and setup refer Intel® EMGD for Windows Embedded Compact 7* Microsoft DirectShow* Codecs with Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900 Video Acceleration Users Guide.

5.2 Microsoft Windows Embedded Compact 2013* Installation

The following sections describe how to install Intel® EMGD on the Microsoft Windows Embedded Compact 2013* operating system.

5.2.1 Prerequisites

The development system should have the following software installed:

- Visual Studio* Professional 2012 update 2
- Application builder for Visual Studio* 2012
- Windows Embedded Compact 2013* Platform Builder
- Latest Board Support Package (BSP) (see Section 5.2.2.1, “Installation and Setup” for download location) or contact your Intel representative for the latest BSP.

The target system must contain one of the following Intel processors:

- .Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900

Note: For proper driver operation you must:

1. Replace the default VBIOS with the latest EMGD VBIOS.
2. Install the latest Intel® EMGD Windows Embedded Compact 2013* software package.

5.2.2 Platform Builder Requirements

You must configure your Platform Builder parameters specific to the options that the system and image requires, for example, options for the operating system. A Board Support Package (BSP) is also required.

Note: Configuration steps for the BSP are beyond the scope of this procedure. An Intel® BSP must be used with Platform Builder to support platform based on the Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/J1900.

5.2.2.1 Installation and Setup

Note: The installation sequences are crucial for compilation success.

1. Install Visual Studio* 2012 Professional update 2.
2. Install application builder for Visual Studio* 2012.
3. Install Windows Embedded Compact 2013* platform builder.

   In the SETUP dialog during installation, select x86 in the processor architecture section.
4. Install Board Support Package (BSP).

   Download the BSP from Adeneo at: http://www.adeneo-embedded.com/Products/Board-Support-Packages/Intel? or Bsquare at: http://www.bsquare.com/software-downloads.aspx

Note: Registration has to be completed before you are able to download the BDP from Adeneo.

5.2.2.2 Configuring INTEL_CS.bat, platform.reg and platform.bib to Include Intel® EMGD to the WEC2013* Image

1. From the C:\WinCE800\platform\INTEL_CS folder, open file **INTEL_CS.bat.** Find the “@REM set IMGRAM256=” and change it to “set IMGRAM256=0”.
2. From the C:\WinCE800\platform\INTEL_CS\FILES folder, open the file **platform.reg**. Find the line **ENDIF BSP_DISPLAY_RAGEXL**, Paste the following codes between **ENDIF BSP_DISPLAY_RAGEXL and ENDIF BSP_NODISPLAY!**

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\PCI\Template\EMGD]

  "Dll"="isr_emgd.dll"

          "Class"=dword:03

   "SubClass"=dword:00

   "ProgIF"=dword:00

   "VendorID"=multi_sz: "8086","8086","8086","8086","8086"

   "DeviceID"=multi_sz: "0f31","0412","0406","0416","0426"

   "Prefix"="IGD"
```

```
    "IsrDll"="isr_emgd.dll"

    "IsrHandler"="isr_handler"

[HKEY_LOCAL_MACHINE\System\GDI\Drivers]

    "Display"="ddi_emgd.dll"

[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
"D3DMOverride"="ddi_emgd.dll"

[HKEY_LOCAL_MACHINE\System\D3DM\Drivers]
"RemoteHook"="ddi_emgd.dll"

[HKEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]

"Candidate3"="Drivers\\Display\\Intel"

;#include "C:\WINCE800\OSDesigns\OSDesign1\OSDesign1\RelDir\

INTEL_CS_x86_Release\emgd_filters.reg"

;[HKEY_LOCAL_MACHINE\System\GDI\Monitors]

;"Total Monitors"=dword:2
```

Note: To configure the Display in the Extended mode, uncomment the above two lines (Remove "**;**").

To include Intel® EMGD DirectShow codecs in the Windows Embedded Compact 2013* OS image, you need to comment out the below line (Remove ";").

```
;#include
"C:\WINCE800\OSDesigns\OSDesign1\OSDesign1\RelDir\INTEL_CS_x86_Re
lease\emgd_filters.reg"
```

3. From the C:\WinCE800\platform\INTEL_CS\FILES folder, open the file platform**.bib**.
4. Add the following codes at the end of the file.

```
ddi_emgd.dll       $(_FLATRELEASEDIR)\ddi_emgd.dll        NK SHK
dp.dll             $(_FLATRELEASEDIR)\dp.dll              NK SHK
analog.dll         $(_FLATRELEASEDIR)\analog.dll          NK SHK
hdmi.dll           $(_FLATRELEASEDIR)\hdmi.dll            NK SHK
isr_emgd.dll       $(_FLATRELEASEDIR)\isr_emgd.dll        NK SHK
libGLESv2.dll      $(_FLATRELEASEDIR)\libGLESv2.dll       NK SH
libegl.dll         $(_FLATRELEASEDIR)\libegl.dll          NK SH
```

Here $(_FLATRELEASEDIR) Refer to the path:

C:\WINCE800\OSDesigns\OSDesign1\OSDesign1\RelDir\INTEL_CS_x86_Release

Download the latest Intel® EMGD Driver Package from the Intel EDC Website or Contact your Intel representative. Unzip and copy the contents of [Driver] folder from the Intel® EMGD driver package to [INTEL_CS_x86_Release] folder.

Intel® EMGD supports hardware accelerated video decode on Windows Embedded Compact 2013*, the Intel® EMGD Windows Embedded Compact 2013* Microsoft DirectShow* filters are provided in the form of middleware codec libraries (DLLs) that will interface with the Intel® EMGD Windows Embedded Compact 2013* driver to operate. For installation and setup refer Intel® EMGD for Windows Embedded Compact 7* and Windows Embedded Compact 2013* Microsoft DirectShow* Codecs with Intel® Atom™ processor E3800 Product Family/Intel® Celeron® Processor N2807/N2930/ J1900 Video Acceleration Users Guide.

5.3 Configuring Intel® EMGD for Microsoft Windows Embedded Compact 7* / Windows Embedded Compact 2013*

The Intel® EMGD provides configuration options to customize the display and others configuration needed by the embedded systems. The Intel® EMGD for Windows Embedded Compact 7*/Windows Embedded Compact 2013* configuration support flexible configuration option for embedded customers such as booting up the display to single/clone/extended/vertical extended mode, setting to turn on read the EDID from the panel or setting user DTD for EDID-less panel, display rotation/flip, etc.

The Intel® EMGD Windows Embedded Compact 7*/Windows Embedded Compact 2013* Driver supports two types of configuration method.

The driver reads the display configuration following the priority below:

1. Read from VBT data generated from BMP Tool
2. Read from Modular BIOS Interface (MBI) generated from BMP Tool

5.3.1 Read from Video BIOS (VBIOS) VBT

The VBT is a block of customizable platform specific data. It holds platform specific information used by the EMGD VBIOS and Driver.

1. Download the latest version of VBIOS from Intel® Embedded Design Center (http://www.intel.com/p/en_US/embedded/hwsw/software/emgd#download) or contact your Intel representative.
2. Save the files to a convenient directory on your hard drive. Notice that there are several types of files:
   — `.bsf` BIOS script files
   — `.dat` Video BIOS Data file
3. Start the BMP utility. Select File, **Open** and point to the directory where the VBIOS files are located. (Example: `vga.dat` and `vga.bsf`)

Note: The BMP tool is available in the VBIOS package.

4. Select the `.dat` file. Note that the `.bsf` file will automatically load to the BMP tool. However, if it does not, you will need to explicitly load it. Appropriate changes can now be made to the VBT. Refer to Section 3.1, "Configuring VBT Using BMP" for more details.
5. After changes have been made, save the file by selecting File, then **Save** or **Save As** to make sure the changes take effect.
6. The `xxx_yyyy.dat` file is now ready to be merged with the system BIOS to create a BIOS file that can then be flashed onto the platform using the appropriate flash tool.

Note: If you will be making VBIOS configuration changes, you will need access to your BIOS vendor's merge tool and flash utility as well as have the support files to make “VBT” (Video Bios Table) changes. Changes are merged into your BIOS and VBIOS images then flashed back onto the platform. Contact your BIOS vendor for details.

7. The driver will read the VBT Data information from the VBIOS.

5.3.2 Read from Modular Bios Interface (MBI)

This option is mainly for firmware less boot. Some of the system will doesn’t have the firmware installed. User can extract the VBT data information to a MBI file. The MBI file can copy into WEC7 environment and driver can read the VBT data information directly from the binary file.

1. Start the BMP utility. Select `File` -> **`Open`** and point to the directory where the VBIOS files are located. (Example: vga.dat and vga.bsf)

Note: The BMP tool is available in the VBIOS package.

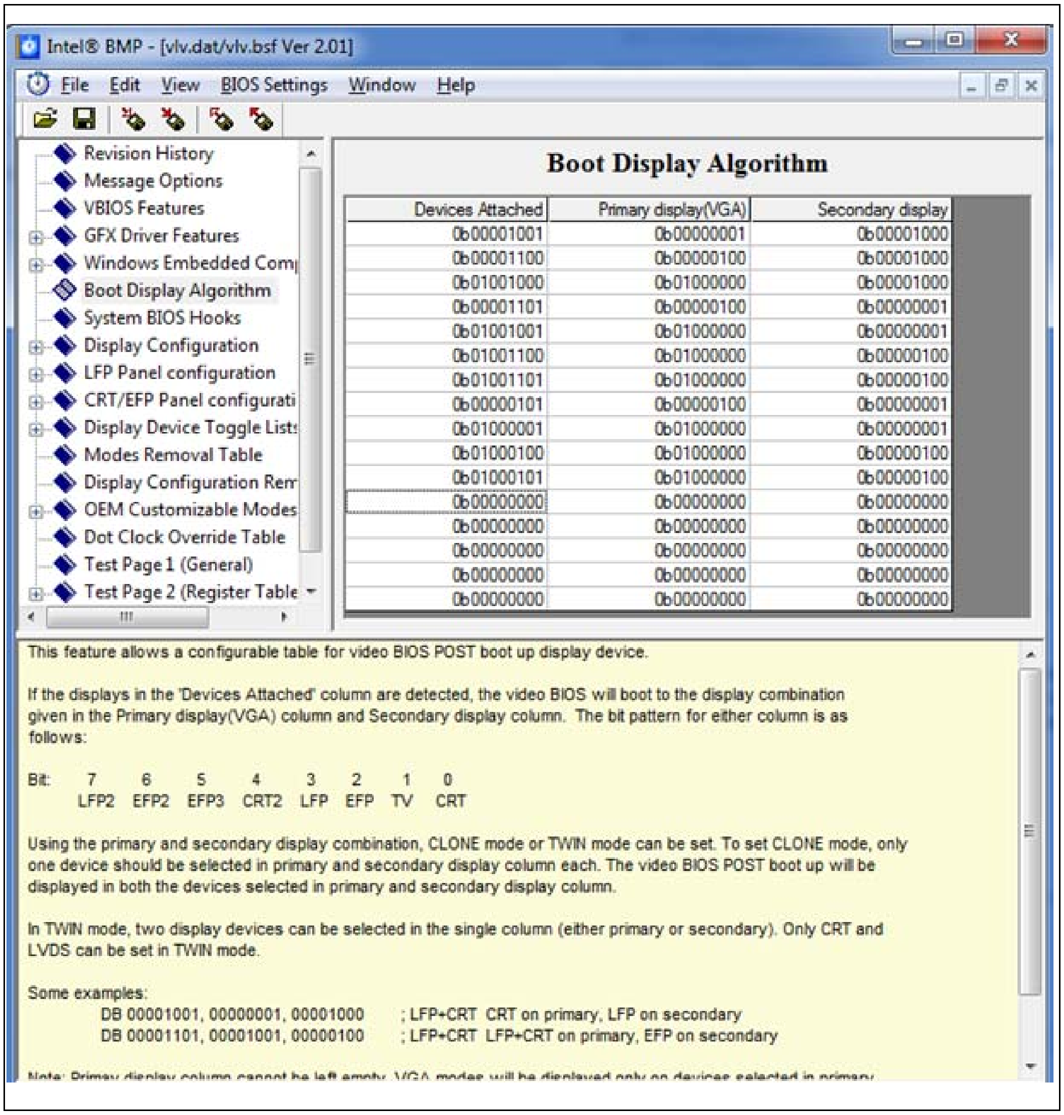
2. Select the `.dat` file. Note that the `.bsf` file automatically loads to the BMP tool. However, if it does not, you will need to explicitly load it.

   Appropriate changes can now be made to the VBT. Refer to Section 3.1, “Configuring VBT Using BMP” for more details.
3. After the changes are made, select File, **Create an MBI module ...**

   This will create a `vga.mbi` file to the desktop.
4. To include the `mbi` file to `nk` system file, go to the `C:\WinCE700\platform\INTEL_CS\`**`FILES`** folder.
5. Edit `platform.bib`, adding this line at the end of the file,:

```
Vga.mbi <path_to_mbi_file>\Vga.mbi      NK SHK
```

5.4 Boot Display Algorithms (For VBIOS VBT and MBI only)



Boot Display Algorithm

Devices Attached	Primary display(VGA)	Secondary display
0b00001001	0b00000001	0b00001000
0b00001100	0b00000100	0b00001000
0b01001000	0b01000000	0b00001000
0b00001101	0b00000100	0b00000001
0b01001001	0b01000000	0b00000001
0b01001100	0b01000000	0b00000100
0b01001101	0b01000000	0b00000100
0b00000101	0b00000100	0b00000001
0b01000001	0b01000000	0b00000001
0b01000100	0b01000000	0b00000100
0b01000101	0b01000000	0b00000100
0b00000000	0b00000000	0b00000000
0b00000000	0b00000000	0b00000000
0b00000000	0b00000000	0b00000000
0b00000000	0b00000000	0b00000000
0b00000000	0b00000000	0b00000000

This feature allows a configurable table for video BIOS POST boot up display device.

If the displays in the 'Devices Attached' column are detected, the video BIOS will boot to the display combination given in the Primary display(VGA) column and Secondary display column. The bit pattern for either column is as follows:

```
Bit:   7     6     5     4     3    2    1    0
      LFP2  EFP2  EFP3  CRT2  LFP  EFP  TV   CRT
```

Using the primary and secondary display combination, CLONE mode or TWIN mode can be set. To set CLONE mode, only one device should be selected in primary and secondary display column each. The video BIOS POST boot up will be displayed in both the devices selected in primary and secondary display column.

In TWIN mode, two display devices can be selected in the single column (either primary or secondary). Only CRT and LVDS can be set in TWIN mode.

Some examples:

```
        DB 00001001, 00000001, 00001000      ; LFP+CRT  CRT on primary, LFP on secondary
        DB 00001101, 00001001, 00000100      ; LFP+CRT  LFP+CRT on primary, EFP on secondary
```

Note: Primary display column cannot be left empty. VGA modes will be displayed only on devices selected in primary

If the displays in the Devices Attached column are detected, the video BIOS boots to the display combination given in the Primary display (VGA) column and Secondary display column. The bit pattern for either column is as follows:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	LFP2	EFP2	EFP3	CRT2	LFP	EFP	TV	CRT

Using the primary and secondary display combination, CLONE mode or TWIN mode can be set. To set CLONE mode, only one device should be selected in primary and secondary display column each. The video BIOS POST boot up will be displayed in both the devices selected in the primary and secondary display columns.

In TWIN mode, two display devices can be selected in the single column (either primary or secondary). Only CRT and LFP can be set in TWIN mode.

Some examples:

`DB 00001001, 00000001, 00001000 ;` LFP+CRT; CRT on primary, LFP on secondary

`DB 00001101, 00001001, 00000100` ; LFP+CRT; LFP+CRT on primary, EFP on secondary

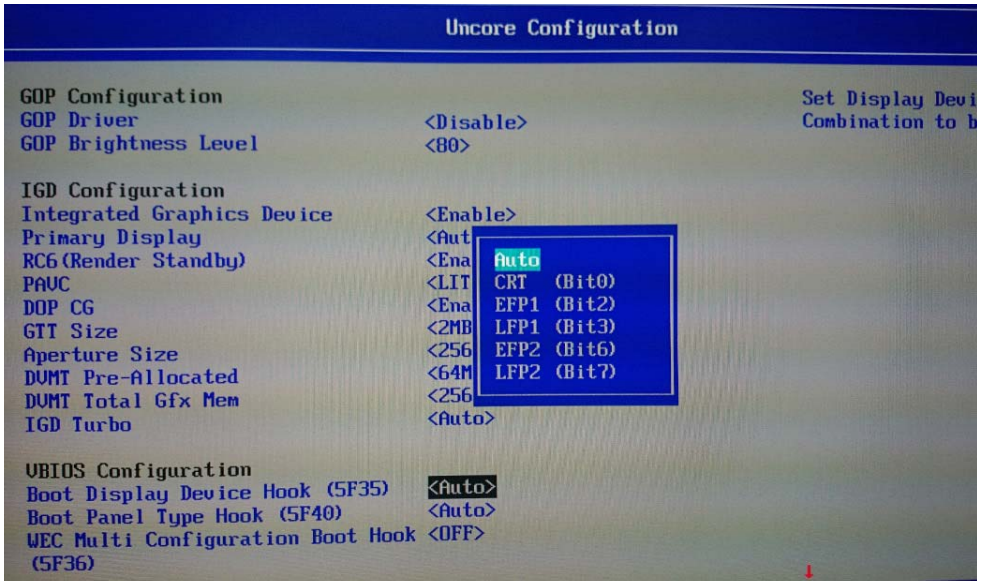
Notes: The primary display column cannot be left empty. VGA modes will be displayed only on devices selected in primary display column.

This table is skipped when a valid display combination is returned by the 5F35h (Boot Display) system BIOS hook.

5.4.1 Boot up Display Devices Hook (Interrupt 5f35H)

Notification is given to system BIOS to perform display device sensing and to override the video display default setting. When this feature is enabled (USE INTERRUPT 15H), the video BIOS will call the interrupt 15h, 5F35h hook as specified in the "*Intel® EMGD - BMP User's Guide Addendum For Windows* Embedded Compact*" in the middle of video BIOS POST. When disabled, 5F35h will not be called.

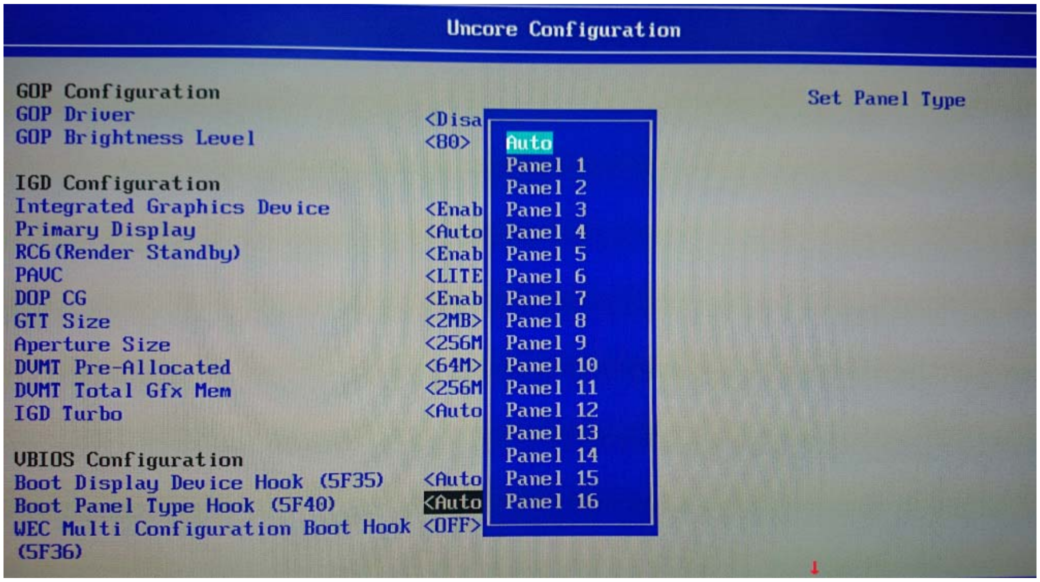
When 5f35H interrupt is enabled, this enable the CMOS setting boot up display devices hook. The VBIOS and driver will boot up follow the CMOS selection.



```
                         Uncore Configuration

GOP Configuration                                        Set Display Devi
GOP Driver                          <Disable>            Combination to b
GOP Brightness Level                <80>

IGD Configuration
Integrated Graphics Device          <Enable>
Primary Display                     <Aut  Auto
RC6(Render Standby)                 <Ena  CRT   (Bit0)
PAVC                                <LIT  EFP1  (Bit2)
DOP CG                              <Ena  LFP1  (Bit3)
GTT Size                            <2MB  EFP2  (Bit6)
Aperture Size                       <256  LFP2  (Bit7)
DVMT Pre-Allocated                  <64M
DVMT Total Gfx Mem                  <256
IGD Turbo                           <Auto>

VBIOS Configuration
Boot Display Device Hook (5F35)     <Auto>
Boot Panel Type Hook (5F40)         <Auto>
WEC Multi Configuration Boot Hook   <OFF>
(5F36)
```

5.4.2 Boot Panel Type Hook (5f40H) - (only for VBIOS VBT)

When 5F40H interrupt is enabled (USE INTERRUPT 15H), it allows the system to return the panel type to be used in booting the system and from then on the video BIOS and driver. When disabled, the BPM panel type will be used.



Uncore Configuration

GOP Configuration
GOP Driver <Disa
GOP Brightness Level <80>

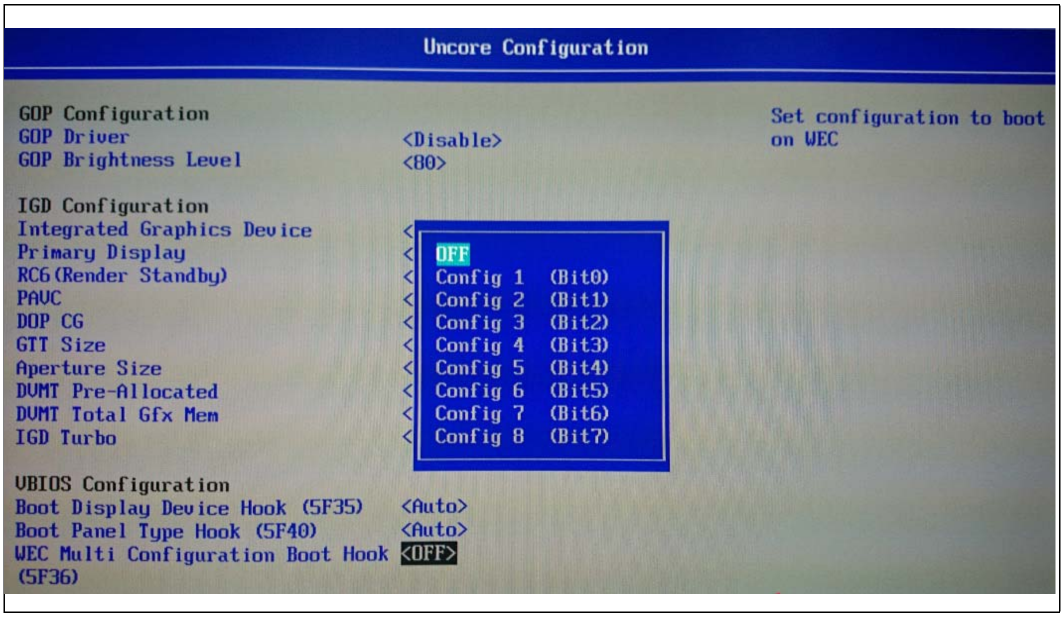
IGD Configuration
Integrated Graphics Device <Enab
Primary Display <Auto
RC6(Render Standby) <Enab
PAVC <LITE
DOP CG <Enab
GTT Size <2MB>
Aperture Size <256M
DVMT Pre-Allocated <64M>
DVMT Total Gfx Mem <256M
IGD Turbo <Auto

VBIOS Configuration
Boot Display Device Hook (5F35) <Auto
Boot Panel Type Hook (5F40) <Auto
WEC Multi Configuration Boot Hook (5F36) <OFF>

Set Panel Type

5.4.3 WEC Multi Configuration Boot Hook (Interrupt 5F36H) – (only for VBIOS VBT)

When 5F36H function is enabled (USE INTERRUPT 15H), user can use the multiple configurations through the CMOS settings. VBIOS and driver might work differently in term of multiple configurations.



```
                              Uncore Configuration

GOP Configuration                                             Set configuration to boot
GOP Driver                          <Disable>                 on WEC
GOP Brightness Level                <80>

IGD Configuration
Integrated Graphics Device          <
Primary Display                     <  OFF
RC6(Render Standby)                 <  Config 1  (Bit0)
PAVC                                <  Config 2  (Bit1)
DOP CG                              <  Config 3  (Bit2)
GTT Size                            <  Config 4  (Bit3)
Aperture Size                       <  Config 5  (Bit4)
DVMT Pre-Allocated                  <  Config 6  (Bit5)
DVMT Total Gfx Mem                  <  Config 7  (Bit6)
IGD Turbo                           <  Config 8  (Bit7)

VBIOS Configuration
Boot Display Device Hook (5F35)     <Auto>
Boot Panel Type Hook (5F40)         <Auto>
WEC Multi Configuration Boot Hook   <OFF>
(5F36)
```

The driver will boot up according to the index number selection.

For example, if users choose the selection Config 2 (Bit 1) in Multi configuration Boot hook, the driver automatically boots up the system with the Config 2 under multiple configuration in VBT.

Note: This configuration is only supported by VBIOS VBT. It is not supported by MBI.

5.5 Miscellaneous

This section describes an additional patch or workaround that is needed for general usage of the driver.

5.5.1 OpenGL ES 2.0 Required Patch

An OpenGL ES* 2.0 patch is required for proper application execution with general usage of the driver. To launch a 3D OpenGL ES* 2.0 application, the stack size needs to be increased by adding the following line of code to the GLES 2.0 application source file:

LDEFSTACK=/STACK:262144,4096

Refer to the Specification Update (Errata) of the driver release for more detailed information about the driver for known issues and possible resolutions.

§ §

Appendix A Intel® 5F Extended Interface Functions

The BIOS provides a set of proprietary function calls to control operation of the extended features. These function calls all use AH = 5Fh in their designed interface for easy identification as a proprietary function.

These functions are designed to maintain maximum compatibility with the Desktop and Mobile Video BIOS. As such many of the definitions behave identically. When the behavior of the Embedded Video BIOS is not identical to the Desktop and Mobile Video BIOS it is noted.

In addition to these 5F functions, the Video BIOS also supports all 4F functions defined by the *VESA* BIOS Extension (VBE) Core Functions Standard, Version 3.0* with the exception of the 0A function (Return VBE Protected Mode Interface). All other functions, from 00 through 09 and 0B are supported by the Video BIOS. The *VESA* BIOS Extension (VBE) Core Functions Standard, Version 3.0* document is available from

http://www.vesa.org/vesa-standards/free-standards/

The table below provides a summary of the Intel® EMGD supported Intel 5F functions.

Table 6. Summary of Intel 5F Extended Interface Functions

Function	Function Name	Description
BIOS Extended Interface Functions		
5F01h	Get Video BIOS Information	Gets VBIOS Build Information.
5F05h	Refresh Rate	Sets a new vertical refresh rate for a given mode and returns the current vertical refresh rate
5F10h	Get Display Memory Information	Returns information about the linear memory.
5F1Ch	BIOS Pipe Access	Sets the BIOS pipe access and returns the BIOS pipe access status.
5F29h	Get Mode Information	Returns information on the requested mode.
5F68h	System BIOS Callback	Allows SoftBIOS to do any system callbacks through INT 15h
Hooks for the System BIOS		
5F31h	POST Completion Notification Hook	Signals the completion of video POST (Power On Self Test)
5F33h	Hook After Mode Set	Allows System BIOS to intercept Video BIOS at the end of a mode set.
5F35h	Boot Display Device Hook	Allows System BIOS to override video display default setting.
5F36h	WEC Multi Configuration Boot Hook	Allows System BIOS to supply a configuration ID that is passed to the driver.
5F38h	Hook Before Set Mode	Allows System BIOS to intercept Video BIOS before setting the mode.
5F40h	Boot panel Type Hook	Supports local flat panel features.

A.1 BIOS Extended Interface Functions

The BIOS provides a set of proprietary function calls to control operation of the extended features. These function calls all use AH = 5Fh in their designed interface for easy identification as a proprietary function.

These functions are designed to maintain maximum compatibility with the Desktop and Mobile Video BIOS. As such many of the definitions behave identically. When the behavior of the Embedded Video BIOS is not identical to the Desktop and Mobile Video BIOS it is noted.

A.1.1 5F01h – Get Video BIOS Information

This function returns the Video BIOS Build information.

Note: This function is an extension of the Desktop and Mobile Video BIOS. If register ECX does not contain ASCII characters “IEGD” then the VBIOS is not described by this specification.

Calling Register:

AX = 5F01h, Get Video Information function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
EBX = 4 bytes Video BIOS Build Number ASCII string, e.g., ‘1000’
ECX = 4 bytes Embedded Identifier, ASCII string ‘IEGD’

A.1.2 5F05h – Refresh Rate

This function sets a new vertical refresh rate for a given mode and returns the current vertical refresh rate and available refresh rate for a given non-VGA mode.

A.1.2.1 5F05h, 00h – Set Refresh Rate

This sub-function sets a new default refresh rate for the selected pipe. If the mode is currently active, the CRT controller and other registers will be automatically programmed setting the requested refresh rate.

Note: This function is not entirely compatible with the Desktop and Mobile versions. It is not possible to set the refresh rate for a given mode in advance. This function sets the “desired” refresh rate which will be applied to all subsequent mode sets when possible. If the mode provided in BL is the current mode, then a mode change will be automatically performed.

Calling Register:

AX = 5F05h, Refresh Rate function
BH = 00h, Set Refresh Rate sub-function
BL = Mode Number
ECX = Refresh rate (indicated by setting one bit):
  Bits 31 - 9 = Reserved
  Bit 8 = 120 Hz
  Bit 7 = 100 Hz
  Bit 6 = 85 Hz
  Bit 5 = 75 Hz
  Bit 4 = 72 Hz
  Bit 3 = 70 Hz
  Bit 2 = 60 Hz
  Bit 1 = 56 Hz
  Bit 0 = 43 Hz (Interlaced - Not supported)

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
  = 005Fh, Function supported and successful
  = 015Fh, Function supported but failed

A.1.2.2 5F05h, 01h – Get Refresh Rate

This sub-function returns current vertical refresh rate for the selected pipe and available refresh rates information for a given Non-VGA mode.

Note: This sub-function returns a status of supported but failed (AX = 015Fh) if executed with a standard VGA mode.

Calling Registers:

AX = 5F05h, Refresh Rate function
BH = 01h, Get Refresh Rate sub-function
BL = Mode number

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
  = 005Fh, Function supported and successful
  = 015Fh, Function supported but failed
EBX = Available refresh rates (indicated by one or more bits set):
  Bits 31 - 9 = Reserved
  Bit 8 = 120 Hz
  Bit 7 = 100 Hz
  Bit 6 = 85 Hz
  Bit 5 = 75 Hz
  Bit 4 = 72 Hz
  Bit 3 = 70 Hz
  Bit 2 = 60 Hz
  Bit 1 = 56 Hz
  Bit 0 = 43 Hz (Interlaced - Not supported)
ECX = Current refresh rate (see EBX for bit definitions)

A.1.3 5F10h – Get Display Memory Information

This function returns information regarding the linear memory starting address, size and memory mapped base address.

Calling Register:

AX = 5F10h, Get Linear Display Memory Information function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
ESI = Display memory base address
ECX = Total physical display memory size (in bytes)
EDX = Available display memory size (in bytes)
EDI = Memory Mapped I/O Base Address
EBX = Stride (memory scan line width in bytes)

A.1.4 5F1Ch – BIOS Pipe Access

This function will set the BIOS pipe access or return the BIOS pipe access status.

A.1.4.1 5F1Ch, 00h – Set BIOS Pipe Access

This sub-function will set the currently selected pipe. All 5f functions operate on the currently selected pipe.

When not in clone modes this value cannot be set.

Calling Registers:

AX = 5F1Ch, BIOS Pipe Access function
BH = 00h, Set BIOS Pipe Access sub-function
CH = BIOS Pipe access:
= 00h, Pipe A
= 01h, Pipe B

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

A.1.4.2 5F1Ch, 01h – Get BIOS Pipe Access

This sub-function will return the currently selected pipe.

Calling Registers:

AX = 5F1Ch, BIOS Pipe Access function
BH = 01h, Get BIOS Pipe Access sub-function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
CH = BIOS Pipe access:
= 00h, Pipe A
= 01h, Pipe B

A.1.5 5F29h – Get Mode Information

This function returns the requested mode's resolution, color depth, and maximum required bandwidth using its current refresh rate. This function is applied to extended-graphics modes only. If the mode number is not an extended graphics mode, the function will return failure.

Calling Registers:

AX = 5F29h, Get Mode Information function
BH = Mode To Use:
= 80h, Current Mode
= 00h - 7Fh, Given Mode Number

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
EBX bits 31 - 16 = Mode horizontal (X) resolution in pixels
EBX bits 15 - 0 = Mode vertical (Y) resolution in pixels
ECX bits 31 - 16 = Maximum bandwidth in megabytes per second
ECX bits 15 - 0 = Color depth in bits per pixel

A.1.6 5F68h – System BIOS Callback

This is a generic function that allows SoftBIOS to do any system callbacks through INT 15h. The Input/Output of this function is dependent on the definition of the desired INT 15h hook except for the EAX register.

Calling Registers:

AX = 5F68h, System BIOS Callback Function
EAX bits 31:16 = System BIOS INT 15h Hook Function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

A.2 Hooks for the System BIOS

The video BIOS performs several system BIOS interrupt function calls (interrupt 15h hooks). Each function provides the system BIOS with the opportunity to gain control at specific times to perform any custom processing that may be required. After each interrupt hook, the system BIOS must return control to the video BIOS. INT 10h calls could be made within the INT 15h hook calls provided that it is not recursive and thus cause a deadlock.

A.2.1 5F31h – POST Completion Notification Hook

This hook signals the completion of video POST (Power On Self Test). The hook executes after the sign-on message is displayed and PCI BIOS resizing.

Calling Registers:

AX = 5F31h, POST Completion Notification Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful

A.2.2 5F33h – Hook After Mode Set

This hook allows the system BIOS to intercept the video BIOS at the end of a mode set.

Calling Registers:

AX = 5F33h, Hook After Mode Set
BH = Number of character columns
BL = Current mode number
CH = Active display page

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful

A.2.3 5F35h – Boot Display Device Hook

This hook allows the system BIOS to override the video display default setting. The graphics BIOS will set the returned video display during POST (power up initialization).

Note: This function is not entirely compatible with the Desktop and Mobile Video BIOS. The bits in CL have a configurable mapping to the Port Numbers as defined in Section 5.4.1. The assigned meanings used in the Desktop specification can be duplicated with a correct configuration. The values below are the default values if no “Common To Port” mapping is provided.

Calling Registers:

AX = 5F35h, Boot Display Device Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh);
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
CL = Display Device Combination to boot (1 = Enable display, 0 = Disable display):
= 00h, VBIOS Default
Bit 7 = LFP1 (or Port 1)
Bit 6 = EFP2 (or Port 1)
Bit 3 = LFP1 (or Port 1)
Bit 2 = EFP1 (or Port 1)
Bit 0 = CRT (or Port 0)

A.2.4 5F36h – WEC Multi Configuration Boot Hook

This function is known as “WEC Multi Configuration Boot Hook” in the Desktop and Mobile Video BIOS. It allows the system BIOS to supply a configuration ID that will eventually be passed to the driver. This configuration ID is unused by the Video BIOS; however, it alters the behavior of the driver as described in Section 5.4.3

Calling Registers:

AX = 5F36h, Config ID Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
CL = Configuration ID
Bit 7 = Config 8
Bit 6 = Config 7
Bit 5 = Config 6
Bit 4 = Config 5
Bit 3 = Config 4
Bit 2 = Config 3
Bit 1 = Config 2
Bit 0 = Config 1

A.2.5 5F38h – Hook Before Set Mode

This hook allows the system BIOS to intercept the video BIOS before setting the mode.

Calling Registers:

AX = 5F38h, Hook Before Set Mode
CL = New video mode to be set

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful

A.2.6 5F40h – Boot Panel Type Hook

This function supports local flat panel only features.

A.2.6.1 5F40h, 05h – Get Configuration ID

This function is used to return the Configuration ID.

Note: This function is known as “Get Local Flat Panel Number” in the Desktop and Mobile Video BIOS. This function performs a similar purpose however, the configuration IDs have no pre-defined meaning. The Configuration ID is reported to the Intel® EMGD.

Calling Registers:

AX = 5F40h, Local Flat Panel Support function
BH = 05h, Get Config ID Subfunction

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

A.2.6.2 5F40h, 08h - Set the LVDS Backlight Level

This function is used to handle the LVDS backlight level.

Note: To enable to LVDS backlight control functionality, ensure you configure the following attributes in the BMP Attribute Settings page:

— Intensity (attribute #70) to your desired value.

— Inverter Frequency (attribute #71) based on your requirement.

— Backlight Method (attribute #72) to Legacy + PWM mode(1)

Calling Registers:

AX = 5F40h, Local Flat Panel Support function
BH = 08h, Set the LVDS backlight level Subfunction
BL = 0 – 255 (backlight level)

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

§ §