



InsydeH2O® 5.4
Alder Lake Power-On
Technical Reference Guide

Revision 0.6
Aug 10, 2021



Insyde Software Corp.

Copyright (c) 2020, All Rights Reserved. Insyde Software.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form, or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of Insyde Software Corp.

Disclaimer

Insyde Software provides this document and the programs "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

This document could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in future revisions of this document. Insyde Software is under no obligation to notify any person of the changes.

The following trademarks are used in this document:

Insyde and InsydeH2O are registered trademarks or trademarks of Insyde Software. All other trademarks or trade names are property of their respective holders.

Insyde Software Corp.

Contents

1 Introduction.....	6
2 Power-On Porting.....	7
2.1 Build project.....	7
2.1.1 Build your project in the path as follows:.....	7
2.1.2 Configure the EC as below:.....	7
2.1.3 Configure Super IO as below:.....	7
2.1.4 Start porting and check Project.dsc/Project.fdf.....	7
2.1.5 Trouble Shooting.....	7
2.1.5.1 OpenSSL Caution.....	7
2.1.5.2 SignTool Caution.....	8
2.2 OpenSSL.....	9
2.2.1 Build OpenSSL in Windows.....	9
2.3 Flash Map.....	10
2.4 EC (BIOS and EC in the same flash part).....	11
2.4.1 Configure the EC as below:.....	11
2.4.2 Put project EC binary in this path.....	12
2.4.3 Modify project EC binary path and filename in Project.fdf.....	12
2.4.4 Set these switches in Project.env.....	12
2.4.5 Implement and modify EC related Kernel and Chipset OEM service in.....	12
2.5 No EC support.....	13
2.6 Disable Intel CRB EC Function.....	13
2.7 Change GOP/VBT.....	13
2.8 GPIO Configuration.....	14
2.8.1 ChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported == TRUE.....	14
2.8.2 ChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported == FALSE.....	15
2.9 Super IO Support.....	15
2.9.1 Create SIO package (SioXXXXPkg).....	15
2.9.2 Import SIO package.....	16
2.9.3 Setting the SIO related PCDs for project own requirements.....	16
2.10 No Super IO.....	17
2.11 PCI IRQ Routing.....	17
2.11.1 Program the Legacy Routing Table.....	17
2.11.2 ACPI routing table.....	18
2.11.3 Example 1: Change internal device 31 Integrated GbE routing to the same PIRQ:.....	18
2.11.4 Example 2: Set INTA of PCIe root port 3 link to PIRQF and link PIRQF to IRQ7:.....	19
2.12 SODIMM.....	21
2.13 On-board RAM.....	21

2.14 BoardID integrate.....	22
2.14.1 There are three PCDs about board ID configuration in our source code:.....	22
2.14.2 How to set board ID correctly on OEM project side?.....	22
2.14.3 Compare to old BOARDID feature.....	23
2.15 Native FSP Build.....	23
3 Power-On Tips.....	24
3.1 Minimum Devices Support.....	24
3.2 Minimum CPU Features Support.....	24
3.3 Boot Mode.....	24
3.4 Policy Setting.....	25
3.5 Emulation Variable.....	25
3.6 Debug Methods.....	26
3.7 Add Option ROM.....	26
3.8 GlobalNvsArea mechanism.....	26
3.9 Software SMI.....	27
3.10 CMOS Offset Definition.....	27
3.11 SVID/SSID.....	27
3.12 Setup Configuration.....	27
4 Project Board Design Checklist.....	29
4.1 H2ODDT USB Port Support.....	29
4.2 H2ODDT Com Port Support.....	29
4.3 H2ODDT PCH UART Support.....	29

Insyde Software Corp.

Revision History

Revision Number	Description	Author	Release Date
0.1	Initial release.	Evonne Li	October 29, 2020
0.2	Add Native FSP build page	Rex Wu	Dec 23, 2020
0.3	Update PCH UART DDT for ADL-P	Conrad Tang	Jan 07, 2021
0.4	Update PCH URAT for efidebug note	Evan Chen	Feb 24, 2021
0.5	Add information about OpenSSL and SignTool	Esther Lee	June 02, 2021
0.6	Modified Native FSP Build section.	Rex Wu	Aug 10, 2021

Insyde Software

1 Introduction

The purpose of the InsydeH2O 5.4 AlderLake Technical Power-On Reference Guide is intended to help guide OEM members and Insyde source code customers in using AlderLake source code for their board power on event.

This document includes:

1. Power on Porting
2. Power on Tips

The following terms are used throughout this document to describe varying aspects of input localization:

\$(PROJECT_REL_PATH)	For CRB, it's Board/Intel For Project, it may be Board/XYZ
\$(PROJECT_PKG)	For CRB, it's AlderLakeSMultiBoardPkg For Project, it may be BoardNameBoardPkg
\$(CHIPSET_REL_PATH)	Intel/AlderLake
\$(CHIPSET_PKG)	AlderLakeChipsetPkg
\$(PLATFORM_SI_PACKAGE)	ClientOneSiliconPkg
\$(PLATFORMSAMPLE_PACKAGE)	AlderLakePlatSamplePkg
\$(PLATFORM_BOARD_PACKAGE)	AlderLakeBoardPkg

3. AlderLakeBoardPkg is part of Intel platform sample code, and it's under \$(CHIPSET_REL_PATH)
4. AlderLakeSMultiBoardPkg is Insyde CRB implementation sample, and it's under \$(PROJECT_REL_PATH)

2 Power-On Porting

2.1 Build project

The CRB binary is supposed to be built from *BoardNameMultiBoardPkg*.

2.1.1 Build your project in the path as follows:

BoardNameBoardPkg is for vendor Boards

1. Not chipset name
Example: *AABoardPkg*, not *AlderLakePkg*
2. Multi-SKU boards use *chipsetMultiBoardPkg*, i.e: *AlderLakeSMultiBoardPkg*
3. Copy CRB BIOS *AlderLakeSMultiBoardPkg* and change the folder name to your OEM *OemBoardPkg*, i.e : Copy *Board\Intel\AlderLakeSMultiBoardPkg* to *Board\OemName\OemBoardPkg*.
4. **Modify ProjectSetup.bat, set CrbBuild = NO.**

2.1.2 Configure the EC as below:

1. If your project supports EC, follow the steps of section 2.3 to configure EC.
2. If your project does not support EC, follow the steps of section 2.4 to configure EC.

2.1.3 Configure Super IO as below:

1. If your project supports SIO, follow the steps of section 2.8.
2. If your project does not support SIO, follow the steps of section 2.9.

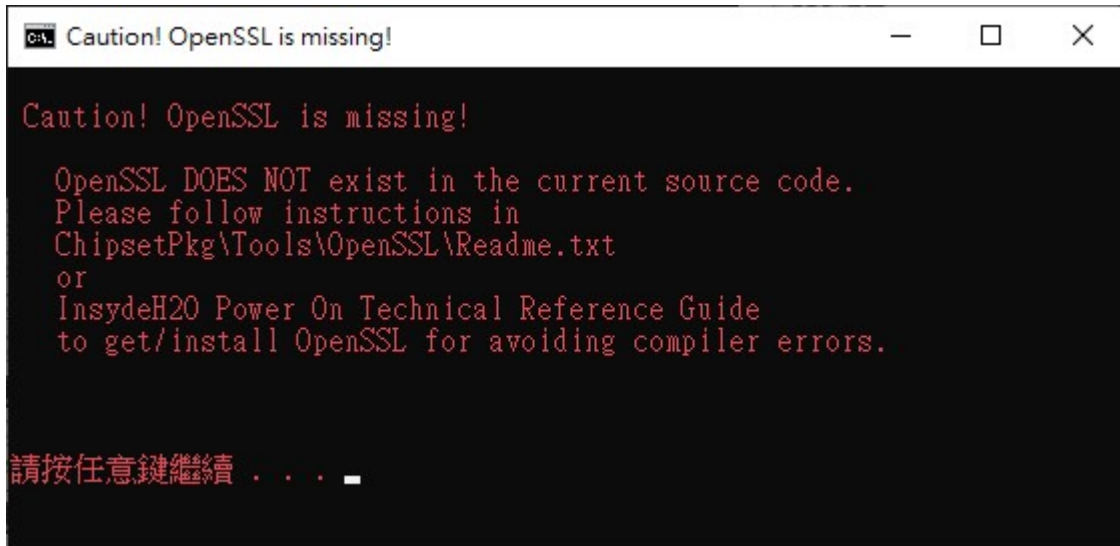
2.1.4 Start porting and check Project.dsc/Project.fdf

Implement which drivers need to be overwritten for your project's requirements.

2.1.5 Trouble Shooting

2.1.5.1 OpenSSL Caution

If below warning message was pop up during build process, it means that sign tool (OpenSSL) is not ready and OpenSSL environment has been checked failed. Please refer to "Chapter 2.2 OpenSSL" to set OpenSSL environment.



NOTE:

Minimum tool requirement :

libcrypto-1_1.dll

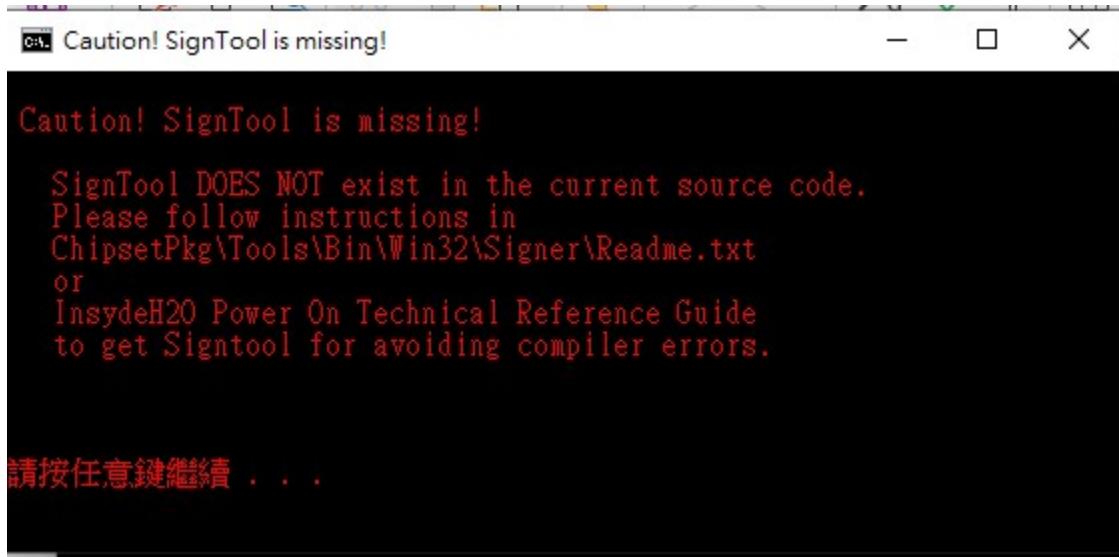
libssl-1_1.dll

openssl.exe

These files can be put in Intel\AlderLake\AlderLakeChipsetPkg\Tools\OpenSSL or refer to “Chapter 2.2 OpenSSL” to set OpenSSL environment.

2.1.5.2 SignTool Caution

If below warning message was pop up during build process, it means that Signtool is not ready and Signtool has been checked failed. Please refer to “Chapter 2.3 Signtool” to set Signtool environment.

**NOTE:**

Singtool.exe

This file can be put in Intel\AlderLake\AlderLakeChipsetPkg\Tools\Bin\Win32\Signer or refer to “Chapter 2.3 Signtool” to get Signtool.

2.2 OpenSSL

OpenSSL is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library. OpenSSL is licensed under an Apache-style license, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions.

OpenSSL is available for Windows* and Linux*. The latest version at the time of this document is OpenSSL 1.1.1g and the version Intel used official is OpenSSL 1.1.0j 20 Nov 2018(TigerLake).

The Linux* version can be downloaded from <http://openssl.org>

The latest source code is in the git repository in GitHub*.

```
$git clone https://github.com/openssl/openssl.git
```

2.2.1 Build OpenSSL in Windows

For the Windows operating system, after fetching source files from OpenSSL.org, run the commands below to build. The source contains all the information for building the binaries on the Windows operating system.

Requirements for the Windows build are Perl* and NMAKE. Both are free (NMAKE via Visual Studio Community Edition <https://visualstudio.microsoft.com/downloads/>).

1. Download and install Perl at <http://strawberryperl.com> or ActivePerl.
2. Download and install Visual Studio Community Edition.
3. Install and set path for nasm.exe and rc.exe in the Windows SDK to the system path.
4. Open Visual Studio, select the folder of openssl, then click the Tools Tab.
 - a. Tools —>Command Line —>Developer Command Prompt
5. Run the commands below.

On Windows, only pick one of the targets for configuration. In Windows, 10 x86 system:

\$perl Configure VC-WIN32

```
$ perl Configure {VC-WIN32 | VC-WIN64A | VC-WIN64I | VC-CE}
$ nmake
$ nmake test
$ nmake install
```

To use ActivePerl*, you may also need to install the required modules. Strawberry Perl* windows version includes all required modules. More detail is in the INSTALL file. When doing the \$name install, make sure to add user full access of the file folder, or login as administrator authority. The default is C:\Program Files (x86)\OpenSSL, and C:\Program Files (x86)\Common Files\SSL. Otherwise, you will observe an error to unable to create directory and copy over files.

NOTE: The Windows version OpenSSL binary can also be downloaded from the link below, if you do not want to build from the source.

<https://slproweb.com/products/Win32OpenSSL.html>

2.3 Signtool

Sign Tool is a command-line tool that digitally signs files, verifies signatures in files, and time-stamps files.

This tool is automatically installed with Visual Studio. To run the tool, use [Visual Studio Developer Command Prompt](#) or [Visual Studio Developer PowerShell](#).

Signtool is available for Windows*. Suggestion for the version is x86.

Download and Install Microsoft Visual Studio at <https://visualstudio.microsoft.com/downloads/>

Or Download and Install Microsoft SDL at <https://developer.microsoft.com/zh-tw/windows/downloads/windows-10-sdk/>

Signtool can be found at C:\Program Files (x86)\Windows Kits\10\bin\(\OS version)\x86

2.4 Flash Map

If BIOS image size does not match the BIOS size of description region, please DO NOT USE null binary image appending BIOS image to meet BIOS size of description region in your project, otherwise it will cause BIOS R/W variable error during POST.

We suggest you follow one of the two scenarios as follows:

1. Adjust setting of description region to meet your BIOS image size.

2. Modify Project.fdf to adjust BIOS image size.

2.5 EC (BIOS and EC in the same flash part)

2.5.1 Configure the EC as below:

Copy gSioGuid.PcdSioDummyCfg from Insyde/SioDummyPkg/Package.dsc and modify the content base on the information below:

```
#
# Count from left
# Byte 0 (TypeH): SIO ID High Byte
# Byte 1 (TypeL): SIO ID Low Byte
# Byte 2 (TypeInstance): SIO Instance
# Byte 3 (Device): SIO Device (1=COM, 2=FLOPPY, 3=LPT, 4=KYBD, 5=MOUSE, 127=CFG)
# Byte 4 (DeviceInstance): SIO Device Instance (0 = 1st, 1 = 2nd, etc...)
# Byte 5 (DeviceEnable): SIO Device Enable (0 = FALSE, 1 = TRUE)
# Byte 6 (DeviceBaseAdr): SIO Device Base Address
# Byte 7 (DeviceSize): SIO Device Size
# Byte 8 (DeviceLdn): SIO Device LDN
# Byte 9 (DeivceIrq): SIO Device IRQ
# Byte 10 (DeviceDma): SIO Device DMA
# *The PCDs must be terminated with a NULL entry.
# *KYBD and Mouse are in the same group, so their DI described in sequence
gSioGuid.PcdSioDummyCfg| {\
0xFF, 0x00, 0x00, 0x01, 0x00, 0x01, UINT16(0x03F8), 0x00, 0x00, 0x04, 0x00, \ # Com
0xFF, 0x00, 0x00, 0x01, 0x01, 0x01, UINT16(0x02F8), 0x00, 0x00, 0x03, 0x00, \ # Com
0xFF, 0x00, 0x00, 0x04, 0x00, 0x01, UINT16(0x0060), 0x00, 0x00, 0x01, 0x00, \ # KYBD
0xFF, 0x00, 0x00, 0x05, 0x01, 0x01, UINT16(0x0060), 0x00, 0x00, 0x0C, 0x00, \ # MOUSE
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, UINT16(0x0000), 0x00, 0x00, 0x00, 0x00 \ # End
Entry
}
```

2.5.2 Put project EC binary in this path

`$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Binary/Insyde/Ec`

2.5.3 Modify project EC binary path and filename in Project.fdf

`FILE = $(PROJECT_PKG)/Binary/Insyde/Ec/EmuEc.bin`

2.5.4 Set these switches in Project.env

Set EDK_GLOBAL_EC_IDLE_PER_WRITE_BLOCK to YES

Set EDK_GLOBAL_EC_SHARED_FLASH_SUPPORT to YES

2.5.5 Implement and modify EC related Kernel and Chipset OEM service in

`$(PROJECT_REL_PATH)\$(PROJECT_PKG)\Library\BaseOemSvcKernelLib*.*`

`$(PROJECT_REL_PATH)\$(PROJECT_PKG)\Library\BaseOemSvcChipsetLib*.*`

Follow the steps below to implement your project own BaseOemSvcKernelLib

1. If the relative files are not exist in `$(PROJECT_REL_PATH)\$(PROJECT_PKG)\Library\BaseOemSvcKernelLib` folder. Copy BaseOemSvcKernelLib.inf and the *.c files that you want to implement from `Insyde\InsydeOemServicesPkg\Library\BaseOemSvcKernelLib` into `$(PROJECT_REL_PATH)\$(PROJECT_PKG)\Library\BaseOemSvcKernelLib`.
2. Delete the *.c filename (which you do not need) in [sources] section of BaseOemSvcKernelLib.inf
3. Add the following description in the [LibraryClasses] section of Project.dsc.
`BaseOemSvcKernelLib|$(PROJECT_PKG)\Library\BaseOemSvcKernelLib\BaseOemSvcKernelLib.inf`
4. Modify the source *.c file for your own project.

Follow the steps below to implement your project own BaseOemSvcChipsetLib

1. If the relative files are not exist in `$(PROJECT_REL_PATH)\$(PROJECT_PKG)\Library\BaseOemSvcChipsetLib` folder. Copy BaseOemSvcChipsetLib.inf and the *.c files that you want to implement from `$(CHIPSET_REL_PATH)\$(CHIPSET_PKG)\Library\BaseOemSvcChipsetLib` into `$(PROJECT_REL_PATH)\$(PROJECT_PKG)\Library\BaseOemSvcChipsetLib`.
2. Delete the *.c filename (which you do not need) in [sources] section of BaseOemSvcChipsetLib.inf
3. Add the following description in the [LibraryClasses] section of Project.dsc.
`BaseOemSvcChipsetLib|$(PROJECT_PKG)\Library\BaseOemSvcChipsetLib\BaseOemSvcChipsetLib.inf`

4. Modify the source *.c file for your own project.

2.6 No EC support

If your project does not support EC then you should follow the steps listed below:

1. Set EDK_GLOBAL EC_IDLE_PER_WRITE_BLOCK to NO in Project.env.
2. Set EDK_GLOBAL EC_SHARED_FLASH_SUPPORT to NO in Project.env.
3. Remove SioDummyPkg:

Remove *!import SioDummyPkg/Package.dsc* from Project.dsc.

Remove *!import SioDummyPkg/Package.fdf* from Project.fdf.

2.7 Disable Intel CRB EC Function

If you want to disable Intel CRB EC function before power on your project. You should follow the steps as follows:

1. Set EDK_GLOBAL USE_INTEL_CRB_H8_EC to NO in Project.env.
2. Set EDK_GLOBAL ENABLE_ASL_ECON to NO in Project.env.

(Set "NO" is just for your platform power on and you want to disable all EC function. You still need to set ENABLE_ASL_ECON to "YES" after your platform can successfully boot to windows, and you want to add your EC ASL code)

3. Remove the *OemSvcEcXXXXXXX.c* from BaseOemSvcChipsetLib folder under *\$(PROJECT_REL_PATH)\\$(PROJECT_PKG)\Library*.
4. Remove the *OemSvcEcXXXXXXX.c* from BaseOemSvcKernelLib folder under *\$(PROJECT_REL_PATH)\\$(PROJECT_PKG)\Library*.
5. Modify BaseOemSvcChipsetLib.inf and BaseOemSvcKernelLib.inf under *\$(PROJECT_REL_PATH)\\$(PROJECT_PKG)\Library*.
6. Delete OemSvcEcXXX.c

2.8 Change GOP/VBT

1. Put project GOP driver in following path. If the folder doesn't exist, create it.

\$(PROJECT_REL_PATH)\\$(PROJECT_PKG)\Binary\UefiGop

2. Add project GOP driver in Project.fdf

Example:

```
INF RuleOverride=BINARY USE=X64 $(PROJECT_PKG)/Binary/UefiGop/XXX.inf
FILE RAW = PCD(gChipsetPkgTokenSpaceGuid.PcdVbtXXXXFile) {
```

```
$(PROJECT_PKG)/Binary/UefiGop/VBT/XXX.bin
```

```
}
```

2.9 GPIO Configuration

Alder Lake use PCD gChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported (PcdsFeatureFlag) to determine the method of GPIO config.

2.9.1 ChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported == TRUE

If gChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported be set to TRUE, project use a Spreadsheet file (file extension: .gpio) to defined the GPIO setting, and project need to do:

1. Enable PCD: gChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported.
2. Add board IDs in [SkuIds] section of \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.dsc.

Example:

```
[SkuIds]
0|DEFAULT
0x04|BoardIdTglULp4Type3
0x08|BoardIdTglUDdr4Type3Tcpd
```

3. Configure project own GPIO_\$(BOARD_NAME).gpio file and put it in the path \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Library/GpioCfgLib/
 - A. Row#1 is attribute names, Row#2 is valid attribute values. Start configuring from Row#3.
 - B. Attribute Names: *Pad Name, Pad Mode, Pad Enable, GPIO_HOSTSW_OWN, GPIO_DIRECTION, GPIO_OUTPUT_STATE, GPIO_INT_CONFIG, GPIO_RESET_CONFIG, GPIO_ELECTRICAL_CONFIG, GPIO_LOCK_CONFIG, GPIO_OTHER_CONFIG, RsvdBits, PHASE_FLAGS*
 - C. Valid Attribute Values:

GpioHardwareDefault;GpioPadModeGpio;GpioPadModeNative1;GpioPadModeNative2;GpioPadModeNative3;GpioPadModeNative4;GpioPadModeNative5,Enabled;Disabled,GpioHostOwnDefault;GpioHostOwnAcpi;GpioHostOwnGpio,GpioDirDefault;GpioDirInOut;GpioDirInInvOut;GpioDirIn;GpioDirInInv;GpioDirOut;GpioDirNone,GpioOutDefault;GpioOutLow;GpioOutHigh,GpioIntDefault;GpioIntDis;GpioIntNmi;GpioIntSmi;GpioIntSci;GpioIntApic;GpioIntLevel;GpioIntEdge;GpioIntLvlEdgDis;GpioIntBothEdge,GpioResetDefault;GpioResumeReset;GpioHostDeepReset;GpioPlatformReset;GpioDswReset,GpioTermDefault;GpioTermNone;GpioTermWpd5K;GpioTermWpd20K;GpioTermWpu1K;GpioTermWpu2K;GpioTermWpu5K;GpioTermWpu20K;GpioTermWpu1K2K;GpioTermNative;GpioNoTolerance1v8;GpioTolerance1v8,GpioLockDefault;GpioPadConfigUnlock;GpioPadConfigLock;GpioOutputStateUnlock;GpioPadUnlock;GpioPadLock,GpioRxRaw1Default;GpioRxRaw1Dis;GpioRxRaw1En;GpioDebounceDis,0,PreMem;PostMem

File Name Example:

```
GPIO_BoardIdTglULp4Type3.gpio
```

File Example:

Please see

`$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Library/GpioCfgLib/GPIO_BoardIdIglULp4Type3.gpio`

4. Modify the file name `GPIO_$(BOARD_NAME).gpio` in [Sources] section of `GpioCfgLib.inf`.

2.9.2 ChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported == FALSE

If `gChipsetPkgTokenSpaceGuid.PcdH2OGpioCfgSupported` be set to FALSE, project will use the OEM service in runtime or use the Structure PCD in build time to define the GPIO setting, and project need to do:

(1) There are two PCD to store GPIO table, "PcdBoardGpioTable", "PcdBoardGpioTablePreMem".

(2) Use OEM services

Customers can modify or replace the GPIO table by OEM service hooks, "OemSvcModifyGpioSettingTablePreMem" and "OemSvcModifyGpioSettingTable", and control the procedure functions of `GpioInit` and `GpioInitPreMem` in Chipset layer by return status.

For example:

EFI_STATUS

OemSvcModifyGpioSettingTable (

*IN OUT GPIO_INIT_CONFIG **GpioTable,*

*IN OUT UINT16 *GpioTableCount*

);

Return status:

@retval EFI_UNSUPPORTED Returns unsupported by default.

@retval EFI_MEDIA_CHANGED Alter the Configuration Parameter.

@retval EFI_SUCCESS The function performs the same operation as caller. The caller will skip the specified behavior and assuming that it has been handled completely by this function.

2.10 Super IO Support

Assume that the SIO name is XXXX in the following description.

2.10.1 Create SIO package (SioXXXXPkg)

Copy a required SIO package and rename as (SioXXXXPkg) into Insyde folder.

2.10.2 Import SIO package

Modify XXXX according to your SIO chip.

```
!import SioXXXXPkg/Package.dsc in Project.dsc.
```

```
!import SioXXXXPkg/Package.fdf in Project.fdf.
```

2.10.3 Setting the SIO related PCDs for project own requirements

Add PCDs in Project.dsc and then modify them if required.

1. Set PcdSioXXXXSetup to create a page of SIO in SCU, its default value is FALSE.

```
gSioGuid.PcdSioXXXXSetup|FALSE
```

2. Set PcdSioXXXXUpdateAsl to patch ASL code for SIO, its default value is TRUE.

Note that if you are using multiple SIO, it must be TRUE.

```
gSioGuid.PcdSioXXXXUpdateAsl|TRUE
```

3. The variable name storing SIO data for SCU.

```
gSioGuid.PcdSioXXXXSetupStr|L"SioXXXXSetup00"
```

4. Configure the parameters for your SIO chip in gSioGuid.PcdSioXXXXCfg.

```
#  
# Device Number: Com: 0x01, Floppy: 0x02, LPT: 0x3, KYBD: 0x04, MOUSE: 0x05, HardWare  
# Monitor: 0x10  
# CIR: 0x07  
#  
# TYPEH: SIO ID High Byte  
# TYPEL: SIO ID Low Byte  
# SI: SIO Instance  
# D: SIO Device  
# DI: SIO Device Instance  
# DE: SIO Device Enable  
# DBASE: SIO Device Base Address  
# SiZE: SIO Device Size  
# LDN: SIO Device LDN  
# DIRQ: SIO Device IRQ
```

```
# DDMA: SIO Device DMA
#
gSioGuid.PcdSioIt8728fCfg| {\
    #SIO TYPE | SI | D | DI | DE | DBASE | SIZE| LDN | DIRQ| DDMA
#-----
0x87, 0x28, 0x00, 0x01, 0x00, 0x01, UINT16(0x03F8), 0x00, 0x00, 0x04, 0x00, \ # Com1
0x87, 0x28, 0x00, 0x01, 0x01, 0x00, UINT16(0x02F8), 0x00, 0x00, 0x03, 0x00, \ # Com2
0x87, 0x28, 0x00, 0x02, 0x00, 0x00, UINT16(0x03F0), 0x00, 0x00, 0x06, 0x00, \ #
Floppy
0x87, 0x28, 0x00, 0x03, 0x00, 0x00, UINT16(0x0378), 0x00, 0x00, 0x07, 0x00, \ # Lpt
0x87, 0x28, 0x00, 0x04, 0x00, 0x01, UINT16(0x0060), 0x00, 0x00, 0x01, 0x00, \ # KYBD
0x87, 0x28, 0x00, 0x05, 0x01, 0x01, UINT16(0x0060), 0x00, 0x00, 0x0C, 0x00, \ #
MOUSE
0x87, 0x28, 0x00, 0x07, 0x00, 0x00, UINT16(0x0320), 0x01, 0x00, 0x0B, 0x00, \ # CIR
0x87, 0x28, 0x00, 0x10, 0x00, 0x00, UINT16(0x0290), 0x00, 0x00, 0x00, 0x00, \ # HWM
0x87, 0x28, 0x00, 0x7F, 0x00, 0x00, UINT16(0x002E), 0x00, 0x00, 0x00, 0x00, \ # CFG
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, UINT16(0x0000), 0x00, 0x00, 0x00, 0x00 \ # End
Entry
}
```

2.11 No Super IO

Do not Import SIO package in \$(PROJECT_REL_PATH)\\$(PROJECT_PKG)\Project.dsc and Project.fdf.

2.12 PCI IRQ Routing

2.12.1 Program the Legacy Routing Table

1. The related PCDs located in \$(CHIPSET_REL_PATH)/\$(CHIPSET_PKG)/Package.dsc for legacy routing table is as follows:

```
gChipsetPkgTokenSpaceGuid.PcdPirqLinkValueArray
gChipsetPkgTokenSpaceGuid.PcdControllerDeviceIrqRoutingEntry
gChipsetPkgTokenSpaceGuid.PcdVirtualBusTable
gChipsetPkgTokenSpaceGuid.PcdPirqPriorityTable
gChipsetPkgTokenSpaceGuid.PcdIrqPoolTable
```

2. Follow the description to modify the settings for your project if necessary:
3. Copy the related PCDs to the corresponding section of \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.dsc and do proper modification.
4. For the detailed format of the PCDs listed above, check \$(CHIPSET_REL_PATH)/\$(CHIPSET_PKG)/Package.dsc.

2.12.2 ACPI routing table

The related files for ACPI routing table is as follows:

`$(CHIPSET_REL_PATH)/AlderLakeBoardPkg/Acpi/AcpiTables/Dsdt/PciTree.asl`

`$(CHIPSET_REL_PATH)/AlderLakeBoardPkg/Acpi/AcpiTables/Dsdt/BusPRT/BxxPRT.asl`

Note: xx present virtual bus number. These files define the _PRT objects under corresponding bus and should be included in PciTree.asl described above

2.12.3 Example 1: Change internal device 31 Integrated GbE routing to the same PIRQC:

1. Modify the device 31 of gChipsetPkgTokenSpaceGuid.PcdControllerDeviceIrqRoutingEntry in \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.dsc:

From

```
0x00, 0xF8, 0x60, UINT16(0xDEB8), 0x61, UINT16(0xDEB8), 0x62, UINT16(0xDEB8),
0x63, UINT16(0xDEB8), 0x00, 0x00, UINT32(0x00000000), 0x00, \ #D31
```

To

```
0x00, 0xF8, 0x62, UINT16(0xDEB8), 0x61, UINT16(0xDEB8), 0x62, UINT16(0xDEB8),
0x63, UINT16(0xDEB8), 0x00, 0x00, UINT32(0x00000000), 0x00, \ #D31
```

***: Device ID is the real ID shift left 3 bits -> Dev 31 = 0x1F, 0x1F * 2^3 = 0xF8

2. According to PcdControllerDeviceIrqRoutingEntry, the bus number of the device 31 is 0x00. So modify \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/AcpiTables/Dsdt/BusPRT/B00PRT.asl

From

```
Name (PR00, Package () {
...
Package () { 0x001FFFFFF, 0, LNKA, 0 },
...
}
```

```
Name (AR00, Package () {
...
Package () { 0x001FFFFFF, 0, 0, 16 },
```

```

...
}
To
Name (PR00, Package () {
...
Package () { 0x001FFFFFF, 0, LNKC, 0 },
...
}
Name (AR00, Package () {
...
Package () { 0x001FFFFFF, 0, 0, 18 },
...
}

```

2.12.4 Example 2: Set INTA of PCIe root port 3 link to PIRQF and link PIRQF to IRQ7:

1. According to gChipsetPkgTokenSpaceGuid.PcdVirtualBusTable, the virtual bus number of PCIe root port 3(B0:D28:F2) is 0x06.

```

gChipsetPkgTokenSpaceGuid.PcdVirtualBusTable|{ \
...
0x00, 0x1c, 0x02, 0x06, \ # PCIe Root Port 3
...
} #EndEntry

```

2. Modify the device on bus 0x06 of gChipsetPkgTokenSpaceGuid.PcdControllerDeviceIrqRoutingEntry in \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.dsc

From

```

0x06, 0x00, 0x62, UINT16(0xDEB8), 0x63, UINT16(0xDEB8), 0x60, UINT16(0xDEB8),
0x61, UINT16(0xDEB8), 0x00, 0xFF, UINT32(0x00000000), 0x00, \ #B06

```

To

```

0x06, 0x00, 0x69, UINT16(0xDEB8), 0x63, UINT16(0xDEB8), 0x60, UINT16(0xDEB8),
0x61, UINT16(0xDEB8), 0x00, 0xFF, UINT32(0x00000000), 0x00, \ #B06

```

3. Assign IRQ7 to PIRQF
Modify from

```
gChipsetPkgTokenSpaceGuid.PcdPirqPriorityTable|{
```

```
...
```

```
0, \# PIRQ F
```

```
...
```

```
} #EndEntry
```

To

```
gChipsetPkgTokenSpaceGuid.PcdPirqPriorityTable|{
```

```
...
```

```
7, \# PIRQ F
```

```
...
```

```
} #EndEntry
```

4. The virtual bus number of PCIe root port 3(B0:D28:F2) is 0x06, so modify \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/AcpiTables/Dsdt/BusPRT/B06PRT.asl.

From

```
Name (PR00, Package () {
```

```
...
```

```
Package () { 0x0000FFFF, 0, LNKC, 0 },
```

```
...
```

```
}
```

```
Name (AR00, Package () {
```

```
...
```

```
Package () { 0x0000FFFF, 0, 0, 18 },
```

```
...
```

```
}
```

To

```
Name (PR00, Package () {
```

```
...
```

```
Package () { 0x0000FFFF, 0, LNKF, 0 },
```

```
...
```

```
}
```

```
Name (AR00, Package ()) {
...
Package () { 0x0000FFFF, 0, 0, 21 },
...
}
```

2.13 SODIMM

For Alder Lake SODIMM SPD customization, make sure the following PCD setting is correct:

1. Set "gBoardModuleTokenSpaceGuid.PcdSpdPresent" to TRUE.

2.14 On-board RAM

For Alder Lake On-board RAM SPD customization, Intel recommend to read the following 2 documents before project porting:

1. Doc# 573387: Memory DQ/DQS/Rcomp MRC User Guide.
2. Doc# 616599: AlderAlder Lake Platform LPDDR4x SPD BIOS Setting Instruction.

Implement or update on board SPD data for projects if necessary.

1. Set "gBoardModuleTokenSpaceGuid.PcdSpdPresent" to FALSE.
2. Set "gChipsetPkgTokenSpaceGuid.PcdH2OMemoryDownSupported" to TRUE.
3. Using the following PCDs to pass the SPD data:
 - A. PcdH2OMemoryDownSpdDataMc0Ch0Dimm0
 - B. PcdH2OMemoryDownSpdDataMc0Ch1Dimm0
 - C. PcdH2OMemoryDownSpdDataMc0Ch2Dimm0
 - D. PcdH2OMemoryDownSpdDataMc0Ch3Dimm0
 - E. PcdH2OMemoryDownSpdDataMc1Ch0Dimm0
 - F. PcdH2OMemoryDownSpdDataMc1Ch1Dimm0
 - G. PcdH2OMemoryDownSpdDataMc1Ch2Dimm0
 - H. PcdH2OMemoryDownSpdDataMc1Ch3Dimm0

Note: There are **2** memory controllers on Alder Lake.

2.15 BoardID integrate

2.15.1 There are three PCDs about board ID configuration in our source code:

1. PcdH2OBoardId: Belong to Kernel, it stores the REAL board ID for platform. It will be used to get the setup default values from Project.var or other multi-configuration mechanism provided by InsydeH2O.

OEM project needs to set the PCD by your own board ID or using OEM service "OemSvcGetBoardId()" to adjust the value.

Note: It should check the PcdH2OBoardId value is whether declared in both [SkuIds] and SKUID_IDENTIFIER of Project.dsc otherwise, the system will hang with the POST code **0x8E**.

2. PcdCrbSkuId: Belong to Chipset, the PCD is used for different SKU related initialization.

OEM project needs to set the PCD according the SKU type they used in the project.

AlderLake RC use below SKU Type for different platform:

AdlSSkuType	1
AdlPSkuType	2
AdlMSkuType	3

3. PcdBoardId: Belong to Chipset, the PCD is used for different RVP board related initialization, update RVP board config and affect those PCDs as following, PcdBoardName/PcdBoardType/PcdPlatformType/PcdPlatformFlavor/PcdAcpiDefaultOemTableId.

OEM project needs to set the PCD according the RVP board they referenced.

4. PcdCrbBoard: Belong to Chipset, the PCD of gChipsetPkgTokenSpaceGuid.PcdCrbBoard to control it's CRB or OEM Board, this PCD will be affected by CrbBuild flag in bat and makefile.

2.15.2 How to set board ID correctly on OEM project side?

1. Set CrbBuild=NO in MultiBoardPkg\ProjectSetup.bat.
2. **Set PcdH2OBoardId to your platform's real board ID (Project Board ID)**, you also can use OemSvcGetBoardId() to return the board ID to system.
3. For follow Kernel SkuID feature, add your project board name to the SKUID_IDENTIFIER of [Defines] section on Project.dsc.
Example: **SKUID_IDENTIFIER = DEFAULT|SkuIdAdlS**
4. For follow Kernel SkuID feature, add your project board ID value and board name to [SkuIds] section on Project.dsc.
Example: **0x03|SkuIdAdlS**
5. Set PcdCrbSkuId correctly, which SKU is your project referenced, normally, this PCD will be set in Board detect function by PcdBoardId.
6. **Set PcdBoardId correctly, which RVP board is your project board referenced**, this PCD will be updated by OemSvcGetBoardId function.

7. **Set PcdBoardBomId and PcdBoardRev correctly, which RVP board is your project board referenced.** In CRB, this PCD will be updated by Board detect function.

* OEM also can customize PcdCrbSkuId, PcdBoardId, PcdBoardBomId and PcdBoardRev in "CustomerMultiBoardDetect" function as below file path,

Board\Intel\AlderLakePMultiBoardPkg\Library\BoardInitLib\Pei\PeiMultiBoardSupportLib.c

2.15.3 Compare to old BOARDID feature

Old	New
Set PcdDefaultBoardId = 0xFF for OEM project	Set PcdCrbBoard = NO (CrbBuild=NO) for OEM project
Use PcdOemProjectReferenceIntelCrb to select Intel Platform BoardIds	Use PcdBoardId to select Intel Platform BoardIds

2.16 Native FSP Build

If you want to use Intel Native FSP binaries on your project, you should follow the steps as follows:

1. Set gChipsetPkgTokenSpaceGuid.PcdNativeFspBuild to TRUE in Project.dsc
2. Set the FSP version at PCD of gChipsetPkgTokenSpaceGuid.PcdNativeFspVersion that you get from Intel website in Project.dsc

```
gChipsetPkgTokenSpaceGuid.PcdNativeFspVersion|"2277_11_182"
```

3. Get Native FSP binaries from Intel website and put it into Native FSP binary folders

```
Intel\AlderLake\AlderLakeFspBinPkg\NativeFspRelease\%NativeFspVersion%\
```

```
Intel\AlderLake\AlderLakePFspBinPkg\NativeFspDebug\%NativeFspVersion%\
```

4. Build the BIOS that you want, release version(uefi64) or debug version(efidebug), but Native FSP build doesn't support Insyde DDT, you only use the release version of FSP binaries when you building BIOS of DDT version.

3 Power-On Tips

3.1 Minimum Devices Support

Disable devices in UefiSetupUtility to minimize the system configuration.

Example:

```
PCH_SETUP.PchHdAudio
```

```
PCH_SETUP.PchLan
```

```
PCH_SETUP.PchSata
```

```
PCH_SETUP.PcieRootPortEn*
```

3.2 Minimum CPU Features Support

Disable CPU features support to minimize CPU capability.

Example:

```
CPU_SETUP.HyperThreading
```

```
CPU_SETUP.ActiveCoreCount
```

```
CPU_SETUP.EnableCx
```

```
CPU_SETUP.TurboMode
```

```
CPU_SETUP.EnableThermalMonitor
```

```
CPU_SETUP.VT
```

```
CPU_SETUP.Txt
```

3.3 Boot Mode

Make sure boot mode is cold boot during bring up, steps are as follows.

1. Set boot mode as *BOOT_WITH_FULL_CONFIGURATION*. *SkipPriorityPolicy = TRUE and return *EFI_SUCCESS* in *OemSvcChangeBootMode* ().
2. Disable fast crisis recovery in $\$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.env$, i.e. Set *USE_FAST_CRISIS_RECOVERY* = NO
3. Return *IsRecovery* = FALSE in $\$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Library/PeiOemSvcKernelLib/OemSvcDetectRecoveryRequest.c$

3.4 Policy Setting

Policy settings are list as follows:

1. Intel RC Pei policy:

```
$(CHIPSET_REL_PATH)\$(PLATFORMSAMPLE_PACKAGE)\Library\PeiPolicyUpdateLib\
PeiPolicyUpdateLib.inf
```

```
PeiPchPolicyUpdatePreMem.c
```

```
PeiSaPolicyUpdatePreMem.c
```

```
PeiCpuPolicyUpdatePreMem.c
```

```
PeiMePolicyUpdatePreMem.c
```

```
PeiPchPolicyUpdate.c
```

```
PeiSaPolicyUpdate.c
```

```
PeiCpuPolicyUpdate.c
```

```
PeiMePolicyUpdate.c
```

```
PeiAmtPolicyUpdate.c
```

2. Intel RC Dxe policy:

```
$(CHIPSET_REL_PATH)\$(PLATFORMSAMPLE_PACKAGE)\Library\DxePolicyUpdateLib\
DxePolicyUpdateLib.inf
```

```
DxeAmtPolicyUpdate.c
```

```
DxeMePolicyUpdate.c
```

```
DxeSaPolicyUpdate.c
```

```
DxePchPolicyUpdate.c
```

```
DxeCpuPolicyUpdate.c
```

```
DxeTbtPolicyUpdate.c
```

3.5 Emulation Variable

Use Emulation Variable when the flash part is not ready during power on phase.

Follow the steps to enable variable emulation:

1. Disable fast crisis recovery:

Modify the following description in \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.env

```
EDK_GLOBAL_USE_FAST_CRISIS_RECOVERY = NO
```

2. Disable VariableRuntimeDxe driver and add EmuVariableRuntimeDxe driver. Modify the following description in the [Components.\$(DXE_ARCH)] section of \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.dsc.

```
!disable InsydeModulePkg/Universal/Variable/VariableRuntimeDxe/VariableRuntimeDxe.inf
InsydeModulePkg/Universal/Variable/EmuVariableRuntimeDxe/EmuVariableRuntimeDxe.inf
```

Modify the following description in the [FV.DXE_FV] section of \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.fdf.

```
!disable InsydeModulePkg/Universal/Variable/VariableRuntimeDxe/VariableRuntimeDxe.inf
INF InsydeModulePkg/Universal/Variable/EmuVariableRuntimeDxe/EmuVariableRuntimeDxe.inf
```

3.6 Debug Methods

There are several debug methods.

Modify the following description in \$(PROJECT_REL_PATH)/\$(PROJECT_PKG)/Project.env.

1. Serial port

```
EFI_DEBUG = YES
```

Note: if using PCH UARTx for efidebug print, please set "Kernel Debug Serial Port" item to "SerialIO UARTx" in Debug Setting SCU page, otherwise, efidebug message would be lost after ExitBootService.

2. Insyde DDT

```
INSYDE_DEBUGGER = YES
```

3.7 Add Option ROM

Implement project option Rom table in:

```
$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Library/DxeOemSvcKernelLib/OemSvcInstallOptionRomTable.c
```

3.8 GlobalNvsArea mechanism

1. Platform global NVS area is defined in:

```
$(CHIPSET_REL_PATH)/$(CHIPSET_PKG)/Include/Protocol/PlatformNvsArea.h
```

```
$(CHIPSET_REL_PATH)/$(CHIPSET_PKG)/Include/PlatformNvsAreaDef.h
```

```
$(CHIPSET_REL_PATH)/AlderLakeBoardPkg/Acpi/AcpiTables/Dsdt/PlatformNvs.asl
```

Note: The definition of PlatformNvsAreaDef.h must match PlatformNvs.asl

2. OEM global NVS area is defined in:

`$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Include/OemPlatformNvsArea.h`

`$(CHIPSET_REL_PATH)/AlderLakeBoardPkg/Acpi/AcpiTables/Dsdt/OemPlatformNvs.asl`

Note: The definition of OemPlatformNvsArea.h must match OemPlatformNvs.asl

3. OEM can modify OEM global NVS data in OemSvcUpdatePlatformNvs.c

3.9 Software SMI

SwSmi is defined in:

1. Insyde/InsydeModulePkg/Include/SmiTable.h
2. `$(CHIPSET_REL_PATH)/$(CHIPSET_PKG)/Include/ChipsetSmiTable.h`
3. `$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Include/OemSwSmi.h`

3.10 CMOS Offset Definition

CMOS offset is defined in:

1. Insyde/InsydeModulePkg/Include/Library/CmosLib.h
2. `$(CHIPSET_REL_PATH)/$(CHIPSET_PKG)/Include/ChipsetCmos.h`
3. `$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Include/OemCmos.h`

3.11 SVID/SSID

There are 2 interfaces to implement SVID/SSID.

1. PCDs:
 - A. gInsydeTokenSpaceGuid.PcdDefaultSsidSvid
 - B. gChipsetPkgTokenSpaceGuid.PcdDefaultSsidSvidPeiTable
2. OEM services:
 - A. OemSvcUpdateFspsUpd.c (only for API mode)
 - B. OemSvcUpdateSsidSvidInfo.c.

3.12 Setup Configuration

Setup configuration is a union structure of kernel chipset and OEM/ODM and defined in

`$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Include/SetupConfig.h.`

1. Kernel setup configuration is defined in Insyde/InsydeModulePkg/Include/KernelSetupData.h.

2. Chipset setup configuration is defined in `$(CHIPSET_REL_PATH)/$(CHIPSET_PKG)/Include/ChipsetSpecificSetupData.h`.
3. OEM/ODM configuration is defined in `OEMRSV[70]` and `ODMRSV[100]` in `$(PROJECT_REL_PATH)/$(PROJECT_PKG)/Include/SetupConfig.h`.
4. Project Board Design Checklist

Insyde Software Corp.

4 Project Board Design Checklist

4.1 H2ODDT USB Port Support

EHCI Controller is removed from Alder Lake, so USB 3.0 debug cable is required for H2ODDT debugging. Please refer "InsydeH2O Alder Lake 5.4 OEM Customization Guide" for the details.

For BIOS side, please modify Project.env:

```
EDK_GLOBAL H2O_DDT_DEBUG_IO = Xhc
```

4.2 H2ODDT Com Port Support

Please refer "InsydeH2O 5.4 Alder Lake OEM Customization Guide - Appendix D" for the details.

For BIOS side, please modify Project.env:

```
EDK_GLOBAL H2O_DDT_DEBUG_IO = Com
```

4.3 H2ODDT PCH UART Support

Please refer "InsydeH2O 5.4 Alder Lake OEM Customization Guide - Appendix D" for the details.

For BIOS side, please modify Project.env:

```
EDK_GLOBAL H2O_DDT_DEBUG_IO = Com
EDK_GLOBAL DEBUG_USE_PCH_COMPORT = YES
```

Run Insyde\InsydeModulePkg\H2ODebug\DebugConfig.exe:

ADL-S: (PCH UART 2)

```
Com port Type = MMIO8
IO/MMIO Address = FE042000
```

ADL-P: (PCH UART 0)

```
Com port Type = MMIO8
IO/MMIO Address = FE03E000
```

Re-build System BIOS